The Influencers' War

Yancheng Zhu
zhu436@wisc.edu

1 Abstract

This technical report presents an ongoing research project at UW-Madison that employs a game-theoretic approach to analyze the behaviors of social media influencers in various game settings. The primary objective is to design a mechanism that incentivizes these influencers to provide unbiased reports to their audience. By modeling influencers and broadcasting companies as players in a general-sum game, the study aims to minimize biases in the content produced. The report delves into concepts such as Nash Equilibrium and best-response dynamics, illustrating how these can be applied to achieve a pure Nash Equilibrium in potential games. Empirical results are provided to demonstrate the effects of different parameters on players' decisions. Additionally, the report explores a mechanism with punishment to deter cheating behaviors among influencers, ensuring more truthful reporting. The findings contribute to the broader understanding of strategic interactions in digital media and offer insights into promoting unbiased information dissemination.

2 Introduction

In today's digital landscape, news travels swiftly around the world. Countless content creators share their perspectives with the public, ranging from major broadcast networks to news agencies. However, due to diverse interests, the manner in which the same event is reported can differ significantly from one agency to another. In our project, we consider broadcasting companies and social media influencers as players in a general-sum game. Our goal is to design a truthful mechanism that minimizes the biases present in the content they produce.

3 Nash Equilibrium and Best-Response Dynamics

A *Nash equilibrium (NE)* in a game is a situation where no player can increase their utility by unilaterally deviating from their current strategy, denoted by s. Here, a strategy can be either pure, meaning it is an action, usually represented by a, or mixed, meaning it is a probability distribution over several actions. When all players are using pure strategies, the NE it is called a *pure NE (PNE)*, otherwise, it is named a *mixed NE (MNE)*.

Best-response dynamics is a procedure to find a PNE in a game. Suppose there are n players, and the action set of player i is $\mathcal{A}_i$, then the procedure works as follows:

- Initialize $a = (a_1, \ldots, a_n)$ arbitrarily.
- While $\exists i \in [n]$ s.t. $a_i \notin \arg\max_{\tilde{a}_i \in \mathcal{A}_i} u_i(\tilde{a}_i, a_{-i})$.
  - $a'_i \leftarrow \arg\max_{\bar{a}_i \in \mathcal{A}_i} u_i(\bar{a}_i, a_{-i})$.
  - $a \leftarrow (a'_i, a_{-i})$.

Best-response dynamics does not always converge, because a game may not have a PNE.

4 The Motivating Example

This is a naive motivating example. Here we consider the most basic setting. There are n articles $x_1, \dots, x_n \in \mathbb{R}^d$. There are two players, Big and Small, with target locations t_B and t_S, respectively. Each of them will simultaneously choose an article from the article set to influence a victim's opinion about a certain event. The victim learns the event by averaging their reports. If Big chooses x_B and Small chooses x_S, then the victim forms an opinion v, and

$$v = \frac{x_B + x_S}{2}.$$

Big will want to minimize

$$||v - t_B||^2.$$

Similarly, Small wishes to minimize

$$||v - t_S||^2.$$

The game here can be extended to having n players with targets $\mathbf{t}_1, \dots, \mathbf{t}_n \in \mathbb{R}^d$. They report $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$, and the victim will assign scalar weights $w_1, \dots, w_n$ to these reports and choose

$$v = \sum_{i=1}^{n} w_i \mathbf{x}_i.$$

In general, a general-sum game may not have pure NE, but our game here is a potential game with the potential function being

$$\psi(\mathbf{x}) = ||\sum_{i=1}^{n} w_i \mathbf{x}_i||^2 - 2\sum_{i=1}^{n} w_i \mathbf{x}_i^T \mathbf{t}_i.$$

Hence, by running best-response dynamics, we can get a pure NE.

5 Weighted Reports

We consider a slightly more sophisticated cognitive model than simply taking the average of Big and Small. Our victim has become more rational; it does not want to accept information that it considers as too extreme or biased. Moreover, it also has its own idea $v \in \mathbb{R}^d$, before the game starts.

In the new setting, the victim first chooses natural numbers σ_B and σ_S as standard deviations. Given reports $x_B \in \mathbb{R}^d$ and $x_S \in \mathbb{R}^d$ from Big and Small, the victim calculates these *trust factors*:

$$g_B = e^{-\frac{||x_B - v||^2}{2\sigma_B^2}}$$
$$g_S = e^{-\frac{||x_S - v||^2}{2\sigma_S^2}}.$$

Here, g_B and g_S represent "how much victims trusts reports coming from the two agents", respectively. The victim updates v by doing the following calculation,

$$v = \frac{v + g_B x_B + g_S x_S}{1 + g_B + g_S}.$$

When an agent reports something that is too far away from the current v, its trust factor will become small, and its report thus has less importance to the victim's new opinion.

