# Game Redesign in No-regret Game Playing

**Yuzhe Ma** , **Young Wu** , **Xiaojin Zhu**
University of Wisconsin–Madison
{yzm234, yw, jerryzhu}@cs.wisc.edu

## Abstract

We study the game redesign problem in which an external designer has the ability to change the payoff function in each round, but incurs a design cost for deviating from the original game. The players apply no-regret learning algorithms to repeatedly play the changed games with limited feedback. The goals of the designer are to (i) incentivize players to take a specific target action profile frequently; (ii) incur small cumulative design cost. We present game redesign algorithms with the guarantee that the target action profile is played in $T-o(T)$ rounds while incurring only $o(T)$ cumulative design cost. Simulations on four classic games confirm the effectiveness of our proposed redesign algorithms.

## 1 Introduction

Consider a finite normal-form game with loss function $\ell^o$. This is the "original game." As an example, the Volunteer's Dilemma (see Table 1) has each player choose whether or not to volunteer for a cause that benefits all players. It is known that all pure Nash equilibria in this game involve a subset of the players free-riding the contribution from the remaining players. $M$ players, who initially do not know $\ell^o$, use no-regret algorithms to individually choose their action in each of the $t = 1 \dots T$ rounds. The players receive limited feedback: suppose the chosen action profile in round $t$ is $a^t = (a_1^t, \dots, a_M^t)$, then the $i$-th player only receives her own loss $\ell_i^o(a^t)$ but not the other players' actions or losses.

Game redesign is the following task. A game designer – not a player – does not like the solution concept to $\ell^o$. Instead, the designer wants to incentivize a target action profile $a^\dagger$, for example "every player volunteers". The designer has the power to redesign the game: before each round $t$ is played, the designer can change $\ell^o$ to some $\ell^t$. The players will receive new losses $\ell_i^t(a^t)$, but the designer pays a design cost $C(\ell^o, \ell^t, a^t)$ in that round for deviating from $\ell^o$. The designer's goal is to make the players play the target action profile $a^\dagger$ in the vast majority ($T - o(T)$) of rounds, while incurring $o(T)$ cumulative design cost. Game redesign naturally emerges in two opposing contexts:

- A benevolent designer (interested party) wants to redesign the game to improve social welfare, as in the Volunteer's Dilemma. This is the motivation behind $k$-implementation [Monderer and Tennenholtz, 2004];
- A malicious designer (attacker) wants to poison the payoffs to force a nefarious target action profile. This is an extension of reward-poisoning attacks (previously studied on bandits [Jun *et al.*, 2018; Liu and Shroff, 2019; Ma *et al.*, 2018; Yang *et al.*, 2021; Guan *et al.*, 2020; Garcelon *et al.*, 2020; Bogunovic *et al.*, 2021; Zuo, 2020; Lu *et al.*, 2021] and reinforcement learning [Zhang *et al.*, 2020; Ma *et al.*, 2019; Rakhsha *et al.*, 2020; Sun *et al.*, 2020; Huang and Zhu, 2019]) to game playing.

For both contexts the mathematical question is the same. Since the design costs are measured by deviations from the original game $\ell^o$, the designer is not totally free in creating new games. Our idea for successful game redesign is:

1. Do not change the loss of the target action profile, i.e. let $\ell^t(a^\dagger) = \ell^o(a^\dagger), \forall t$. If game redesign is indeed successful, then $a^\dagger$ will be played for $T - o(T)$ rounds. As we will see, $\ell^t(a^\dagger) = \ell^o(a^\dagger)$ means there is no design cost in those rounds under our definition of $C$. The remaining rounds incur at most $o(T)$ cumulative design cost.
2. The target action profile $a^\dagger$ forms a strictly dominant strategy equilibrium. This ensures no-regret players will eventually learn to prefer $a^\dagger$ over any other action profiles.

Game redesign is closely related to the $k$-implementation problem [Monderer and Tennenholtz, 2004]. Both aim to manipulate player behaviors by changing the payoff. However, there are major differences: $k$-implementation assumes players know the game, while in our case the players have to learn the game; $k$-implementation only allows increase to existing payoffs, while we allow both positive (subsidy) and negative (tax) changes. Our interior design (Algorithm 1) indeed produces a 0-implementation in their terminology because we keep the payoff of the desired strategy profile unchanged. Nonetheless, our players have to discover this strategy profile by exploration, meaning that the designer will still incur costs especially in earlier rounds.

More broadly, game redesign is related to, but distinct from, constrained mechanism design. The players in game redesign are no-regret learners, not rational (best-response) players of a repeated game.

## 2 Formal Definition

We first describe the original game without the designer. There are $M$ players. Let $\mathcal{A}_i$ be the finite action space of player $i$, and let $A_i = |\mathcal{A}_i|$. The original game is defined by the loss function $\ell^o : \mathcal{A}_1 \times \ldots \mathcal{A}_M \mapsto \mathbb{R}^M$. The players do not know $\ell^o$. Instead, we assume they play the game for $T$ rounds using no-regret algorithms. This may be the case, for example, if the players are learning an approximate Nash equilibrium in zero-sum $\ell^o$ or coarse correlated equilibrium in general sum $\ell^o$. In running the no-regret algorithm, the players maintain their own action selection policies $\pi_i^t \in \Delta^{\mathcal{A}_i}$ over time, where $\Delta^{\mathcal{A}_i}$ is the probability simplex over $\mathcal{A}_i$. In each round $t$, every player $i$ samples an action $a_i^t$ according to policy $\pi_i^t$. This forms an action profile $a^t = (a_1^t, \ldots, a_M^t)$. The original game produces the loss vector $\ell^o(a^t) = (\ell_1^o(a^t), ..., \ell_M^o(a^t))$. However, player $i$ only observes her own loss value $\ell_i^o(a^t)$, not the other players' losses or their actions. All players then update their policy according to their no-regret algorithms.

