Multiplayer Modeling via Multi-Armed Bandits

Robert C. Gray
College of Media Arts & Design
Drexel University
Philadelphia, PA, USA
robert.c.gray@drexel.edu

Jichen Zhu
IT University of Copenhagen
Copenhagen, Denmark
jichen.zhu@gmail.com

Santiago Ontañón*
College of Computing & Informatics
Drexel University
Philadelphia, PA, USA
so367@drexel.edu

***Abstract*—This paper focuses on *player modeling* in multiplayer adaptive games. While player modeling has received a significant amount of attention, less is known about how to use player modeling in multiplayer games, especially when an experience management AI must make decisions on how to adapt the experience for the group as a whole. Specifically, we present a multi-armed bandit (MAB) approach for modeling groups of multiple players. Our main contributions are a new MAB framework for multiplayer modeling and techniques for addressing the new challenges introduced by the multiplayer context, extending previous work on MAB-based player modeling to account for new group-generated phenomena not present in single-user models. We evaluate our approach via simulation of virtual players in the context of multiplayer adaptive exergames.**

***Index Terms*—multi-armed bandit, multiplayer modeling, machine learning, linear regression, reinforcement learning**

I. Introduction

Multi-armed bandits (MABs, or "bandits") are a class of sequential decision problem in which an agent must make a selection from a group of options repeatedly, observing rewards resulting from its choices and aiming to maximize the total reward over the course of the selections [1], [2]. MAB techniques can assist in addressing the *exploration/exploitation problem*, where each selection must consider the utility of *exploring* the options to gain information about their potential rewards versus *exploiting* the option currently believed to be the best. When deployed as the experience management (EM) agent in an adaptive game, bandits have been shown to be effective at modeling individuals based on their behavior [3], and an MAB-based AI can serve as the basis for effective interventions and game adaptation. However, previous research on MAB-based player modeling (and player modeling at large) has focused on adapting experiences for individuals rather than groups of players.

In this paper, we explore the use of MAB-based models in multiplayer environments, where such environments introduce new design challenges over single-player experiences. Specifically, we identify three main challenges in MAB-based multiplayer modeling: *best-choice estimation*, *exploration strategies*, and *social fairness*.

To illustrate these three challenges, we consider a scenario in which the AI has some set of intervention options it can apply to the environment and wishes to maximize some target metric for each player (e.g., amount of time playing). In a single-player scenario, the best choice is simply the intervention that is predicted to result in the highest value for this metric. However, in a multiplayer scenario, the AI must contend with multiple measurements of the target metric from multiple players, and there may not be a single intervention that is the best for all players. Moreover, the choice of intervention may not necessarily correlate directly with player experience due to dynamics among players, adding complexity to the prediction of outcomes for each intervention. Additionally, MABs often *explore* by choosing interventions different from the one believed to be the best in order to better understand their potential; however, typical exploration strategies in MABs do not necessarily apply to multiplayer settings due to player interactions, as we will elaborate later. Thus, this prompts the following questions: how do we best leverage multiple measurements from multiple players to make the most accurate predictions when determining the *best choice*? How do we ensure that our exploration considers an exploration space that more accurately reflects the *players' experience*? Finally, how do we maintain *social fairness* over time with repeated selections?

We describe each of these problems in more detail in Section IV. Solving these issues will be essential for the design of MABs that aim to model group-based and social phenomena in players. In this paper, we propose solutions to the first two challenges and leave the third (social fairness) for future work.

Therefore, this paper has three main contributions. The first is a framework for handling the additional complexity of modeling multiple players in MAB-based approaches. Our second contribution is a solution for managing the effect that multiple rewards and predictions have on the bandit strategy's assessment of what the "best" choice means. Finally, our third contribution is a first solution for handling exploration in multiplayer scenarios, where we present a modified version of *forced exploration* that considers not only the prior bandit selections but also the previous experiences administered to the players through those selections.

The remainder of this paper is structured as follows. First, we discuss existing literature related to this research space. Second, we introduce our motivating scenario in which we aim to model player behavior related to social comparison in an adaptive multiplayer game that encourages physical activity

Work partially funded by the NSF (Grant Number: IIS-1816470)

*Currently at Google

978-1-6654-3886-5/21/$31.00 ©2021 IEEE

(PA). Next, we construct a simulation of this scenario with virtual players and an MAB-based model for evaluating our approach. We then discuss our approach in detail, describing both our Multiplayer Regression Oracle (MRO) for multiplayer bandits as well as our Multiplayer Forced Exploration (MFE) algorithm. Finally, we evaluate the performance of our approach in the simulated environment.

II. Background and Related Work

The following introduces related work around multi-armed bandits and player modeling.

A. Multi-Armed Bandits

A multi-armed bandit (MAB) problem is a decision problem [1], [2] where an agent is faced with many opportunities, each with an unknown potential reward (ρ) from which it must make a selection. This decision is repeatedly requested of the agent, usually over a limited number of iterations (in practical cases) known as the *horizon* of the agent's operation. In keeping with an analogy of playing slot machines in a casino, the agent is tasked with *pulling* one of the *arms* each iteration until the horizon is reached.