If there are n players, then

$$g_i = e^{-\frac{||x_i - v||^2}{2\sigma_i^2}},$$

and

$$v = v = \frac{v + \sum_{i=1}^{n} g_i x_i}{1 + \sum_{j=1}^{n} g_j}.$$

It is not clear If the game is a potential game or not. But when there are 2 players and $d = 1$, best-response dynamics does converge.

5.1 Empirical Results

We let both players (Big and Small) use Gaussian Weighting and run BRD several times; then we analyze how different parameters can affect two players' choices.

5.1.1 Standard Deviation

In this section, we analyze the effects of different σ on the two players' decisions by fixing v, t and only changing σ.

We first let these 2 players share the same σ, and set $v = 0$, $t_B = 50$, $t_S = -50$. Assume both players start looking for their best responses at v, i.e., they both place their first points at the victim's belief. Suppose $\sigma_B = \sigma_S = 10$, we now analyze the best response for Big. We plot the loss function and the updated belief positions based on Big's choices, see Figure 1.

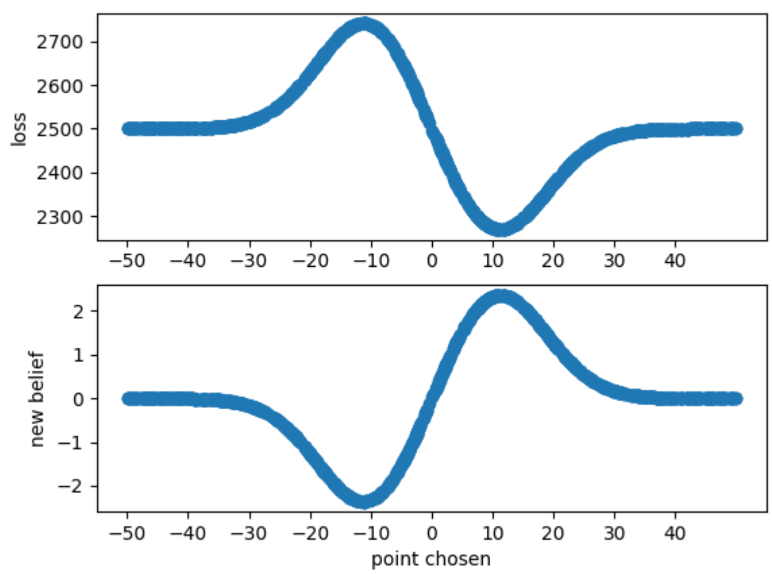

Figure 1: Loss function and updated belief of Big's choices in the first iteration

From the graph, we can see that the optimal x_B is slightly larger than 10; its exact numerical value is 11.25. Since these two players' targets are symmetric around the y-axis, intuitively Small should pick the point with x-coordinate equals -11.25, and this accords with our computation. For the second iteration, Big will respond to the point picked by Small in the first round.

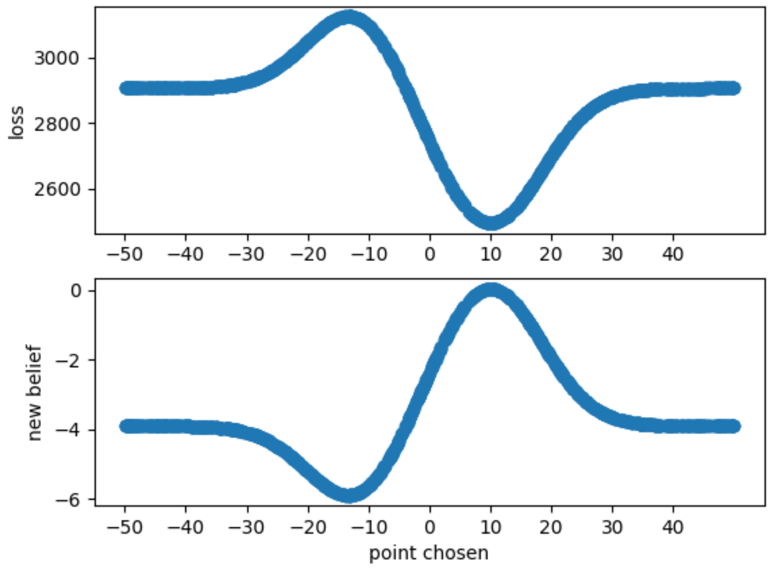

Figure 2: Loss function and updated belief of Big's choices in the second iteration

In this iteration, Big chooses 10.021 as its response; as indicated by Figure 2, we can know that this is indeed its best response. By symmetry, the other player will choose its additive inverse. This concludes iteration 2.