We now bring in the designer. The designer knows $\ell^o$ and wants players to frequently play an arbitrary but fixed target action profile $a^\dagger$. We stress that $a^\dagger$ does not need to coincide with any solution concept in $\ell^o$. At the beginning of round $t$, the designer commits to a potentially different loss function $\ell^t$. Note this involves preparing the loss vector $\ell^t(a)$ for all action profiles $a$ (i.e. "cells" in the payoff matrix). The players then choose their action profile $a^t$. Importantly, the players receive losses $\ell^t(a^t)$, not $\ell^o(a^t)$. For example, in games involving money such as the volunteer game, the designer may achieve $\ell^t(a^t)$ via taxes or subsidies, and in zero-sum games such as the rock-paper-scissors game, the designer essentially "makes up" a new outcome and tell each player whether they win, tie, or lose via $\ell_i^t(a^t)$; The designer incurs a cost $C(\ell^o, \ell^t, a^t)$ for deviating from $\ell^o$. The interaction among the designer and the players is summarized as below.

**Protocol**: Game Redesign

```
Designer knows $\ell^o$, $a^\dagger$, $M$, $\mathcal{A}_{1:M}$, and player no-regret rate $\alpha$
  for $t = 1, \ldots, T$ do
    Designer prepares new loss function $\ell^t$.
    Players form action profile $a^t = (a_1^t, ..., a_M^t)$, where $a_i^t \sim \pi_i^t, \forall i \in [M]$.
    Player $i$ observes its loss $\ell_i^t(a^t)$, updates policy $\pi_i^t$.
    Designer incurs cost $C(\ell^o, \ell^t, a^t)$.
  end for
```

The designer has two goals simultaneously:

1. To incentivize the players to frequently choose the target action profile $a^\dagger$ (which may not coincide with any solution concept of $\ell^o$). Let $N^T(a) = \sum_{t=1}^T \mathbb{1}\left[a^t = a\right]$ be the number of times an action profile $a$ is chosen in $T$ rounds, then this goal is to achieve $\mathbf{E}\left[N^T(a^\dagger)\right] = T - o(T)$.

2. To have a small cumulative design cost $C^T := \sum_{t=1}^T C(\ell^o, \ell^t, a^t)$, specifically $\mathbf{E}\left[C^T\right] = o(T)$.

The per-round design cost $C(\ell^o, \ell^t, a)$ is application dependent. One plausible is to account for the **overall cost** in all action profiles, not just what is actually chosen: an example is $C(\ell^o, \ell^t, a^t) = \sum_a \|\ell^o(a) - \ell^t(a)\|_1$. Note that it ignores the $a^t$ argument. In many applications, though, only the chosen action profile costs the designer (the **implementation cost** in [Monderer and Tennenholtz, 2004]). An example is $C(\ell^o, \ell^t, a^t) = \|\ell^o(a^t) - \ell^t(a^t)\|_1$. We use a slight generalization of the latter cost:

**Assumption 1.** *The non-negative designer cost function $C$ satisfies $\forall t, \forall a^t, C(\ell^o, \ell^t, a^t) \le \eta \|\ell^o(a^t) - \ell^t(a^t)\|_p$ for some Lipschitz constant $\eta$ and norm $p \ge 1$.*

This implies no design cost if the losses are not modified, i.e., when $\ell^o(a^t) = \ell^t(a^t)$, $C(\ell^o, \ell^t, a^t) = 0$ .

## 3 Assumption: No-Regret Players

The designer assumes that the players are each running a no-regret learning algorithm like EXP3.P [Bubeck and Cesa-Bianchi, 2012]. It is well-known that for two-player ($M = 2$) zero-sum games, no-regret learners can find an approximate Nash Equilibrium [BLUM, 2007]. More general results suggest that for multi-player ($M \ge 2$) general-sum games, no-regret learners can find an approximate Coarse Correlated Equilibrium [Hart and Mas-Colell, 2000]. We first define the player's regret. We use $a_{-i}^t$ to denote the actions selected by all players except player $i$ in round $t$.

**Definition 1.** *(Regret). For any player $i$, the best-in-hindsight regret with respect to a sequence of loss functions $\ell_i^t(\cdot, a_{-i}^t), t \in [T]$, is defined as*

$$R_i^T = \sum_{t=1}^T \ell_i^t(a_i^t, a_{-i}^t) - \min_{a_i \in \mathcal{A}_i} \sum_{t=1}^T \ell_i^t(a_i, a_{-i}^t). \quad (1)$$

*The expected regret is defined as $\mathbf{E}\left[R_i^T\right]$, where the expectation is taken with respect to the randomness in the selection of actions $a^t, t \in [T]$ over all players.*

**Remark.** *The loss functions $\ell_i^t(\cdot, a_{-i}^t), t \in [T]$ depend on the actions selected by the other players $a_{-i}^t$, while $a_{-i}^t$ further depends on $a^1, ..., a^{t-1}$ of all players in the first $t-1$ rounds. Therefore, $\ell_i^t(\cdot, a_{-i}^t)$ depends on $a_i^1, ..., a_i^{t-1}$. That means, from player $i$'s perspective, the player is faced with a non-oblivious (adaptive) adversary [Slivkins, 2019].*

**Remark.** *Note that $a_i^* :=$ argmin$_{a_i \in \mathcal{A}_i} \sum_{t=1}^T \ell_i^t(a_i, a_{-i}^t)$ in* (1) *would have meant a baseline in which player $i$ always plays the best-in-hindsight action $a_i^*$ in all rounds $t \in [T]$. Such baseline action should have caused all other players to change their plays away from $a_{-i}^1, ..., a_{-i}^T$. However, we are disregarding this fact in* (1)*. For this reason,* (1) *is not fully counterfactual, and is called the best-in-hindsight regret [Bubeck and Cesa-Bianchi, 2012]. The same is true when we define the expected regret.*

Our key assumption is that the learners achieve sublinear regret. This assumption is satisfied by standard bandit algorithms such as EXP3.P [Bubeck and Cesa-Bianchi, 2012].