In the most common formulation, the *stochastic bandit problem*, the rewards offered by each of these arms are modeled as independent, static distributions of potential reward values, only discoverable by testing the arms repeatedly. The agent's goal is to maximize the total value of rewards observed over the course of the horizon, where each pull must consider the opportunity cost of *exploiting* the agent's current knowledge of the reward space to maximize return versus *exploring* among the arms to gain a more accurate understanding of the reward space. The agent's design can be conceptually divided into a *policy* that evaluates this trade-off to guide exploration pulls and an *oracle* that predicts future rewards to maximize exploitation pulls [4].

Variants of the MAB problem add features and constraints that can better align the MAB problem and its techniques with real-world problems. Relevant to the ideas discussed in this research, *contextual bandits* introduce an external context vector that describes additional information regarding the bandit's environment, potentially offering insight to the agent [5]. *Short-horizon bandits* explore situations where the agent is given very few opportunities to pull arms [4], and *restless bandits* approach scenarios in which not every arm is available on every pull [6].

B. Player Modeling

Player modeling refers to the practice of inferring and tracking individual characteristics of a player based on observations of player qualities or behavior [7]. Adaptive games often leverage player modeling to dynamically adjust difficulty, manage user interface, or guide game narrative based on player preference, performance, or behavior [8]–[12]. In this way, player modeling can assist in managing the experience of a player in a software environment by providing an *experience management* (EM) AI agent with necessary insight into the player's needs so it can perform effective interventions on the environment [13].

Multiplayer modeling, or the modeling of more than one player simultaneously, is an understudied field that introduces significant challenges over individual modeling [14]. When a virtual environment is shared by multiple players and singular EM game adaptations must affect those players simultaneously, additional concerns emerge for the AI. For instance, AI predictions must incorporate multiple models, and when an AI intervention benefits one player but negatively affects another, the assessment of that intervention's value must be reconsidered. Additionally, the dynamics among players introduce new phenomena not present in single-player environments (e.g., interpersonal skills, leadership, jealousy, etc.) that may provide additional opportunities for modeling.

MABs have been shown to be effective in establishing player models based on player behavior [3], [4]. When used as the basis for an EM AI, an MAB strategy is distinctly capable of exploring and exploiting a user's response to various intervention states while maximizing reward over time. In this kind of MAB-based modeling approach where arms are associated with player characteristics, the method becomes the model [3], and the arm predictions provided by the MAB work both to guide the intervention and to describe the system's estimation of the player's underlying preferences or traits.

III. Motivating Scenario

Our primary interest in multiplayer modeling is the opportunity to investigate phenomena that manifest only in group environments. For example, while traditional single-player scenarios might enable modeling of individual traits such as player preferences or skill, multiplayer environments yield opportunities to model social dynamics such as teamwork, peer pressure, and so on. The following discusses our domain of interest and the scenario we have devised to evaluate our multiplayer modeling approach.

A. Social Comparison Orientation

In this paper, we consider the phenomenon of *social comparison*, the psychological processes by which individuals compare themselves to others [15]. Social comparisons are made by individuals, often subconsciously, to evaluate their own performance (self-evaluation), to gather insight toward their future success (self-improvement), or to improve their self-confidence (self-enhancement) [16]. Current models hold that individual characteristics, known as an individual's *social comparison orientation* (SCO) [17], [18], will determine their frequency, purpose, and emotional reactions around these comparisons. For example, some individuals may tend to seek out (or *prefer*) comparisons to others doing better than them (*upward comparisons*) for the purpose of self-improvement, while some may prefer to seek out others doing worse than they are (*downward comparisons*) for the purpose of finding relief in their comparatively better performance.

In this research, we construct predictive models of player SCO based on their behavior when introduced to various comparison opportunities (i.e., upward and downward) in the domain of exercise games (or *exergames*) that aim to encourage physical activity (PA) in players. Specifically, we construct a scenario in which players are encouraged to walk more by measuring a player's daily steps via a pedometer (e.g., a Fitbit device) and motivating them through a team-focused game activity based around that measurement.

B. Multiplayer Experience Management Setting

In the experiments reported in this paper, we used a simulated environment with simulated players. In our scenario, to provide for social comparison opportunities, players are placed into teams where they regularly (i.e., daily) engage in a software activity that reports both a player's own step performance and that of their teammate. As an intervention, we construct an MAB-based AI that generates the step count of a third teammate, one that is not known to the other two teammates to be artificially constructed.

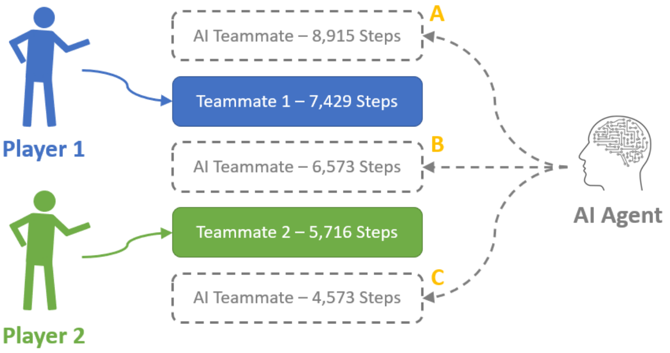
The two *participant teammates* (as virtual players, discussed below) are both presented with the step counts of all members of their team and are prompted to select one of them to gain more insight into that teammate's profile and performance details. This promotes the comparison activity, where both the profile selection and the resulting motivating effect are determined by the virtual player's modeled SCO.