We repeat this process, until both players do not want to change their points anymore. The whole development of this game can be found in Figure

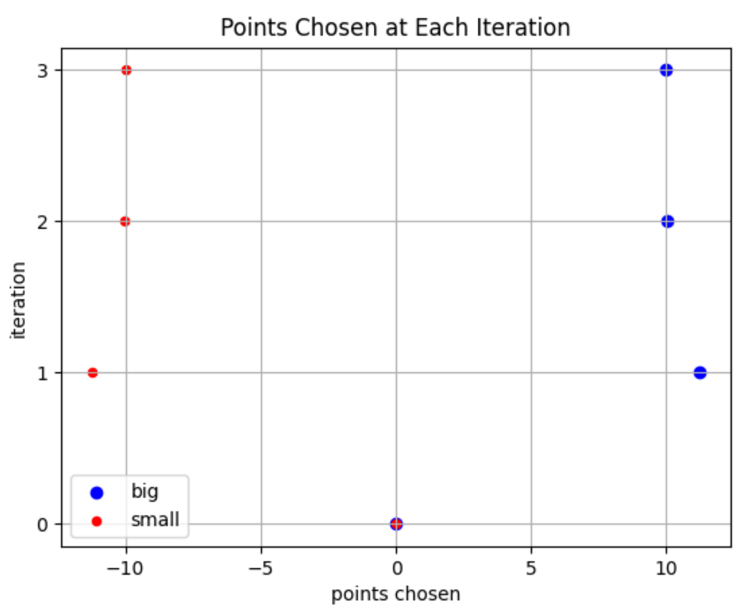

Figure 3: Game development when $\sigma_1 = \sigma_2 = 10$

We repeat this process, until both players do not want to change their points anymore. The whole development of this game can be found in Figure 3.

We repeat this setup with different σ values, the result is shown in Table 1.

t_B	t_S	v (when game starts)	$\sigma_B,\ \sigma_S$	x_B^*	x_S^*	number of iterations
50.0	-50.0	0.0	1	1.0	-1.0	3
50.0	-50.0	0.0	5	5.0	-5.0	3
50.0	-50.0	0.0	10	10.0	-10.0	3
50.0	-50.0	0.0	15.7	15.7	-15.7	3
50.0	-50.0	0.0	50	50.0	-50.0	3
50.0	-50.0	0.0	100	100.0	-100.0	3

Table 1: Experiments with different σ

We can also let $\sigma_B \neq \sigma_S$. Consider the case where $\sigma_B = 5,\ \sigma_S = 10,\ x_0 = x_1 = x_2 = 0.0, t_B = 50.0, t_S = -50.0$. We analyze Big's and Small's best responses during the first iteration by drawing the loss functions, see Figure 4 and Figure 5.