**Assumption 2.** *(No-regret Learner) We assume the players apply no-regret learning algorithm that achieves expected regret $\mathbf{E}\left[R_i^T\right] = O(T^\alpha), \forall i$ for some $\alpha \in [0, 1)$.*

# 4 Game Redesign Algorithms

There is an important consideration regarding the allowed values of $\ell^t$. The original game $\ell^o$ has a set of "natural loss values" $\mathcal{L}$. For example, in the rock-paper-scissors game $\mathcal{L} = \{-1, 0, 1\}$ for the player wins (recall the value is the loss), ties, and loses, respectively; while for games involving money it is often reasonable to assume $\mathcal{L}$ as some interval $[L, U]$. Ideally, $\ell^t$ should take values in $\mathcal{L}$ to match the semantics of the game or to avoid suspicion (in the attack context). Our designer can work with discrete $\mathcal{L}$ (section 4.3); but for exposition we will first allow $\ell^t$ to take real values in $\tilde{\mathcal{L}} = [L, U]$, where $L = \min_{x \in \mathcal{L}} x$ and $U = \max_{x \in L} x$. We assume $U$ and $L$ are the same for all players and $U > L$, which is satisfied when $\mathcal{L}$ contains at least two distinct values.

## 4.1 Algorithm: Interior Design

The name refers to the narrow applicability of Algorithm 1: the original loss values for the target action profile $\ell^o(a^\dagger)$ must all be in the interior of $\tilde{\mathcal{L}}$. Formally, we require $\exists \rho \in (0, \frac{U-L}{2}]$, $\forall i, \ell_i^o(a^\dagger) \in [L + \rho, U - \rho]$. In Algorithm 1, we present the interior design. The key insight is to keep $\ell^o(a^\dagger)$ unchanged: If the designer is successful, $a^\dagger$ will be played in $T - o(T)$ rounds. In these rounds, the designer cost is zero. The other $o(T)$ rounds each incur bounded cost. Overall, this ensures sublinear design cost. For the design to be successful, the designer can make $a^\dagger$ the strictly dominant strategy. The designer can do this by judiciously increasing or decreasing the loss of other action profiles in $\ell^o$: there is enough room because $\ell^o(a^\dagger)$ is in the interior. In fact, the designer can design a time-invariant game $\ell^t = \ell$ as Algorithm 1 shows.

**Algorithm 1** Interior Design

**Input:** the target action profile $a^\dagger$; the original game $\ell^o$.
**Output:** a time-invariant game $\ell$ constructed as follows:

$$\forall i, a, \ell_i(a) = \begin{cases} \ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho & \text{if } a_i = a_i^\dagger, \\ \ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho & \text{if } a_i \neq a_i^\dagger, \end{cases} \tag{2}$$

where $d(a) = \sum_{j=1}^M \mathbb{1}\left[a_j = a_j^\dagger\right]$.

**Lemma 3.** *The redesigned game* (2) *satisfies:*

1. *$\forall i, a, \ell_i(a) \in \tilde{\mathcal{L}}$, thus $\ell$ is valid.*
2. *For every player $i$, the target action $a_i^\dagger$ strictly dominates any other action by $(1 - \frac{1}{M})\rho$, i.e., $\ell_i(a_i, a_{-i}) = \ell_i(a_i^\dagger, a_{-i}) + (1 - \frac{1}{M})\rho, \forall i, a_i \neq a_i^\dagger, a_{-i}$.*
3. *$\ell(a^\dagger) = \ell^o(a^\dagger)$.*
4. *If the original loss for the target action profile $\ell^o(a^\dagger)$ is zero-sum, then the redesigned game $\ell$ is also zero-sum.*

Our main result is that Algorithm 1 achieves the design goal with sublinear cumulative design cost. It is worth noting that although many entries in the redesigned game $\ell$ can appear to be quite different than the original game $\ell^o$, their contribution to the design cost is small because the design discourages them from being played often.

**Theorem 4.** *Using Algorithm 1, the designer can achieve $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^\alpha)$ while incurring expected cumulative design cost $\mathbf{E}\left[C^T\right] = O(\eta M^{1+\frac{1}{p}} T^\alpha)$.*

**Corollary 5.** *If the players use EXP3.P, the designer can achieve $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^{\frac{1}{2}})$ while incurring expected cumulative design cost $\mathbf{E}\left[C^T\right] = O(\eta M^{1+\frac{1}{p}} T^{\frac{1}{2}})$.*

If the original game $\ell^o$ is two-player zero-sum, then under redesign, players will think that $a^\dagger$ is a Nash equilibrium.

**Corollary 6.** *Assume $M = 2$ and $\ell^o$ is zero-sum. Then with the redesigned game* (2)*, the expected averaged policy $\mathbf{E}\left[\bar{\pi}_i^T\right] = \mathbf{E}\left[\frac{1}{T}\sum_t \pi_i^t\right]$ converges to a point mass on $a_i^\dagger$.*

## 4.2 Boundary Design

When $\ell^o(a^\dagger)$ has some values hitting the boundary of $\tilde{\mathcal{L}}$, the designer cannot apply Algorithm 1 directly because the loss of other action profiles cannot be increased or decreased further to make $a^\dagger$ a dominant strategy. However, a time-varying design can still achieve the design goals with sublinear design cost. In Algorithm 2, we present the boundary design which is applicable to both boundary and interior $\ell^o(a^\dagger)$ values.