The steps presented by the third *AI teammate* constructed by the AI provides additional comparison opportunities to both players; in any daily session, the steps of the fabricated AI teammate could be positioned above, below, or between those of the participant teammates to provide additional upward and downward comparisons relative to each. In this way, the positioning of the AI's reported steps becomes the experience management lever through which the AI can increase or decrease the upward and downward comparison opportunities. As illustrated in Figure 1, through this lever, the AI can adapt the overall experience toward what it believes will provide the most effective social comparison-based motivation.

C. Restless Bandit Design

Our proposed intervention is carried out by an MAB-based AI agent that considers potential AI teammate step positions as arms corresponding to the placements discussed in the previous section (i.e., above, below, or between the steps of the two participant players). The participant teammates' steps can vary, yielding days in which one human teammate ("P1") may have higher steps than the other ("P2") and days in which the opposite is true. In the case where the AI teammate's steps would be placed between those of the two participant teammates, we recognize that this differentiation may be significant to the effect of that arm; therefore, we choose to design a *restless bandit* [6] that conditionally activates and deactivates arms based on which of the two participant teammates has more steps.

Specifically, we construct our restless MAB with four arms that will determine the placement of the AI teammate's steps within a session. The arms are as follows:



A
AI Teammate – 8,915 Steps
Teammate 1 – 7,429 Steps
Player 1
B
AI Teammate – 6,573 Steps
AI Agent
Teammate 2 – 5,716 Steps
C
AI Teammate – 4,573 Steps
Player 2

Fig. 1. Two participant teammates remotely engage in a software activity each day in which both teammates' steps from the previous day are presented. A third teammate, not known to the participants to be artificially constructed, is added to the team by the AI. The MAB determines the placement of the AI teammate's steps relative to the steps of the participants (option A, B, or C), providing additional comparison opportunities that can leverage the SCO preferences modeled for both participants.

- **A**: The AI teammate's steps are set 20% above the higher of the two participant teammates.
- **B1**: In the case where P1's steps are higher than those of P2, the AI teammate's steps are set to the midpoint between the two participant teammates' steps. This arm is not available if P2's steps are higher than those of P1.
- **B2**: In the case where P2's steps are higher than those of P1, the AI teammate's steps are set to the midpoint between the two participant teammates' steps. This arm is not available if P1's steps are higher than or equal to those of P2.
- **C**: The AI teammate's steps are set 20% below the lower of the two participant teammates.

D. Virtual Player Design

In this research, we perform simulations with virtual players to explore and optimize our approach, similar to our previous bandit-based modeling research [3]. In pursuing a more accurate simulation, we construct these virtual players to exhibit behavior that would reflect what would be expected from real human participants. For instance, we implement a simulator for generating human-like daily steps for our virtual players.

In an analysis of the Mechanical Turk step data in a 2016 study by Furberg et al. [19], the daily step data ($n = 1665$) was found to conform to the probability density function of a gamma distribution $\Gamma(k = 2.8, \theta = 3100)$ [3]. We also discovered associations between a person's steps on a given day (S_t) and their steps in the days of the prior week. We therefore constructed a regression model for these records that included the observed steps for that participant in each of the previous seven days ($S_{t-1}, ..., S_{t-7}$), performing backward elimination to determine the features that were statistically significant ($p < 0.05$) in predicting S_t. All features survived this process except the steps from five days prior (S_{t-5}).

To best represent real human behavior, we use this insight to construct a simulator for daily step generation for our virtual players that we call the *Pattern Step Simulator*. This simulator generates steps for the virtual players that conform overall to the gamma distribution while also maintaining the associative relationships among days within players via the process described in our previous work [4].

Finally, in order for our virtual players to be amenable to the intervention and react to comparison opportunities, we adopt a model for SCO that can impart individual differences among virtual players. Reflecting the design of the Iowa-Netherlands Comparison Orientation Measure (INCOM) [17], the psychology instrument commonly used to evaluate SCO, this model consists of two integers $(0 \leq u \leq 1, 0 \leq d \leq 1)$ that respectively represent the virtual player's propensity to seek out upward and downward comparisons.

In our simulation, the virtual players are equipped to make all the participant player decisions earlier discussed in the multiplayer exercise, including selecting a profile for comparison based on the virtual player's (u, d) model. The values and relative magnitude of these variables determine the likelihood that a virtual player will select a profile in a particular direction and will react positively or negatively to these comparisons (i.e., increase or decrease their daily steps that day following the session), as demonstrated in our previous work [3].

IV. Multiplayer Modeling via MABs

The inclusion of multiple players in a shared environment may enable new player characteristics to model, but it also introduces new challenges to the construction of those models [14]. Specifically, we identify the three main challenges as 1) *best-choice estimation*, where multiple rewards from multiple players must be considered in MAB exploitation, 2) *exploration strategies*, which must now consider more than arm selection history when assessing the exploration space of player experiences, and 3) *social fairness*, where single decisions made by the AI can affect an experience or virtual environment shared by multiple individuals.

