Beyond Winning and Losing: Modeling Human Motivations and Behaviors Using Inverse Reinforcement Learning

Baoxiang Wang*
bxwang@cse.cuhk.edu.hk

Xianjun Sam Zheng*
sam.zheng@gmail.com

Tongfang Sun
tongfs@uw.edu

Shengyu Zhang
syzhang@cse.cuhk.edu.hk

ABSTRACT

In recent years, reinforcement learning (RL) methods have been applied to model gameplay with great success, achieving super-human performance in various environments, such as Atari, Go, and Poker. However, those studies mostly focus on winning the game and have largely ignored the rich and complex human motivations, which are essential for understanding different players' diverse behaviors. In this paper, we present a novel method called Multi-Motivation Behavior Modeling (MMBM) that takes the multifaceted human motivations into consideration and models the underlying value structure of the players using inverse RL. Our approach does not require the access to the dynamic of the system, making it feasible to model complex interactive environments such as massively multiplayer online games. MMBM is tested on the World of Warcraft Avatar History dataset, which recorded over 70,000 users' gameplay spanning three years period. Our model reveals the significant difference of value structures among different player groups. Using the results of motivation modeling, we also predict and explain their diverse gameplay behaviors and provide a quantitative assessment of how the redesign of the game environment impacts players' behaviors.

INTRODUCTION

In recent years, reinforcement learning (RL) methods have been applied to model gameplay with great success, achieving super-human performance in various environments, such as Atari, Go, and Texas hold'em poker [10, 19, 11]. Those studies, however, primarily focus on winning the game, and the goal of the computer agent is to take actions that can maximize the cumulative scalar rewards, such as achieving high scores or beating the opponents. They have mostly ignored the rich and complex human motivations, which are essential for understanding different players' reward mechanism as well as their complex and diverse behaviors. In fact, numerous behavioral and psychology studies [17, 2] have shown that when people are playing games, apart from competing and winning, they also try to connect with others, or they just want to have some fun or enjoyment by themselves. An extensive survey of game motivation [22, 21] with 30,000 players on Massively-Multiplayer Online Games (MMOGs) confirms that human players have complex and multifaceted motivations. As shown in Tbl. 1, the study categorizes the complex motivations of gameplay into ten different types and three different groups, namely, Achievement, Social, and Immersion.

*These authors contribute equally to this work.

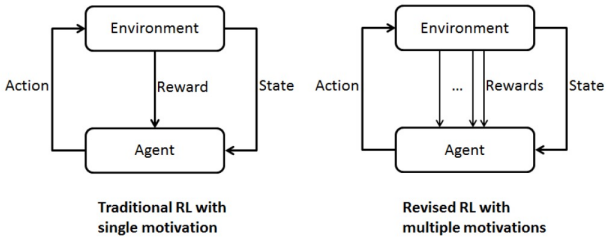

Figure 1. In the typical RL model (left), an agent or player has only one single motivation and maximizes one scalar reward. In MMBM (right), an agent or player has multiple motives and the goal is to optimize the combination of different rewards based on each agent's value structure.

In this paper, we propose a novel method called Multi-Motivation Behavior Modeling (MMBM) that is based on RL and takes into consideration the multifaceted human motivations. The objective of MMBM is to model the underlying value structure of the players from the observed human behavior. By incorporating the motivation theory in [22] of gameplay, we extend the standard RL framework to cover multiple rewards situations. In MMBM, the goal of the agents (or players) is not simply to maximize one scalar reward under one single motivation such as achieving high scores, but instead to maximize the combination of multiple rewards based on the multi-faceted motivations. Fig. 1 illustrates the difference between the typical RL vs. our proposed MMBM. The challenge in discovering human' motivations is that they are not explicitly observable. Instead, we have to infer them from the players' behaviors, which can be achieved by using inverse reinforcement learning (IRL). In MMBM, we extent IRL to uncover the complex, multi-dimensional reward mechanism. Our model first quantifies each dimension of the reward signal individually Based on the motivation theory. The individual signals are subsequently combined under the assumption that each player appears in the trajectory are acting at the best of optimizing their objectives. In this way, decomposition of the full reward signal is reduced into a linear program, which is solved efficiently, and subsequently the value structures of the players can be computed.