**Algorithm 2** Boundary Design

```
Input: the target action profile $a^\dagger$; a loss vector $v \in \mathbb{R}^M$
    whose elements are in the interior, i.e., $\forall i, v_i \in [L + \rho, U - \rho]$ for some $\rho > 0$; the regret rate $\alpha$; $\epsilon \in (0, 1-\alpha)$;
Output: a time-varying game with loss $\ell^t, t \in [T]$.
  Use $v$ in place of $\ell^o(a^\dagger)$ in (2) and apply the interior design 1. Call the resulting game the "source game" $\underline{\ell}$.
  Define a "destination game" $\bar{\ell}$ where $\bar{\ell}(a) = \ell^o(a^\dagger), \forall a$.
  Interpolate the source and destination games:
      $\ell^t = w_t \underline{\ell} + (1 - w_t)\bar{\ell}$      (3)
  where $w_t = t^{\alpha+\epsilon-1}$.
```

The designer can choose any loss vector $v$ as long as $v$ lies in the interior of $\tilde{\mathcal{L}}$. We give two exemplary choices of $v$.

1. Let the average player cost of $a^\dagger$ be $\bar{\ell}(a^\dagger) = \sum_{i=1}^M \ell_i^o(a^\dagger)/M$, then if $\bar{\ell}(a^\dagger) \in (L, U)$, one could choose $v$ to be a constant vector with value $\bar{\ell}(a^\dagger)$. The nice property about this choice is that if $\ell^o$ is zero-sum, then $v$ is zero-sum, thus property 4 is satisfied and the redesigned game is zero-sum. However, note that when $\bar{\ell}(a^\dagger)$ does hit the boundary, the designer cannot choose this $v$.
2. Choose $v$ to be a constant vector with value $(L + U)/2$. This choice is always valid, but may not preserve the zero-sum property of the original game unless $L = -U$.

The designer applies the interior design on $v$ to obtain a "source game" $\underline{\ell}$. Note that the target action profile $a^\dagger$ strictly dominates in the source game. The designer also creates a "destination game" $\bar{\ell}(a)$ by repeating the $\ell^o(a^\dagger)$ entry everywhere. The boundary algorithm then interpolates between the source and destination games with a decaying weight $w_t$. Note after interpolation (3), the target $a^\dagger$ still dominates by roughly $w_t$. We design the weight $w_t = t^{\alpha+\epsilon-1}$ so that cumulatively, the sum of $w_t$ grows with rate $\alpha + \epsilon$, which is faster than the regret rate $\alpha$. This is a critical consideration to enforce frequent play of $a^\dagger$. Also note that asymptotically, $\ell^t$ converges toward the destination game. Therefore, in the long run, when $a^\dagger$ is played the designer incurs diminishing cost, resulting in $o(T)$ cumulative design cost.

**Lemma 7.** *The redesigned game* (3) *satisfies:*

1. $\forall i, a, \ell_i^t(a) \in \tilde{\mathcal{L}}$*, thus the loss function is valid.*
2. *For every player* $i$*, the target action* $a_i^\dagger$ *strictly dominates any other action by* $(1 - \frac{1}{M})\rho w_t$*, i.e.,* $\ell_i^t(a_i, a_{-i}) = \ell_i^t(a_i^\dagger, a_{-i}) + (1 - \frac{1}{M})\rho w_t, \forall i, t, a_i \neq a_i^\dagger, a_{-i}$.
3. $\forall t, C(\ell^o, \ell^t, a^\dagger) \leq \eta(U - L)M^{\frac{1}{p}} w_t$
4. *If the original loss for the target action profile* $\ell^o(a^\dagger)$ *and the vector* $v$ *are both zero-sum, then* $\forall t, \ell^t$ *is zero-sum.*

Given Lemma 7, we provide our second main result.

**Theorem 8.** *Using Algorithm 2, the designer can achieve* $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^{1-\epsilon})$ *while incurring expected cumulative design cost* $\mathbf{E}\left[C^T\right] = O(M^{1+\frac{1}{p}}T^{1-\epsilon} + M^{\frac{1}{p}}T^{\alpha+\epsilon})$.

**Remark.** *By choosing a larger* $\epsilon$ *in Theorem 8, the designer increases* $\mathbf{E}\left[N^T(a^\dagger)\right]$*. However, the design cost can grow. When* $\epsilon = \frac{1-\alpha}{2}$*, the design cost attains the minimum order* $O\left(T^{\frac{1+\alpha}{2}}\right)$ *and* $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(T^{\frac{1+\alpha}{2}})$

**Corollary 9.** *Assume the no-regret learning algorithm is EXP3.P. The designer can achieve expected number of target plays* $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^{\frac{3}{4}})$ *while incurring* $\mathbf{E}\left[C^T\right] = O\left(M^{\frac{1}{p}}(1 + M)T^{\frac{3}{4}}\right)$ *design cost.*

### 4.3 Discrete Design

In previous sections, we assumed the games $\ell^t$ can take arbitrary continuous values in the relaxed loss range $\tilde{\mathcal{L}} = [L, U]$. However, there are many real-world situations where continuous loss does not have a natural interpretation. For example, in the rock-paper-scissors game, the loss is interpreted as win, lose or tie, thus $\ell^t$ should only take value in the original loss value set $\mathcal{L} = \{-1, 0, 1\}$. We now provide a discrete redesign to convert any game $\ell^t$ with values in $\tilde{\mathcal{L}}$ into a game $\hat{\ell}^t$ only involving loss values $L$ and $U$, which are both in $\mathcal{L}$. Specifically, the discrete design is illustrated in Algorithm 3.