Regarding best-choice estimation, we consider a scenario in which an AI aims to maximize a particular metric in players that it can measure directly (e.g., frequency of feature use) and provides interventions that adapt the game toward increasing this metric. As the MAB applies its interventions (based on arm selections) over time, it can observe how the metric changes among the players and adjust accordingly. One way to manage metrics from multiple players might be to simply work with the average, but this approach may yield lower rewards; aggregating each player's metric into a combined value requires a loss of resolution on the data that might have been useful toward the MAB's understanding of the players.

As for exploration strategies, when adapting an MAB-based approach to a multiplayer context, the nature of both exploitation and exploration must be reconsidered because the arms of the bandit no longer necessarily share a one-to-one relationship with the experience provided to a player. Instead, observed rewards belong to a collection of players, predictions and exploitative arm pulls affect multiple individuals, and arms alone can no longer be used as a proxy for explored intervention states.

Finally, with regard to social fairness, where each of the AI's decisions will affect an experience shared by multiple players, we must consider how the right choice for the AI may not always be the arm that predicts the greatest reward. In sole pursuit of maximizing results, if over time the MAB continually favors a choice that benefits certain players, other players may be ignored or marginalized. In the worst case, a choice that favors certain players might work against or at the expense of others repeatedly. As the experience of all the players is placed in the care of the AI, the question is raised regarding the responsibility the AI has in ensuring that players receive equitable consideration. However, though we identify social fairness as a key challenge in multiplayer modeling, we do not address this challenge in this research but aim to explore it in our future work.

Therefore, we present the following two approaches toward addressing (respectively) challenges 1 and 2 above. The first addresses *multiplayer best-choice estimation*, exploring how a bandit strategy might interpret rewards and combine predictions for multiple players to determine the best overall choice. The second addresses *multiplayer exploration strategies*, investigating how explorative arm selections should consider not only the distribution of past selections but also the distribution of player experience that those selections have rendered.

A. Multiplayer Best-Choice Estimation

The MAB-based AI is driven by both its *oracle*, the predictive model built from observations of player behavior that predicts future rewards, and its *policy*, the decision process that determines when the MAB should explore choices (i.e., to improve the training data the oracle uses to make predictions) versus exploit the choices that the oracle currently believes will maximize results. For our policy, we use a standard ϵ-greedy ("epsilon greedy") strategy in which a parameter ϵ determines the percentage of time in which the policy will explore randomly among the choices not predicted by the oracle to be the best $(0 < \epsilon < 1)$.

As for the contribution of this work, our *Multiplayer Regression Oracle* (MRO) maintains a separate linear regression for each player. Because there are multiple players to consider in each arm pull, the regressions modeling each are combined into a multi-part oracle when a decision must be made by the AI. We also extend our previous work in regression-based oracles [4] by replacing the arbitrary *oracle value* as a feature in the regression with a representation of the arm that reflects the actual state of the intervention. Specifically, instead of assigning a numerical value to each arm (e.g., $A = 0, B = 1, ...$) and submitting those values to the regression, we provide the difference between the player's steps and the other two teammates as the value for the regression's arm feature.

Additionally, with insights regarding human step behavior resulting from our analysis of data from the Furberg et al. dataset [19], we also include in the regression oracle the observed daily steps for that virtual player over the past seven days [4]. Note that although the step data analysis did not indicate the steps five days prior (S_{t-5}) to be statistically

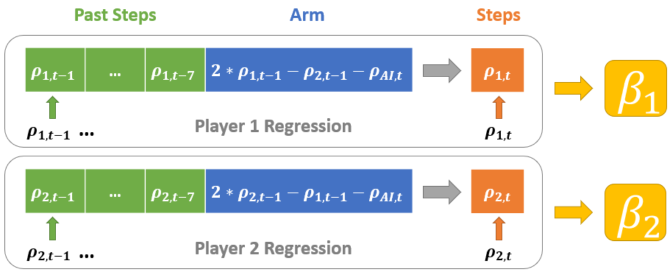


Fig. 2. The MRO includes a regression model for each player, where the rewards (i.e., steps) observed for each day are associated with an array of independent variables consisting of the rewards for the seven days prior as well as a feature conveying the nature of the arm selected that day. The regression models are updated following each observation, and the resulting coefficients for both players (β_1,β_2) are used by the oracle to make predictions.

significant, we do not omit it from the predictive model. We wish to explore the potential for the oracle to exploit previous rewards to predict future rewards in general, and thus in our experiments we prefer not to bias the approach with specific knowledge of the domain.

The complete MRO, illustrated in Figure 2, therefore associates the steps from previous days as well as the difference in steps between the player and both of their teammates against the resulting daily steps reported for that player. Specifically, the player regression models consist of dependent variables including the player's steps each day of the preceding week ($S_{t-1}, ..., S_{t-7}$) along with a feature representing the arm via the net result of the comparison between the player's steps and those of their teammates. In this, we include V_{t-1} as the steps of the other participant teammate the previous day and A_t as the steps for the AI teammate constructed by the bandit agent. These are associated with the independent variable S_t, or the steps observed for the player following their participation in the daily intervention, as indicated in Equation 1, where the $\frown$ operator denotes concatenation:

$$S_t = \epsilon_t + \sum_{i=1}^{8} \beta_i \{S_{t-1}, ..., S_{t-7}\} \frown (2S_{t-1} - V_{t-1} - A_t) \tag{1}$$