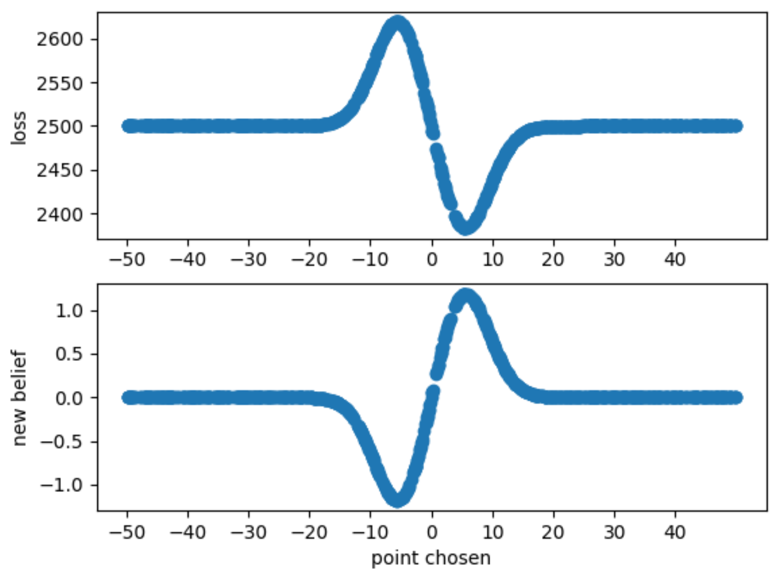

Figure 4: Big's loss function

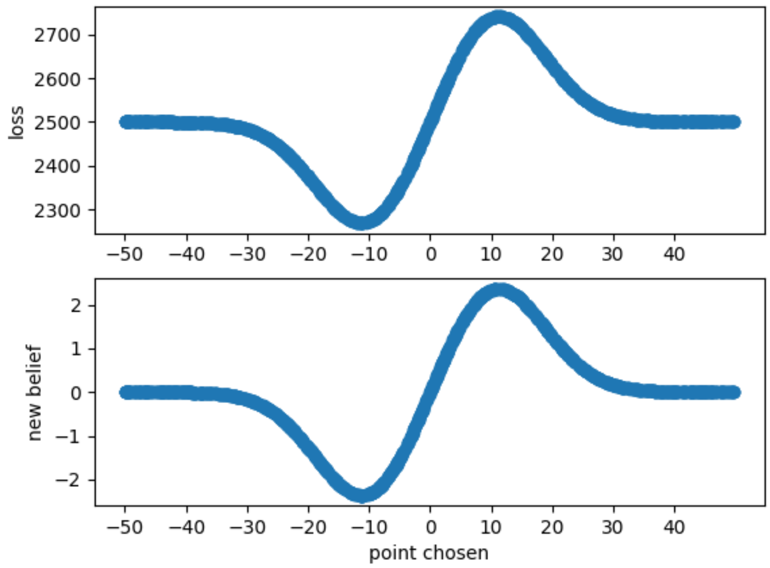

Figure 5: Small's loss function

In order to maximize utility, Big chooses $x_B = 5.62$, Small chooses $x_S = -11.25$ within this iteration. Figure 6 shows the situation for the second iteration. In this iteration, $x_B = 4.37,\ x_S = -10.75$.

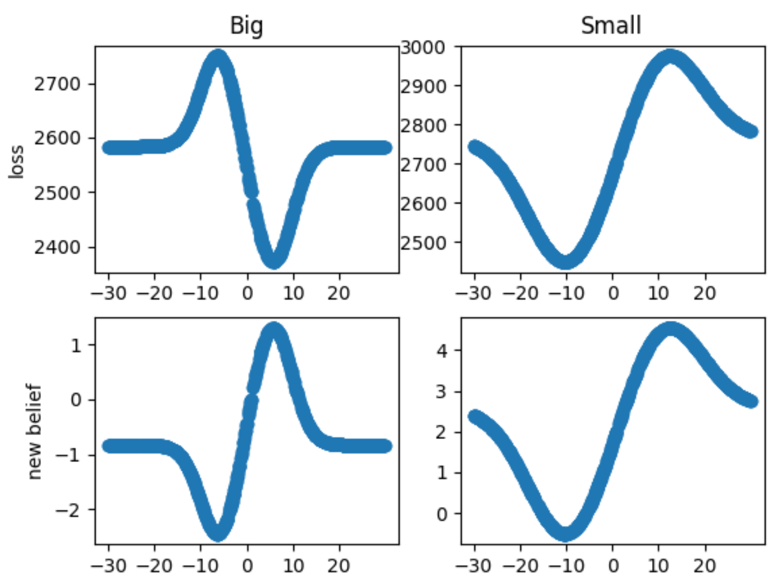

Figure 6: Loss functions in the second Iteration

Table 2 summarizes this game.

x_B	x_S	v	Iteration
0.0	0.0	0.0	0
5.62	-11.25	-1.45	1
4.37	-10.75	-1.36	2
4.366	-10.70	-1.36	3

Table 2

We repeat the experiment with fixed σ_B and different σ_S, and document the results in Table 3.

t_B	t_S	v (when game starts)	σ_B	σ_S	x_B^*	x_S^*	number of iterations
50.0	-50.0	0.0	5	10	4.366	-10.70	3
50.0	-50.0	0.0	5	20	3.39	-22.09	3
50.0	-50.0	0.0	5	50	1.88	-56.03	3
50.0	-50.0	0.0	5	100	1.02	-112.38	3

Table 3: Experiments with different σ_S

5.1.2 Victim Belief

In this section, we study the behaviors of the two players with fixed targets and different victim beliefs. We first consider a special case, where $\sigma_B = \sigma_S = 10,\ t_B = 10,\ t_S = -10,\ v = 10$. Now, in order to minimize the loss, Big can choose to either place its point at 10 or put it as far away from 10 as possible.