**Algorithm 3** Discrete Design

**Input:** the target action profile $a^\dagger$; a loss vector $v \in \mathbb{R}^M$ whose elements are in the interior, i.e., $\forall i, v_i \in [L+\rho, U-\rho]$ for some $\rho > 0$; the regret rate $\alpha$; $\epsilon \in (0, 1-\alpha)$;

**Output:** a time-varying game with loss $\hat{\ell}^t \in \mathcal{L}$ as below:

$$\forall t, i, a, \hat{\ell}_i^t(a) = \begin{cases} U & \text{with probability } \frac{\ell_i^t(a)-L}{U-L} \\ L & \text{with probability } \frac{U-\ell_i^t(a)}{U-L}. \end{cases} \qquad (4)$$

It is easy to verify $\mathbf{E}\left[\hat{\ell}^t\right] = \ell^t$. In experiments we show such discrete games also achieve the design goals.

### 4.4 Thresholding the Redesigned Game

For all designs in previous sections, the designer could impose an additional min or max operator to threshold on the original game loss, e.g., for the interior design, the redesigned game loss after thresholding becomes $\forall i, a$,

$$\ell_i(a) = \begin{cases} \min\{\ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho, \ell^o(a)\} & \text{if } a_i = a_i^\dagger, \\ \max\{\ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho, \ell^o(a)\} & \text{if } a_i \neq a_i^\dagger. \end{cases} \qquad (5)$$

We point out a few differences between (5) and (2). First, (5) guarantees a dominance gap of “at least” (instead of exactly) $(1 - \frac{1}{M})\rho$. As a result, the thresholded game can induce a larger $N^T(a^\dagger)$ because the target action $a^\dagger$ is redesigned to stand out even more. Second, one can easily show that (5) incurs less design cost $C^T$ compared to (2) due to thresholding. Therefore, Theorem 4 still holds. However, thresholding no longer preserves the zero-sum property.

## 5 Experiments

We perform empirical evaluations of game redesign algorithms on four games — the volunteer’s dilemma (VD), tragedy of the commons (TC), prisoner’s dilemma (PD) and rock-paper-scissors (RPS). Throughout the experiments, we use EXP3.P [Bubeck and Cesa-Bianchi, 2012] as the no-regret learner. The concrete form of the regret bound for EXP3.P is illustrated in the appendix. Based on that, we derive the exact form of our theoretical upper bounds for Theorem 4 and Theorem 8, and we show the theoretical value for comparison in our experiments. We let the designer cost function be $C(\ell^o, \ell^t, a^t) = \|\ell^o(a^t) - \ell^t(a^t)\|_p$ with $p = 1$. For VD, TC and PD, the original game is not zero-sum, and we apply the thresholding (5) to slightly improve the redesign performance. For the RPS game, we apply the design without thresholding to preserve the zero-sum property. The results we show in all the plots are produced by taking the average of 5 trials.

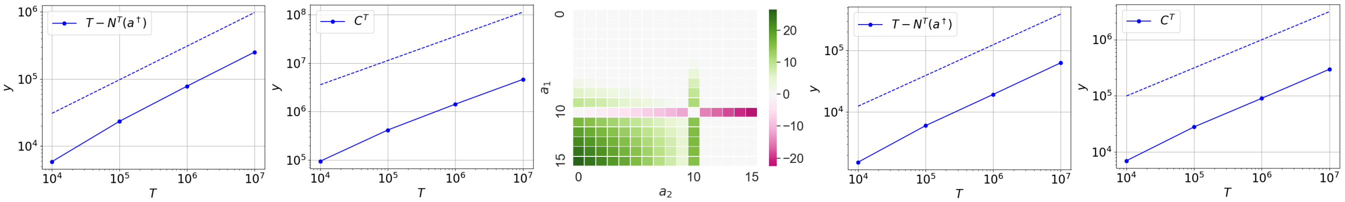
### 5.1 Volunteer’s Dilemma (VD)

In volunteer’s dilemma (Table 1) there are $M$ players. Each player has two actions: volunteer or not. When there exists at least one volunteer, those players who do not volunteer gain 1 (i.e. a $-1$ loss). The volunteers receive zero payoff. On the other hand, if no players volunteer, then every player loss 10.

| | | Other players | |
|---|---|---|---|
| | | exists a volunteer | no volunteer exists |
| Player $i$ | volunteer | 0 | 0 |
| | not volunteer | $-1$ | 10 |

Table 1: The loss function $\ell_i^o$ for individual player $i$ in VD.

As mentioned earlier, all pure Nash equilibria involve free-riders. The designer aims at encouraging all players to volunteer, i.e., the target action profile $a_i^\dagger$ is “volunteer” for any player $i$. Note that $\forall i, \ell_i^o(a^\dagger) = 0$, which lies in the interior of $\mathcal{L} = [-1, 10]$. Therefore, the designer could apply the interior design Algorithm 1. The margin parameter is $\rho = 1$. We let $M = 3$. In table 2, we show the redesigned game



(a) $T-N^T(a^\dagger)$ on TC. (b) $C^T$ on TC. (c) Loss change on TC. (d) $T-N^T(a^\dagger)$ on PD. (e) $C^T$ on PD.