Each day when a new session is performed, and for both players separately (P1 and P2), the regression coefficients β are recalculated based on all observations for those players. Then, a prediction is calculated for each player and arm a based on those coefficients. First, a regression set $\hat{x}_{t,a}$ is constructed that joins the player's previous week's steps with what the comparison factor would be if that arm were pulled:

$$\hat{x}_{t,a} = \{S_{t-1}, ..., S_{t-7}\} \frown (2S_{t-1} - V_{t-1} - A_{t,a}) \tag{2}$$

A reward is predicted for the current day for each arm ($\hat{\rho}_{t,a}$) by evaluating each of today's features as they would be if that arm were pulled ($\hat{x}_{t,a}$) multiplied by the player's corresponding regression coefficients β.

$$\hat{\rho}_{t,a} = \sum_{i=1}^{m} \beta_i \hat{x}_{t,a,i} \tag{3}$$

The result is a predicted step value, which is standardized by subtracting the mean μ_p and dividing by the standard deviation σ_p of all such rewards previously observed for that player. This results in a value ($\hat{\rho}'_{t,a}$) indicating the predicted reward in terms of the number of standard deviations from the expected value for each arm and is calculated for each player separately.

$$\hat{\rho}'_{t,a} = \frac{\hat{\rho}_{t,a} - \mu_p}{\sigma_p} \tag{4}$$

For each arm, the corresponding $\hat{\rho}'_{t,a}$ values are averaged across players to create an aggregate predicted reward for that arm. The arm with the highest aggregate predicted reward is then selected by the oracle.

B. Multiplayer Exploration Strategies

To bootstrap the enhancement of the regression oracle, the MAB may select arms in the initial pulls that attempt to maximize exploration in a strategy referred to as *forced exploration*, which has been shown to improve MAB performance in short-horizon scenarios [4]. In the forced exploration period (i.e., the first n pulls), the bandit strategy considers the selections that will best provision its oracle by selecting among the less explored arms for which the confidence in reward is lower.

When modeling individual players, forced exploration is relatively straightforward: achieving the best exploration is simply a matter of balancing how many times each arm is pulled. In typical cases, this automatically offers a player a balanced exposure to the experiences offered by the AI. However, in the multiplayer case, the experience offered to each player is not wholly dictated by the arm that is selected; instead, aspects of the dynamics among the players may also factor into the resulting intervention state, and balancing arm pulls does not necessarily lead to a balanced *experience* for all players. For instance, a player in our simulation who has received arms A and C three times each could have seen anywhere from three upward and nine downward to nine downward and three upward opportunities, depending on the relative steps of their teammate during those experiences. A simplified example of such a scenario is illustrated in Figure 3.

Therefore, in our approach for *Multiplayer Forced Exploration* (MFE), we consider not only the current arm pull count but also the number of upward and downward comparisons that both players have experienced so far. Because both factors are important for properly constructing the players' regression models, we adopt the following approach to address both concerns when deciding which arm to pull to maximize exploration (i.e., to maximize the exposure of the players' regression models to new situations).

We track two arrays, C and P, where the former contains values regarding the frequency of intervention *cases*, or specific circumstances experienced by the players that are relevant to our model, and latter contains values regarding the frequency of *pulls* of each arm experienced by the players. The C array consists of as many values as there are intervention states s relative to each player p, where $C = (C_{j,k} : 1 \leq j \leq s, 1 \leq k \leq p)$. In our particular scenario, we record the number

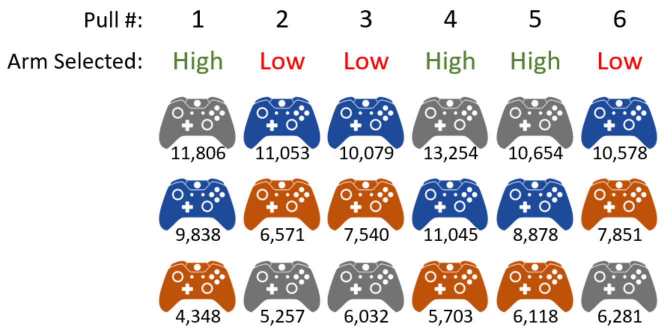


Fig. 3. In an example multiplayer system modeled as a two-armed bandit problem for simplicity, the steps of an AI teammate (grey) can either be placed "High" (20% above the highest) or "Low" (20% below the lowest). A sample sequence of six pulls with traditional forced exploration shows that both arms were indeed pulled an equal number of times. However, due to dynamics outside the AI's control, P1 (blue) receives a total of three upward comparison opportunities while receiving nine downward comparison opportunities. In this example, arm pull frequencies cannot serve as a reliable proxy for player experience within the intervention.

of upward and downward comparisons ($s = \{up, down\}$) experienced so far by each of the two players ($p = \{P1, P2\}$). The P array consists of n values corresponding to the bandit's arms $A = \{a_1, ..., a_n\}$, where $P = (P_1, ...P_n)$. In our particular scenario, we record the number of times each arm in our restless bandit has been pulled in aggregate among the experiences of both players.