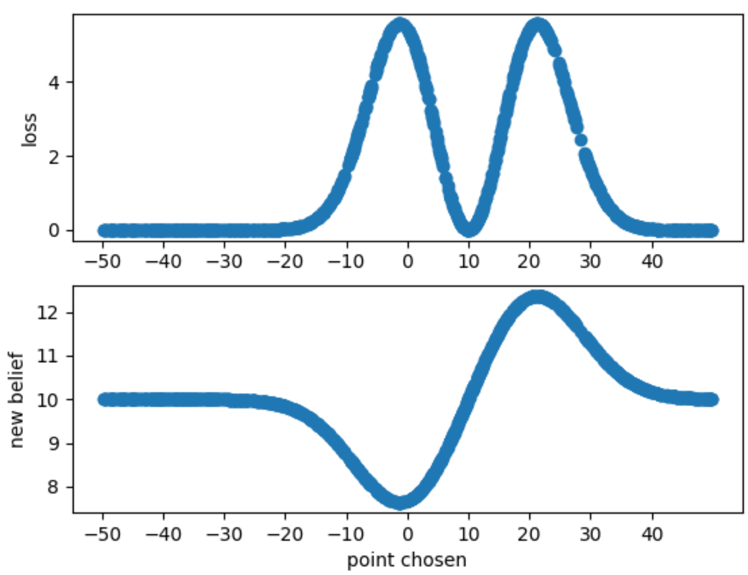

Figure 7: Loss function of Big when $v = t_B = 10$

Things are more straightforward for Small; it will prefer to place its point near the origin to drag the victim as close to t_S as possible. Calculation indicates that Small should move x_S to -1.25. This accords with the graph of its loss function, Figure 8.

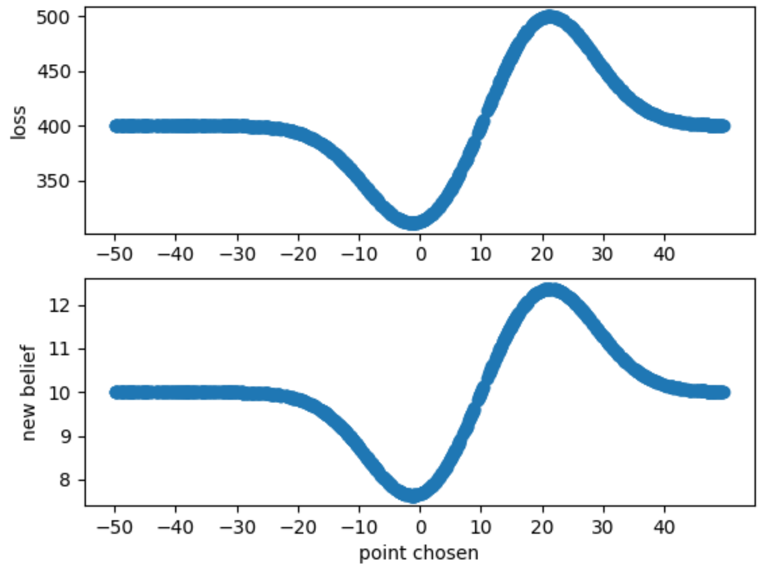

Figure 8: Small's loss function

The whole game is presented in Figure 9

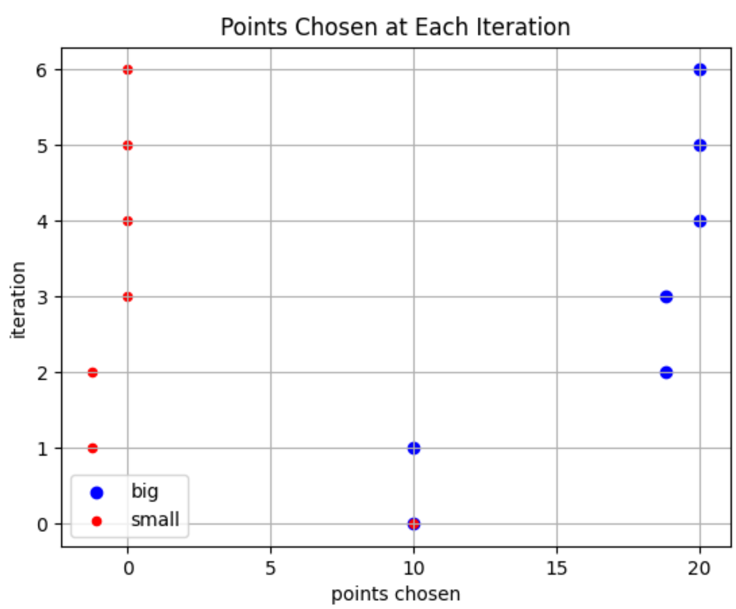

Figure 9

We set $t_B = 50,\ t_S = -50, \sigma_B = \sigma_S = 10$, and run the simulation with different initial v for several rounds. The results are recorded in Table 4.