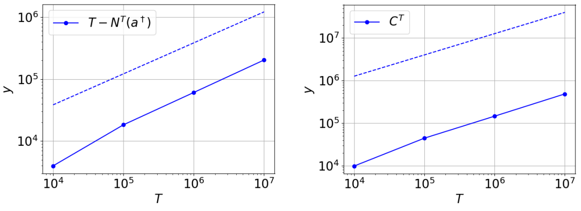
Figure 1: Interior design on TC and PD. The dashed lines are the theoretical upper bound.

$\ell$. Note that when all three players volunteer (i.e., at $a^\dagger$), the loss is unchanged compared to $\ell^o$. Furthermore, regardless of the other players, the action "volunteer" strictly dominates the action "not volunteer" by at least $(1-\frac{1}{M})\rho = \frac{2}{3}$ for every player. When there is no other volunteers, the dominance gap is $\frac{32}{3} \geq (1-\frac{1}{M})\rho$, which is due to the thresholding in (5). We

| | | Number of other volunteers | | |
|---|---|---|---|---|
| | | 0 | 1 | 2 |
| Player $i$ | volunteer | $-2/3$ | $-1/3$ | $0$ |
| | not volunteer | $10$ | $1/3$ | $2/3$ |

Table 2: The redesigned loss function $\ell_i$ for player $i$ in VD.

simulated play for $T = 10^4, 10^5, 10^6, 10^7$, respectively on this redesigned game $\ell$. In Figure 2a, we show $T - N^T(a^\dagger)$ against $T$. The plot is in log scale. The standard deviation estimated from 5 trials is less than 3% of the corresponding value and is hard to see in log-scale plot, thus we do not show that. We also plot our theoretical upper bound in dashed lines for comparison. Note that the theoretical value indeed upper bounds our empirical results. In Figure 2b, we show $C^T$ against $T$. Again, the theoretical upper bound holds. As our theory predicts, for the four $T$'s the designer increasingly enforces $a^\dagger$ in 60%, 82%, 94%, and 98% of the rounds, respectively; The per-round design costs $C^T/T$ decreases at 0.98, 0.44, 0.15, and 0.05, respectively.



(a) Number of rounds $a^t \neq a^\dagger$ grows sublinearly.

(b) The cumulative design cost grows sublinearly too.

Figure 2: Interior design on VD with $M = 3$. The dashed lines are theoretical upper bounds.

### 5.2 Tragedy of the Commons (TC)

Our second example is the tragedy of the commons (TC). There are $M = 2$ farmers who share the same pasture to graze sheep. Each farmer $i$ is allowed to graze at most 15 sheep, i.e., the action space is $\mathcal{A}_i = \{0, 1, ..., 15\}$. The more sheep are grazed, the less well fed they are, and thus less price on market. We assume the price of each sheep is $p(a) = \sqrt{30 - \sum_{i=1}^{2} a_i}$, where $a_i$ is the number of sheep that farmer $i$ grazes. The loss function of farmer $i$ is then $\ell_i^o(a) = -p(a)a_i$, i.e. negating the total price of the sheep that farmer $i$ owns. The Nash equilibrium strategy of this game is that every farmer grazes $a_i^* = 12$ sheep.

It is well-known that this Nash equilibrium is suboptimal. Instead, the designer hopes to maximize social welfare: $p(a)(a_1 + a_2)$, which is achieved when $a_1 + a_2 = 20$. Moreover, to promote equity the designer desires that the two farmers graze the same number of sheep. Thus the target action profile is $a_i^\dagger = 10, \forall i$. Note that the original loss function takes value in $[-15\sqrt{15}, 0]$ while $\ell_i^o(a^\dagger) = -10\sqrt{10}$, thus this is the interior design scenario. Due to the large number of entries, we only visualize the difference $\ell_1(a) - \ell_1^o(a)$ for player 1 in Figure 1c; the other player is the same. We observe three patterns of loss change. For most $a$'s, e.g., $a_1 \leq 6$ or $a_2 \geq 11$, the original loss $\ell_1^o(a)$ is already sufficiently large and satisfies the dominance gap in Lemma 3, thus the loss remains unchanged. For those $a$'s where $a_1^\dagger = 10$, the designer reduces the loss to make the target action more profitable. For those $a$'s close to the bottom left ($a_1 > a_1^\dagger$ and $a_2 \leq 10$), the designer increases the loss to enforce the gap $(1 - \frac{1}{M})\rho$.

We simulated the game play for $T = 10^4, 10^5, 10^6, 10^7$ rounds and show the results in Figure 1a, 1b, and 1c. Again the game redesign is successful: the figures confirm $T - o(T)$ target action play and $o(T)$ cumulative design cost. Numerically, for the four $T$'s the designer enforces $a^\dagger$ in 41%, 77%, 92%, and 98% of rounds, and the per-round design costs are 9.4, 4.2, 1.4, and 0.5, respectively.

### 5.3 Prisoner's Dilemma (PD)

Our third example is the prisoner's dilemma (PD). There are two prisoners, each can stay mum or fink. The original loss $\ell^o$ is given in Table 5a. The Nash equilibrium strategy of this game is that both prisoners fink. Suppose a mafia designer hopes to force $a^\dagger$ =(mum, mum) by sabotaging the losses. Note that $\forall i, \ell_i^o(a^\dagger) = 2$, which lies in the interior of the loss range $\mathcal{L} = [1, 5]$. Therefore, this is again an interior design scenario. In Table 5b we show the redesigned game $\ell$. Note that when both prisoners stay mum or both fink, the designer does not change the loss. However, when one prisoner stays mum and the other finks, the designer reduces the loss for the mum prisoner and increases the loss for the betrayer. We simulated plays for $T = 10^4, 10^5, 10^6$, and $10^7$. In Figure 1d and 1e, we plot the number of non-target selections $T - N^T(a^\dagger)$ and the cumulative design cost $C^T$ for PD. Both grow sublinearly as $T$ increases. The designer enforces $a^\dagger$ in 85%, 94%, 98%, and 99% of rounds. The per-round design costs are 0.71, 0.28, 0.09, and 0.03, respectively.