To choose the arm that will maximize exploration, the MAB calculates the impact that each arm would have on both of these arrays in terms of the balance among their component values. Specifically, it examines for both arrays (C and P) and each arm a the value of the *delta variance function* $\Delta\sigma^2_a(X)$ that denotes the change in variance (σ^2) that array X would incur if arm a were pulled (thereby creating X_a).

$$\Delta\sigma^2_a(X) = \sigma^2(X_a) - \sigma^2(X) \tag{5}$$

For example, with respect to the P array, we find the histories of both players in a three-arm bandit scenario with arms $A = \{I, J, K\}$ have been aggregated to yield $P = [4, 6, 7]$. The variance of this array is $\sigma^2(P) = 1.555$. The MAB agent will then calculate the change in variance of this array that would occur when each of the potential arms were pulled. For instance, pulling the third arm (K) would result in an array $P_K = [4, 6, 8]$, the variance of which would be $\sigma^2(P_K) = 2.667$, and for which the change in variance would be $\Delta\sigma^2_K(P) = 1.555 - 2.667 = -1.11$. Alternatively, pulling the first arm (I) would result in an array $P_I = [5, 6, 7]$, the variance of which would be $\sigma^2(P_I) = 0.667$, and for which the change in variance would be $\Delta\sigma^2_I(P) = 1.555 - 0.667 = 0.89$. Because $+0.89$ is larger than -1.11, arm I would be considered to achieve greater exploration than arm K.

This is performed for every arm a for both arrays (C and P). A pair of weighted averages (W_C,W_P) is applied to both of the resulting values to find the final *Exploration Score* E_a for that arm:

$$E_a = W_C * \Delta\sigma^2_a(C) + W_P * \Delta\sigma^2_a(P) \tag{6}$$

The arm a that yields the highest value for E_a is selected as the arm to explore. It is worth noting that traditional forced exploration could be formalized as Equation 6 with $W_C = 0, W_P > 0$. As part of our simulation experiments, we explore a range of potential values for W_C and W_P to examine their relative influence on bandit performance.

V. Experimental Evaluation

To evaluate our framework, we conduct two experiments exploring bandit strategy performance against our simulation using virtual players. In these experiments, we examine the effects of our two proposed techniques along with standard UCB1 [20], UCBT [4], and ϵ-greedy policy strategies. First, we evaluate our proposed solution for the *best-choice estimation* problem by investigating the potential benefit of our MRO in its own bandit strategy: a variant of the standard ϵ-greedy strategy that replaces the default predictor with our multi-part regression model predictor. Second, we evaluate our proposed solution for the *exploration strategy* problem by exploring various weights (W_C, W_P) in our MFE approach to determine the effect that concern for either array factor (C and P, respectively) may have on performance.

For the policy strategies requiring the tuning of a parameter (e.g., UCB1's exploration constant C), we ran pre-tests of the strategy at various values for the parameter to determine the value that promoted the highest performance for that strategy in our simulation. This collection of pre-tests promoted a value of $C = 800$ for the UCB1 strategy and $\epsilon = 0.01$ for the ϵ-greedy strategy. Note that UCBT is a parameter-less UCB strategy and does not require any pre-test tuning [4].

Each bandit variant was examined over a horizon of h = 70 (i.e., a 10-week intervention with daily sessions), where we recorded observed rewards for each of the 70 time steps (or *pulls*). Each of these examinations was considered a *trial*, and the presented results of every experiment represent the average of both players' rewards over 500,000 trials. For both virtual players in every trial, the SCO player models (composed of the u and d factors discussed in Section III-D) were randomized.

A. Multiplayer Best-Choice Estimation Results

Traditional, single-player bandit strategies rely on the expected value of past rewards to make their predictions. In these approaches, the oracle chooses the arm to exploit by averaging the rewards of each arm in the past and choosing the arm with the greatest expected value. Some strategies (such as UCB-class strategies) also consider the variance of past rewards to construct confidence bounds that influence their policy decisions. In the multiplayer scenario facilitated by our simulation, which yields the rewards of two virtual players each pull, we combine the rewards of both virtual players by averaging them when they are received by these strategies.

In our first experiment, we introduce our MRO strategy that uses an ϵ-greedy ($\epsilon = 0.01$) policy and a two-part regression for oracle predictions, as described in Section IV-A. In this

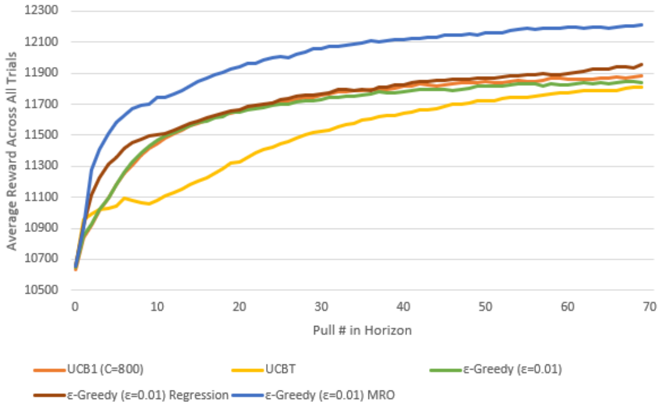


Fig. 4. Comparison of bandit strategy performance using ϵ-greedy and UCB-based policies in a short-horizon, multiplayer simulation with virtual players. The experimental MRO strategy (blue) implements the same policy as the traditional ϵ-greedy approach (green), except that it incorporates the multi-part regression oracle rather than the standard expected value-based predictor. The ϵ-greedy regression strategy (brown) also uses a regression oracle, but it does not maintain a separate regression model for each player.