t_B	t_S	v (when the game starts)	$\sigma_B,\ \sigma_S$	x_B^*	x_S^*	number of iterations
50.0	-50.0	0.0	10	10.0	-10.0	3
50.0	-50.0	-5.0	10	5.0	-15.0	3
50.0	-50.0	10.0	10	20.0	0.0	3
50.0	-50.0	-20.0	10.0	-10.0	-30.0	3
50.0	-50.0	50.0	10	60.0	40.0	6
50.0	-50.0	-70.0	10	-56.549	-56.549	3

Table 4: Experiments with different v

6 Mechanism With Punishment

6.1 Problem Setting

Let us revisit the naive model where the victim simply takes the average of influencers' reports and give it a small upgrade. There are n players each wants the victim to be close to targets $t_1 \dots t_n \in \mathbb{R}$, and these targets are common knowledge.

- $\forall i \in [n]$: players i observes one random number from nature: $z_i \sim N(\mu, \sigma^2)$. z_i is private to player i and hence can be viewed as its type.
- $\forall i \in [n]$: player i chooses a reporting function f_i, and reports $x_i = f_i(z_i, t_{1:n})$.
- The victim receives $x_{1:n}$, and estimate μ by $v = g(x_{1:n}, t_{1:n})$.
- The victim suffers penalty $(v - \mu)^2$; $\forall i \in [n]$: player i suffers penalty $(v - t_i)^2$.

In order to influence the victim, a player may produce a report that is not close to its z, and thus harms the victim's interest. In order to eliminate this kind of cheating behavior, the victim has to design the g carefully.

6.2 A Special Case

Consider the case when $\sigma = 0$, i.e., all players get μ from nature. We propose a mechanism that makes everyone reporting μ a NE:

- The victim computes $\hat{x} = \frac{\sum_{i=1}^n x_i}{n}$.
- $\forall i \in [n]$, the victim calculates $\hat{x}_{-i} = \frac{\sum_{j \neq i} x_j}{n-1}$.
- The victim determines $i^* = \text{argmax}_i (\hat{x}_{-i} - x_i) \dfrac{\hat{x}_{-i} - t_i}{|\hat{x}_{-i} - t_i|}$.
- The victim chooses $\epsilon \in \mathbb{N}$ and uses $v = g(x_{1:n}, t_{1:n}) = \hat{x} + \max\{0,\ |\hat{x}_{-i^*} - x_{i^*}| \dfrac{\hat{x}_{-i^*} - t_{i^*}}{|\hat{x}_{-i^*} - t_{i^*}|}\}\epsilon$ to estimate μ.
- The victim suffers $(\mu - v)^2$, and player i suffers $(v - t_i)^2$ for all i.

The idea is that the victim first finds the player that cheats the most and then moves v in the direction that increases that cheater's cost. When $\epsilon > \frac{1}{n}$, all players reporting μ is a *strict NE*; every player will be worse-off by cheating. In fact, when $\epsilon > \frac{1}{n}$, all players reporting any point together is a strict NE.

Proof. This is a proof sketch. Suppose the other players are reporting $x_{-i} = \hat{x}_{-i} = \mu$. If player i reports $x_i = \mu$, the victim produces $v = \mu$, and player i suffers $(\mu - t_i)^2$. If player i reports x_i such that

$$(\hat{x}_{-i} - x_i) \frac{\hat{x}_{-i} - t_i}{|\hat{x}_{-i} - t_i|} \leqslant 0.$$

Then

$$\begin{aligned}(v - t_i)^2 &= [(1 - \frac{1}{n})\mu + \frac{x_i}{n} - t_i]^2 \\ &= [\mu - t_i + \frac{1}{n}(x_i - \mu)]^2 \\ &> (\mu - t_i)^2.\end{aligned}$$