|  | $R$ | $P$ | $S$ |
|---|---|---|---|
| $R$ | $-0.5, 0.5$ | $0, 0$ | $-0.5, 0.5$ |
| $P$ | $0, 0$ | $0.5, -0.5$ | $0, 0$ |
| $S$ | $0, 0$ | $0.5, -0.5$ | $0, 0$ |

(a) $\ell^t (t = 1)$.

|  | $R$ | $P$ | $S$ |
|---|---|---|---|
| $R$ | $0.62, -0.62$ | $0.75, -0.75$ | $0.62, -0.62$ |
| $P$ | $0.75, -0.75$ | $0.87, -0.87$ | $0.75, -0.75$ |
| $S$ | $0.75, -0.75$ | $0.87, -0.87$ | $0.75, -0.75$ |

(b) $\ell^t (t = 10^3)$.

|  | $R$ | $P$ | $S$ |
|---|---|---|---|
| $R$ | $0.94, -0.94$ | $0.96, -0.96$ | $0.94, -0.94$ |
| $P$ | $0.96, -0.96$ | $0.98, -0.98$ | $0.96, -0.96$ |
| $S$ | $0.96, -0.96$ | $0.98, -0.98$ | $0.96, -0.96$ |

(c) $\ell^t (t = 10^7)$.

Table 3: The redesigned RPS games $\ell^t$ for selected $t$ (with $\epsilon = 0.3$). Note the target entry $a^\dagger = (R, P)$ converges toward $(1, -1)$.

|  | $R$ | $P$ | $S$ |
|---|---|---|---|
| $R$ | $0, 0$ | $1, -1$ | $-1, 1$ |
| $P$ | $-1, 1$ | $0, 0$ | $1, -1$ |
| $S$ | $1, -1$ | $-1, 1$ | $0, 0$ |

(a) The original loss $\ell^o$.

|  | $R$ | $P$ | $S$ |
|---|---|---|---|
| $R$ | $1, 1$ | $1, 1$ | $-1, 1$ |
| $P$ | $-1, -1$ | $1, -1$ | $-1, -1$ |
| $S$ | $-1, 1$ | $-1, -1$ | $-1, -1$ |

(b) $\hat{\ell}^t (t = 1)$.

|  | $R$ | $P$ | $S$ |
|---|---|---|---|
| $R$ | $1, -1$ | $1, 1$ | $-1, -1$ |
| $P$ | $1, -1$ | $1, -1$ | $1, -1$ |
| $S$ | $1, -1$ | $1, -1$ | $1, 1$ |

(c) $\hat{\ell}^t (t = 10^3)$.

|  | $R$ | $P$ | $S$ |
|---|---|---|---|
| $R$ | $1, -1$ | $1, -1$ | $1, -1$ |
| $P$ | $1, -1$ | $1, -1$ | $1, -1$ |
| $S$ | $1, -1$ | $1, -1$ | $1, -1$ |

(d) $\hat{\ell}^t (t = 10^7)$.

Table 4: Instantiation of discrete design on the same games as in Table 3. The redesigned loss lies in $\mathcal{L} = \{-1, 0, 1\}$.

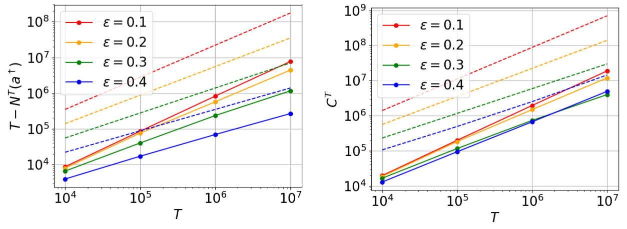
|  | mum | fink |
|---|---|---|
| mum | $2, 2$ | $5, 1$ |
| fink | $1, 5$ | $4, 4$ |

(a) The original loss $\ell^o$ of PD.

|  | mum | fink |
|---|---|---|
| mum | $2, 2$ | $1.5, 2.5$ |
| fink | $2.5, 1.5$ | $4, 4$ |

(b) The redesigned loss $\ell$ of PD.

Table 5: Interior redesign on Prisoner's Dilemma.



(a) Number of rounds $a^t \neq a^\dagger$. (b) The cumulative design cost.

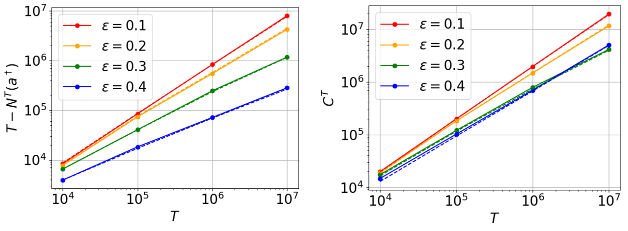
Figure 3: Boundary design on RPS. The dashed lines are the theoretical upper bound.

## 5.4 Rock-Paper-Scissors (RPS)

Our last example is the RPS game, as shown in Table 4a.

**Boundary Design.** Suppose the target profile is $a^\dagger = (R, P)$. Since $\ell^o(a^\dagger) = (1, -1)$ hits the boundary of loss range $\tilde{\mathcal{L}} = [-1, 1]$, the designer must use the boundary design. For simplicity we choose $v$ with $v_i = \frac{U+L}{2}, \forall i$. This choice of $v$ preserves the zero-sum property. Table 3 shows the redesigned games at $t = 1, 10^3$ and $10^7$ under $\epsilon = 0.3$. Note that the designer maintains the zero-sum property of the games. Also note that the redesigned loss function guarantees strict dominance of $a^\dagger$ for all $t$, but the dominance gap decreases as $t$ grows. Finally, the loss of the target action $a^\dagger = (R, P)$ converges to the original loss $\ell^o(a^\dagger) = (1, -1)$ asymptotically, thus the designer incurs diminishing cost.