TABLE I
AVERAGE REWARDS OF BANDIT STRATEGIES IN MULTIPLAYER SCENARIO ($h = 70$, ±99.9% CI)

Bandit Strategy	Overall Avg. Reward	Pull #70
ϵ-greedy ($\epsilon = 0.01$) MRO	**11976.4 (±116.0)**	**12208.6**
ϵ-greedy ($\epsilon = 0.01$) w/ Reg.	11707.2 (±96.6)	11943.1
ϵ-greedy ($\epsilon = 0.01$)	11654.8 (±101.2)	11838.2
UCB1 ($C = 800$)	11671.8 (±106.6)	11882.0
UCBT	11489.2 (±112.2)	11806.9

case, the separate rewards observed for both virtual players do not need to be combined but can instead be given to each player's regression model directly. For comparison, we include an additional ϵ-greedy strategy with a single-model regression oracle (versus the MRO's multi-model oracle) that averages both players' rewards when they are observed in the same fashion as the traditional strategies.

The results of this five-strategy experiment are presented in Table I with confidence intervals and visualized in Figure 4, where the average of both players' rewards (across all trials) are shown at each pull over the horizon. These results demonstrate the value of a predictor that does not simply consider every arm in isolation but rather the system as a whole. While we expect all of these bandit strategies to approach the same performance in the limit (i.e., an infinite horizon), we see the MRO strategy outperform ($p < .001$) all three traditional strategies and the single-model regression oracle variant in this short horizon that is more relevant to our scenario.

B. Multiplayer Exploration Strategy Results

Where we present that single-player forced exploration can be formalized as Equation 6 with $W_C = 0, W_P > 0$, we promote an approach that includes consideration for the C array factor (i.e., $W_C > 0$). Therefore, our second experiment explores various values for W_C and W_P to assess the impact they have on a multiplayer bandit strategy's performance.

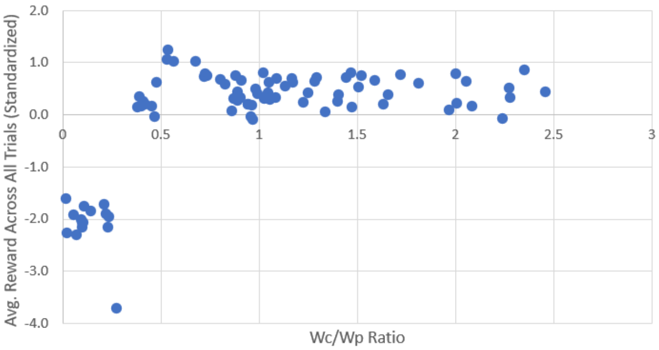


Fig. 5. Results of experiments with randomized values for W_C and W_P, with the ratio between the values plotted against the (standardized) reward observed. Experiments with ratio values higher than 3.0 were omitted from visualization but included in analysis. A higher ratio of W_C/W_P appears to correspond with higher performance, with an apparent threshold between 0.27 and 0.38 in our scenario, and where bandit strategies below this threshold demonstrated lower performance. Note that strategies using traditional forced exploration (i.e., $W_C = 0$) would all present below this threshold.

We performed 100 experiments using a standard ϵ-greedy ($\epsilon = 0.01$) strategy and adopting our MFE technique for the first 8 pulls of each trial. In each of the 100 experiments, the values for W_C and W_P were each randomly selected from the interval $[0, 3]$. The average rewards observed for each pull of each trial were aggregated and standardized across all experiments, with results visualized in Figure 5.

These results suggest that forced exploration strategies underperform in our multiplayer scenario when the W_C/W_P ratio drops below a threshold, supporting our intuition that W_C performs better at values higher than zero; that is, in multiplayer scenarios, the inclusion of the C array in forced exploration may hold an advantage over traditional approaches that consider only the P array.

VI. DISCUSSION

Our primary contribution in this paper is a new MAB framework for multiplayer modeling that addresses concerns regarding both exploitation and exploration in a multiplayer context. The addition of multiple players not only exposes new player characteristics available for modeling, but it also introduces new challenges for how an MAB might predict the impact of an intervention and breaks the assumption that MAB arms share a direct mapping to player experience.

The results of our experiments regarding our MRO approach support our intuition that MAB-based models for multiplayer scenarios should reconsider their process of exploitation and arm prediction, and we believe our strategy significantly outperformed the other strategies for three reasons. First, by combining all of the MAB's performance history into a single model for a player, the predictions made by the oracle are able to leverage the knowledge of every previous pull into every future prediction. In contrast, expected value-based oracles such as those used in the traditional scenarios only make use of each arm's respective history when predicting their future performance.

Second, our regression oracle enables the introduction of additional features, such as a player's previous output, which enables the oracle's predictions to exploit any inherent patterns that exist in that output (e.g., an individual's daily step behavior). It is worth noting that while the simulation was constructed to explicitly model these patterns in its step generation process, we expect such patterns to emerge organically in other real-world contexts where the MAB is repeatedly observing the same human player over time.