A significant advantage of MMBM is that it utilizes only off-policy learning: Each of the individual reward signals is estimated by Q-learning with deep Q-networks (DQN), and MMBM's IRL algorithm takes only the trajectories as its input. In this way, MMBM does not require a simulator of the

Table 1. Components of game motivation

Components	Sub-components
Achievement	Advancement, Mechanics, Competition
Social	Socializing, Relationship, Teamwork
Immersion	Discovery, Role Playing, Customization, Escapism

environment, nor does it inquire human players' counterfactual actions that do not exist in the dataset. This is beneficial since most of the existing IRL methods have to have access to either the simulation environment or actual human policies, which are usually costly to obtain or simply do not exist. For large and complex games, MMBM provides a feasible way to analyze the historical data.

We apply MMBM to model the players' behaviors and motivations of World of Warcraft, which is one of the most successful massively multiplayer online role-playing games with millions of subscribers worldwide. We test the MMBM on the World of Warcraft Avatar History (WoWAH) dataset [8] with 70,000 users' gameplay spanning over a three-year period. Our method outputs the value structure which is the most succinct description of the game environment in the perspective of the human players. On top of the value structure, it also predicts the players' behaviors accurately, outperforming existing approaches such as large-margin Q-learning [15] and policy imitation via classifier. Moreover, it reveals the different reward functions and diverse value structures among different player groups, which interestingly agrees with previous knowledge-based studies on WoW [6, 12]

PRELIMINARIES

Inverse Reinforcement Learning

The process of recovering the reward function from observed trajectories is inverse reinforcement learning (IRL). It reverses the input and output pairs of RL algorithms, computing the rewards function according to the policies or actions of the agents. The basic assumption of IRL is that, though the reward function is unknown, it exists and the agents' actions are conducted to maximize the cumulative reward. The assumption is illustrated by a few different forms in mathematic forms, including linear IRL [1, 13], max-entropy IRL [16], and large-margin Q-learning [15], and etc. Most of the existing IRL algorithms have either of the two requirements: they need to access the dynamics of the environment [1, 10], which is usually provided by a simulator of the game; or to access the policy function [23], which requires the agent to retroactively compute the counterfactual action at a historical decision point. Such requirements are expensive in complex and massive games where human players involved. Hence, an IRL algorithm without those need is desired. Approaches such as large-margin Q-learning [14, 7] and our proposed MMBM do not require that and are suitable for complex, real-world game environments.

Deep Q-Learning and Large Margin Q-Learning

We first define some notions in RL. In a game environment, at each round t, the player conducts an action a_t according to their own policy $\pi(\cdot)$ and current game state s_t. The player may not be able to obtain the full game state (such as events happened out of the vision of the player) and uses the observation x_t to substitute s_t. The player subsequently receives the feedback from the environment, including a scalar reward r_t and the observation x_{t+1} of the next round. The player's intention is to maximize his/her discounted cumulative reward, also known as the action-value function,

$$Q^{\pi}(s,a) = \mathbb{E}[R_t | s_t = s, a_t = a, \pi], \tag{1}$$

where $R_t = \sum_{t' \geq t} \gamma^{t'-t} r_{t'}$. Deep Q-Learning uses a Deep Q-Network (DQN) to estimates the $Q^{\pi^*}(s,a)$ value, where π^* is the maxima of the Q-value over all policies. DQN uses the recursive relation of $Q^{\pi^*}(s,a)$, known as the Bellman equation

$$Q^{\pi^*}(s_t,a_t) = r_t + \gamma \max_{a'} Q^{\pi^*}(s_{t+1},a'). \tag{2}$$

The estimation is conducted by minimizing the Bellman error

$$L_1 = \mathbb{E}[\frac{1}{2}(Q^{\pi^*}(s_t,a_t|\theta) - y')^2], \tag{3}$$

where $y' = r_t - \gamma \max_{a'} Q^{\pi^*}(s_{t+1},a'|\theta')$ is the target value function and θ' the parameter of the target network.