We ran Algorithm 2 for $\epsilon = 0.1, 0.2, 0.3, 0.4$. For each $\epsilon$ we simulated game play for $T = 10^4, 10^5, 10^6$ and $10^7$. In Figure 3a, we show $T - N^T(a^\dagger)$ under different $\epsilon$ (solid lines) and the theoretical upper bounds of Theorem 8 (dashed lines) for comparison. In Figure 3b, we show $C^T$ and the upper bounds. Note that both $T - N^T(a^\dagger)$ and $C^T$ grow sublinearly. For $\epsilon = 0.3$, for the four $T$'s the designer forces $a^\dagger$ in 34%, 60%, 76%, and 88% rounds. The per-round design costs are 1.7, 1.2, 0.73 and 0.40. The results are similar for the other $\epsilon$'s. We note that empirically the cumulative design cost achieves the minimum at some $\epsilon \in (0.3, 0.4)$ while Theorem 8 suggests the minimum at $\epsilon^* = 0.25$. We investigate this inconsistency in the appendix.



(a) Number of rounds $a^t \neq a^\dagger$. (b) The cumulative design cost.

Figure 4: Discrete redesign for $a^\dagger = (R, P)$ with natural loss values in $\mathcal{L}$. The dashed lines are the corresponding boundary design.

**Discrete Design.** We now compare the performance of discrete design (Algorithm 3) with the boundary design. The target profile is still $a^\dagger = (R, P)$. Recall the purpose of discrete design is to only use natural game loss values, in the RPS case $\mathcal{L} = \{-1, 0, 1\}$. Figure 4 shows that the performance of the discrete design nearly matches the boundary design. When $\epsilon = 0.3$, for the four $T$'s discrete design enforces $a^\dagger$ 35%, 59%,75% and 88% of the time. The per-round design costs are 1.7, 1.2, 0.79, and 0.41. Overall, discrete design does not lose much performance. Table 4 shows the redesigned "random" games at $t = 1, 10^3$ and $10^7$ under $\epsilon = 0.3$. As $t$ increases, the redesigned loss function converges to a constant function that takes the target loss value $\ell^o(a^\dagger)$.

# 6 Conclusion

We studied the game redesign problem where players apply no-regret algorithms to play the game. We show that a designer can enforce a target action profile in $T - o(T)$ rounds while incurring $o(T)$ cumulative design cost. Experiments demonstrate the performance of our redesign algorithms.

## Acknowledgments

Project supported in part by NSF grants 1545481, 1704117, 1836978, 2041428, 2023239, ARO MURI W911NF2110317, and AF CoE FA9550-18-1-0166.

## References

[BLUM, 2007] A BLUM. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*, pages 79–102, 2007.

[Bogunovic *et al.*, 2021] Ilija Bogunovic, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 991–999. PMLR, 2021.

[Bubeck and Cesa-Bianchi, 2012] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[Garcelon *et al.*, 2020] Evrard Garcelon, Baptiste Roziere, Laurent Meunier, Olivier Teytaud, Alessandro Lazaric, and Matteo Pirotta. Adversarial attacks on linear contextual bandits. *arXiv preprint arXiv:2002.03839*, 2020.

[Guan *et al.*, 2020] Ziwei Guan, Kaiyi Ji, Donald J Bucci Jr, Timothy Y Hu, Joseph Palombo, Michael Liston, and Yingbin Liang. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. *arXiv preprint arXiv:2002.07214*, 2020.

[Hart and Mas-Colell, 2000] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[Huang and Zhu, 2019] Yunhan Huang and Quanyan Zhu. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *International Conference on Decision and Game Theory for Security*, pages 217–237. Springer, 2019.

[Jun *et al.*, 2018] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Xiaojin Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3640–3649, 2018.

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

[Lu *et al.*, 2021] Shiyin Lu, Guanghui Wang, and Lijun Zhang. Stochastic graphical bandits with adversarial corruptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8749–8757, 2021.

[Ma *et al.*, 2018] Yuzhe Ma, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, pages 186–204. Springer, 2018.

[Ma *et al.*, 2019] Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Xiaojin Zhu. Policy poisoning in batch reinforcement learning and control. In *Advances in Neural Information Processing Systems*, pages 14570–14580, 2019.

[Monderer and Tennenholtz, 2004] Dov Monderer and Moshe Tennenholtz. k-implementation. *Journal of Artificial Intelligence Research*, 21:37–62, 2004.

[Rakhsha *et al.*, 2020] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. *arXiv preprint arXiv:2003.12909*, 2020.

[Slivkins, 2019] Aleksandrs Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.

[Sun *et al.*, 2020] Yanchao Sun, Da Huo, and Furong Huang. Vulnerability-aware poisoning mechanism for online rl with unknown dynamics. *arXiv preprint arXiv:2009.00774*, 2020.

[Yang *et al.*, 2021] Lin Yang, Mohammad Hajiesmaili, Mohammad Sadegh Talebi, John Lui, and Wing Shing Wong. Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[Zhang *et al.*, 2020] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. *arXiv preprint arXiv:2003.12613*, 2020.

[Zuo, 2020] Shiliang Zuo. Near optimal adversarial attack on ucb bandits. *arXiv preprint arXiv:2008.09312*, 2020.