Third, because the MRO internally tracks players using separate models, it is able to maintain a higher level of resolution on each player during predictions. In contrast, the traditional strategies and the single-model regression strategy, which average the rewards across players at the time of observation, lose this resolution in the aggregation process. We expect there is advantage in waiting until the time of prediction to perform this aggregation as demonstrated in our approach, and we intend to investigate this further in our future work.

As for our MFE approach, we found that consideration for the C (intervention case) array (i.e., $W_C > 0$) outperformed traditional forced exploration that emphasized the P (pull frequency) array. While it does appear that both factors are important for maximizing performance, we believe a threshold exists for multiplayer scenarios regarding the W_C/W_P ratio beneath which strategies may underperform. This threshold is likely variable and dependent on the specifics of the scenario.

The experimental support for a non-zero W_C confirms our intuition that MAB-based models for multiplayer scenarios should reconsider their definition of exploration. When the arm selection merely reflects an aspect of the intervention state's construction, a player's actual behavioral responses to intervention states may derive more directly from their individual differences and the group dynamics. Therefore, we believe that tracking salient aspects of the intervention state for each player is, if not a more effective way to engage exploration, then at least an essential component for consideration.

VII. Conclusion

This paper focused on the understudied field of multiplayer modeling and presented an MAB-based multiplayer modeling approach. We identified the three main challenges for MAB-based multiplayer modeling that include *best-choice estimation*, *exploration strategies*, and *social fairness*; we addressed the first two by introducing a new oracle toward better predictions of player impact during exploitation pulls and a new forced exploration approach toward more accurate assessment of player experiences during exploration pulls.

Our experiments demonstrated that our Multiplayer Regression Oracle, which allows for players to be tracked separately and combined during predictions, significantly outperformed traditional and single-model regression strategies when applied to our multiplayer scenario. Our results also validated our Multiplayer Forced Exploration approach, supporting our intuition that exploration in multiplayer scenarios should consider player experience history as rendered in the game environment rather than the history of pulled arms alone.

As part of our future work, we would like to extend our analysis of multiplayer scenarios beyond two players, where we expect the discussed concerns will be further amplified. We also plan to examine the third challenge, namely that of *social fairness*, where the oracle might use alternative calculations (besides averaging) when making predictions for the group that consider outcomes beyond strict performance. Finally, in our current work, we are examining the potential for these techniques in human user studies in the context of exergames.

References

[1] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[2] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[3] R. C. Gray, J. Zhu, D. Arigo, E. Forman, and S. Ontañón, "Player modeling via multi-armed bandits," in *in Proceedings of the 15th International Conference on the Foundations of Digital Games*, 2020.

[4] R. C. Gray, J. Zhu, and S. Ontañón, "Regression oracles and exploration strategies for short-horizon multi-armed bandits," in *Proceedings of the 2020 IEEE Conference on Games*, 2020.

[5] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*. Citeseer, 2007, pp. 817–824.

[6] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, vol. 25, no. A, pp. 287–298, 1988.

[7] D. Charles and M. Black, "Dynamic player modeling: A framework for player-centered digital games," in *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, 2004, pp. 29–35.

[8] R. C. Gray, *Adaptive Game Input Using Knowledge of Player Capability: Designing for Individuals with Different Abilities*. Drexel University, 2018.

[9] J. Zhu, Y. Feng, A. Furqan, R. C. Gray, T. Day, J. Nebolsky, and K. Caro, "Towards supporting social motion-based games for health with agents," in *Proceedings of the 21st ACM Conference on Computer-Supported Cooperative Work and Social Computing*, 2018.

[10] J. Zhu, D. H. Dallal, R. C. Gray, J. Villareale, S. Ontañón, E. M. Forman, and D. Arigo, "Personalization paradox in behavior change apps: Lessons from a social comparison-based personalized app for physical activity," in *ACM PACM on Human Computer Interaction*, 2020.

[11] K. Z. Gajos and K. Chauncey, "The influence of personality traits and cognitive load on the use of adaptive user interfaces," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, 2017, pp. 301–306.

[12] P. W. Weyhrauch, *Guiding interactive drama*. Carnegie Mellon University, 1997.

[13] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. André, "Player modeling," *Artificial and Computational Intelligence in Games*, 2013.

[14] J. Zhu and S. Ontanón, "Experience management in multi-player games," in *Proceedings of the IEEE Conference on Games*, 2019.

[15] L. Festinger, "A Theory of Social Comparison Processes," *Hum. Relations*, vol. 7, no. 2, pp. 117–140, May 1954.

[16] J. V. Wood, "Theory and research concerning social comparisons of personal attributes." *Psychol. Bull.*, vol. 106, no. 2, p. 231, 1989.

[17] F. X. Gibbons and B. P. Buunk, "Individual differences in social comparison: Development of a scale of social comparison orientation," *J. Pers. Soc. Psychol.*, vol. 76, no. 1, pp. 129–142, 1999.

[18] A. P. Buunk and F. X. Gibbons, "Social comparison: The end of a theory and the emergence of a field," *Organ. Behav. Hum. Decis. Process.*, vol. 102, no. 1, pp. 3–21, 2007.

[19] R. Furberg, J. Brinton, M. Keating, and A. Ortiz, "Crowd-sourced fitbit datasets 03.12.2016-05.12.2016," 2016. [Online]. Available: http://doi.org/10.5281/zenodo.53894

[20] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, May 2002.