While DQN estimates the $Q(s,a)$ function of π^* from the reward signals, which is subsequently used to retrieve the optimal policy, large-margin Q-learning approximates the action-value function corresponding to the observed behavior directly. Suppose the policy and the reward signals of the player or a group of players are unknown and we have observed a set of state-action pairs generated by such a policy. Large-margin Q-learning [14, 7] assumes (as most of the IRL algorithms do) that the players' actions are intended to maximize their action-value, namely,

$$Q^*(s,a) \geq \max_{a' \in \mathscr{A}(s)} Q^*(s,a') \tag{4}$$

is satisfied for all state-action pairs with a margin. Note that $\mathscr{A}(s)$ is the set of all feasible actions under state s. Adding a large margin toward the difference between inequality (4), it results in the error term

$$L_2 = \frac{1}{2}(Q^{\pi^*}(s_t,a_t) - (l_{s,a} + \max_{a'} Q^{\pi^*}(s_t,a'))^2, \tag{5}$$

where $l_{s,a}$ is the margins, which could either be pre-defined parameters trainable parameters.

METHODS

Reward Mechanism Modeling in MMBM

We present our MMBM algorithm to compute the underlying reward function of the agents. Our method can be viewed as a two-step workflow: The first step, known as Q-learning, estimates the reward functions of the players at different states of gameplay environment; The second step, a variant of inverse reinforcement learning (IRL), estimates the combination or weights of the different rewards learned at the first step. In essence, the two-step methodology decomposes the complex interactions between players and a game environment into multiple quantitative metrics and solve them separately. An

intuitive illustration of the two-step framework on WoWAH is shown in Fig. 2, while the formal algorithm is described in 1.

The fundamental idea behind the first step is that in a complex environment, given the same situation or state, different players respectively perform their optimal actions and exhibit diverse behaviors. For example, a player who values more about the relationship with his/her teammates would spend more time on team-based activities than those players who focus more on their advancements or achievements. Because he/she receives more overall rewards, by getting more teamwork-based rewards that he/she values. Hence, the combination of multiple reward signals is essential to model the users' behavior and their underlying value structure.

MMBM learns the weights of the combination of multiple motivations from the user behavior data using IRL techniques. Formally, let $\mathscr{T}$ be the set of state-action pairs of a user or a group of users. It consists of the choices of the users (correspond to the term *actions* a_t in RL; t stands for time step index) under various of situation or scenarios (correspond to the term *states* s_t in RL). We can also infer from the states that the agents are receiving feedback on multiple rewards $f_t = (f_t^1, \ldots, f_t^n)$ simultaneously. Assume that the players are optimal in processing any information available to them and display optimal trajectory towards their objectives [1]. Then, given the same environment state, different players perform diverse actions or display complex behaviors must be resulted from their different motivations or value structure $f_t^1, \ldots, f_t^n$. With the assumption, it reduces to find a valid combination of rewards such that under that combination every action is optimal, that is, there does not exist another feasible action that yields a higher total reward. Let $\phi \in \mathbb{R}^n$ be the combination weights, subjecting to $||\phi||_1 = 1,\ \phi \geq 0$, define the reward as

$$r_t = \phi^T f_t \tag{6}$$

The action-value function (or Q-function) describes the objective of the user

$$Q^*(s,a) = \mathbb{E}[R_t | s_t = s, a_t = a, \pi^*], \tag{7}$$

where

$$R_t = \sum_{t' \geq t} \gamma^{t'-t} r_{t'}.$$

Q function gives the expected cumulative rewards the user gets if the user chooses optimal action a under state s and follows the best policy thereafter. It is the function that should satisfy the previously discussed action optimality, which can be formulated as

$$Q^*(s,a) \geq \max_{a' \in \mathscr{A}(s)} Q^*(s,a'), \tag{8}$$

where $a' \in \mathscr{A}(s)$ is the set of all possible actions the user can take at the state s. Since $Q^*(s,a)$ is a function of ϕ, solving inequation (8) will yield the combination weight ϕ we want.