If player i reports x_i such that

$$(\hat{x}_{-i} - x_i)\frac{\hat{x}_{-i} - t_i}{|\hat{x}_{-i} - t_i|} > 0.$$

Then

$$\begin{aligned}(v - t_i)^2 &= [(1 - \frac{1}{n})\mu + \frac{x_i}{n} + |\hat{x}_{-i} - x_i|\frac{\hat{x}_{-i} - t_i}{|\hat{x}_{-i} - t_i|}\epsilon - t_i]^2 \\ &> (\mu - t_i)^2.\end{aligned}$$

□

6.3 The General Case

When players receive different z values, the game becomes unstable. The game can converge at a state where influencers try to cheat, but not that much so that they do not become the biggest cheater. Here we show some BRD simulations with $\epsilon = \frac{1.5}{n}$:

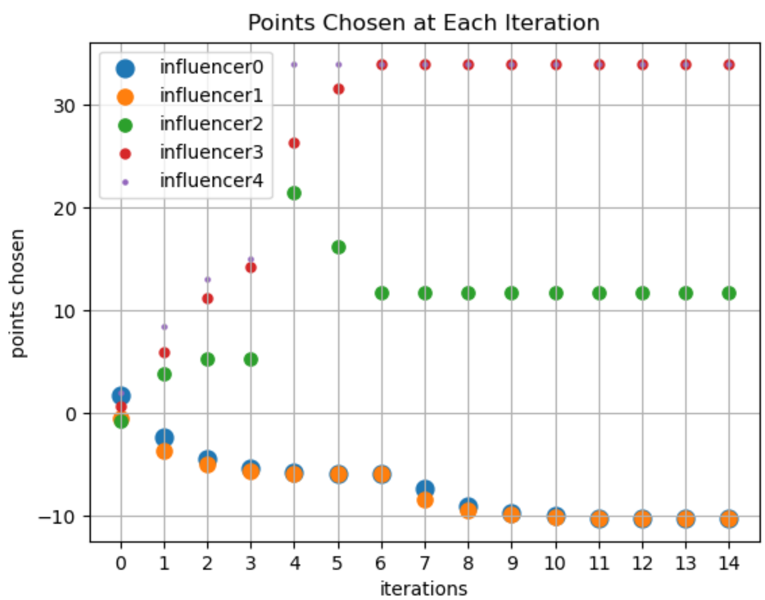

Figure 10: The game with $N(1, 1)$ as the distribution, and agents' targets at -30, -10, 5, 20, 40

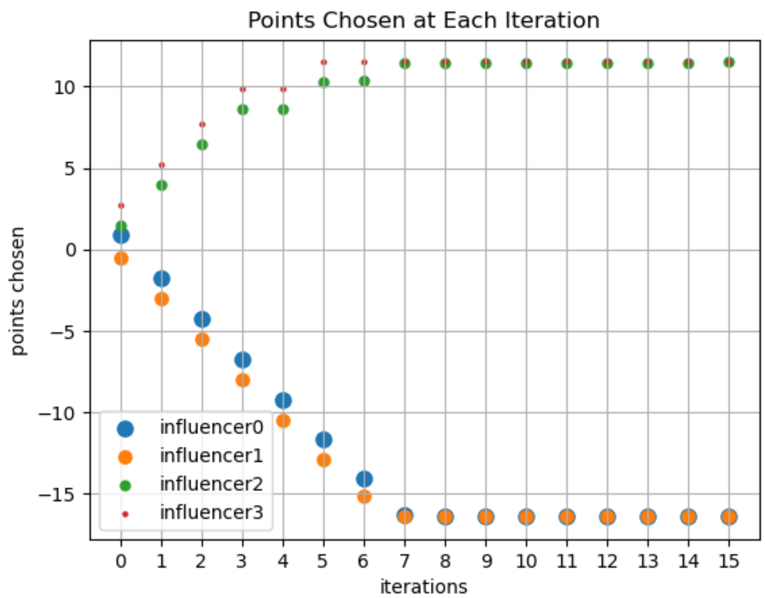

Figure 11: The game with $N(1,1)$ as the distribution, and agents' targets at -30, -10, 5, 20