Though (8) itself could be infeasible and hard to solve, we apply two approximation to find the solution. First, let $Q^i(s,a)$ be the action-value function, as if f^i is the only existing reward signal

$$Q^i(s,a) = \mathbb{E}[\sum_{t' \geq t} \gamma^{t'-t} f_{t'}^i | s_t = s, a_t = a, \pi^i] \tag{9}$$

At this moment we assume the such Q^i function can be accurately estimated. We then apply linear scalarization [15] from IRL to explicitly separate out the weights ϕ

$$Q^*(s,a) = \phi^T \tilde{Q}(s,a), \tag{10}$$

where the vector of function $\tilde{Q}(s,a) = (Q^1(\cdot), \ldots, Q^n(\cdot))$. Second, we introduce the slack variables $\xi_{s,a}$, which models the cases that users behave less optimally, such as making mistakes or just playing randomly. $\xi_{s,a}$ sets the threshold of the difference between the actual action-value $Q^*(s,a)$ and the largest possible action-value $\max_{a' \in A(s)} Q^*(s,a')$ over all feasible actions. The value of $\xi_{s,a}$ is positive whenever inequality (8) is not satisfied, and zero otherwise. Solving the inequation (8) is reduced to minimizing the summation of the slack variable $\xi_{s,a}$ over all observed pairs, which is

$$-\sum_{s,a} \left[\min(0, Q^*(s,a) - \max_{a' \in A(s) \setminus a} Q^*(s,a')) \right] .. \tag{11}$$

After the two approximation steps, minimizing such total slacks is reduced into a linear program (LP) problem as follows.

Given that the action-value function $\tilde{Q}(s,a)$ for each of the reward signal, and let $\mathscr{T}$ be the set of observed state-action pairs of (that is, our dataset), minimization of the summation of the slack variables (11) is formulated into the following LP

$$\begin{aligned} \underset{\phi,\xi}{\text{minimize}} \quad & \sum_{s,a} \xi_{s,a} \\ \text{subject to} \quad & \phi^T(\tilde{Q}(s,a) - \tilde{Q}(s,a')) \geq -\xi_{s,a}, \\ & \qquad \forall (s,a) \in \mathscr{T}, a' \in \mathscr{A}(s) \setminus a \\ & \phi \geq 0,\ ||\phi||_1 \geq 1 \\ & \xi_{s,a} \geq 0,\ \forall (s,a) \in \mathscr{T}. \end{aligned} \tag{12}$$

As LP can be solved efficiently, MMBM finds the composition of the rewards by solving the weights ϕ of different reward signals in Eq. (12).

The remaining problem is to estimate the action-value action-value function $\tilde{Q}(s,a)$, which is solved by Q-learning via DQN. Referring to Algorithm 1, line #13-16 are the decomposing part which is reduced to LP (12) and line #4-12 are the DQN approach in MMBM, which is standard in RL and detailed in the next section with respect to our environment settings. With our two-step approach, MMBM takes the history of state-action pairs as input, which is usually logged during the gameplay. It solves ϕ , which is a quantitative description of human players' motivations and the value structure.

Off-Policy Action-value Function Approximation

In Alg. 1, MMBM requires the approximation of the action-value function $Q^i(s,a)$ for each of the component of the rewards . Such approximation should be a fair estimation of the cumulative reward the user would receive if the user chooses

[1]The assumption is reasonable as we take multiple dimension of reward signals into consideration