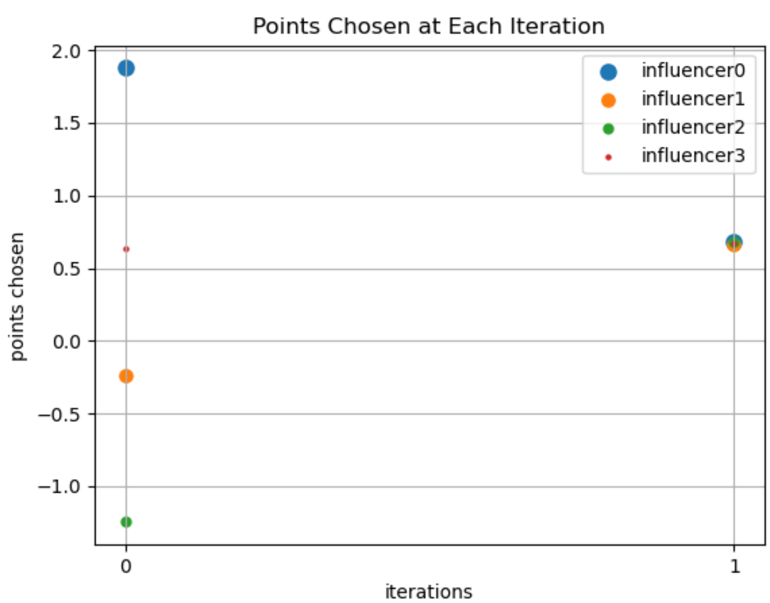

Figure 12: The game with $N(1,1)$ as the distribution, and agents‘ targets all at 5

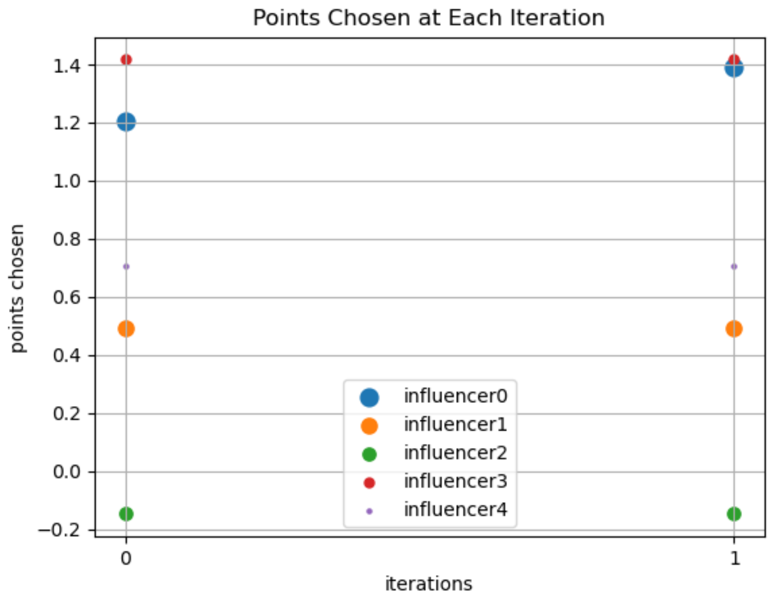

Figure 13: The game with $N(1, 1)$ as the distribution, and 5 agents' targets all at 1

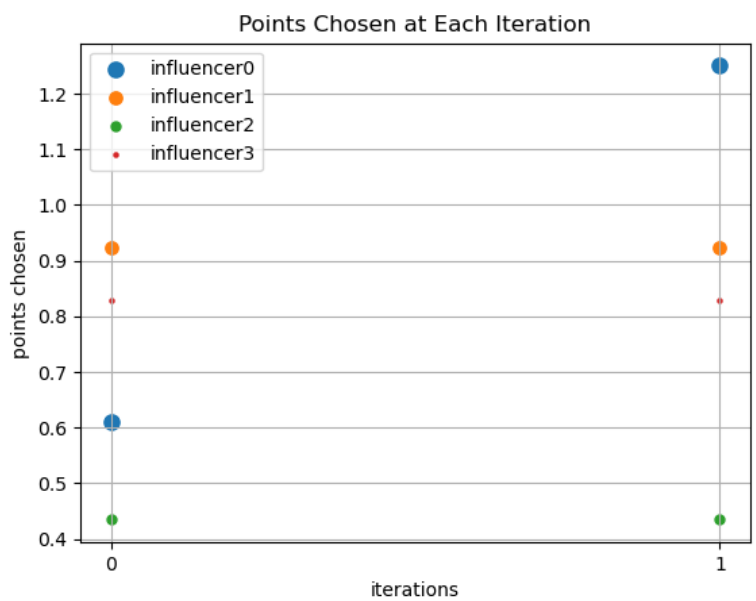

Figure 14: The game with $N(1, 1)$ as the distribution, and 4 agents' targets all at 1