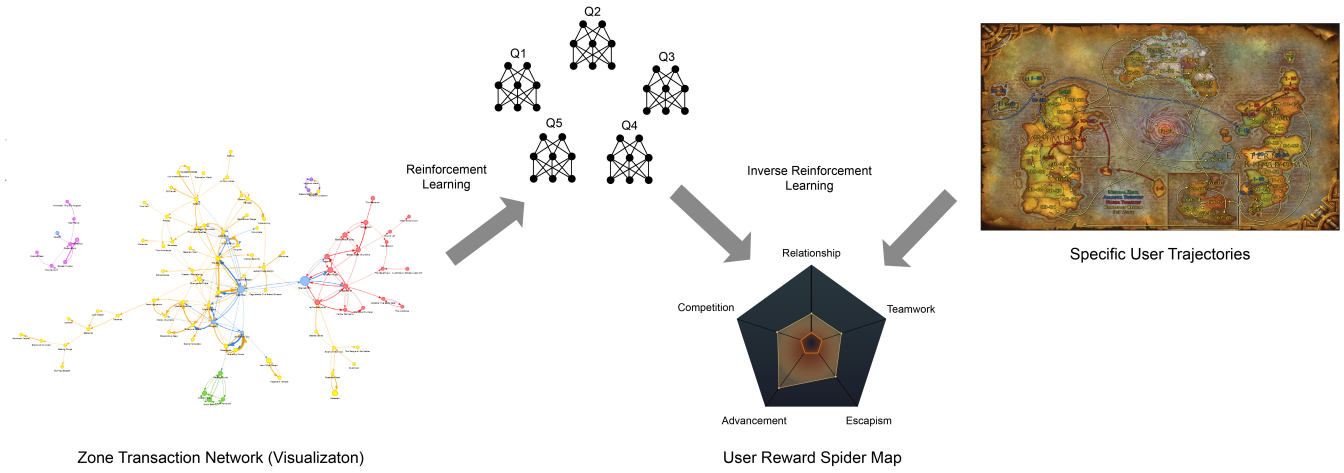

Figure 2. Illustrative execution of Alg. 1 on WoWAH

Algorithm 1 MMBM

Parameters: learning rate α, discount factor γ
Initialization: initialize network parameters w^i randomly
Input: set $\mathscr{T}$ of trajectories
for $i = 1$ **to** n **do**
  for t **to** size of $\mathscr{T}$ **do**
    Calculate f_t^i
  end for
  repeat
    Compute $L_1^i = \mathbb{E}[\frac{1}{2}(Q^i(s_t, a_t|w^i) - y')^2]$
    Update $w^i = w^i - \alpha \nabla_{w^i} L_1^i$
  until convergence of $Q^i(s,a)$
end for
for t **to** size of $\mathscr{T}$ **do**
  Compute $\tilde{Q}(s,a) = (Q^1(\cdot), \ldots, Q^n(\cdot))$
end for
Find ϕ by solving linear program (12)

an action a at the state s and maximizes the i-th reward thereafter. DQN uses the recursive property (i.e. Bellman equation) that the action-value estimator should have, that is, the cumulative reward since the current step onwards should be the immediate reward plus the cumulative reward since the next step onwards. Using the property, DQN updates the action-value function iteratively, by moving the $Q^i(s,a)$ value toward (by upgrading the parameters) it's target $r_t + \gamma Q^i(s', a^*)$. By the time $Q^i(\cdot)$ converges to satisfy the Bellman equation, it estimates the action-value given any state-action pair.

An advantage of Q-learning is that it learns off-policy property, which implies that the action-value function approximation does not rely on real-time data or any game simulator. To understand this, we observe that the $(s,a,s',a^*$ tuples used in the iterations of the update could be feed into the model with an arbitrary order and could involve any a without being required that a is generated by a certain policy. It is very important to our algorithm because since the interaction records between the agent and the environment, such as the computer-human iteration, are usually available in its offline mode. This means our MMBM does not require the dynamics of the environment for training the model. Using gameplay historical data or player behavior log data as the input, MMBM models complex game environments such as massively multiplayer online games.

The function approximator which parametrizes the action-value functions largely depends on the environment. Taking our experiments on the WoWAH dataset as an example, the DQN architecture is designed according to the available observations and is applied to all reward signals $i = 1, \ldots, n$. As shown in Fig. 3, the categorical elements of the input (e.g. *race*, *class*, etc.) are first processed by an embedding layer [9], while the numeral elements (e.g. session length, current level etc.) are first fed into a fully connected (FC) layer with rectifier non-linearity. The output of embedding layer and FC layer are then concatenated and fed into another FC layer with rectifier non-linearity. A final FC layer is applied to compute the $Q(s,a)$ value for each action $a \in \cup_s \mathscr{A}(s)$. The detailed introduction of the environment and the details of each of the input variables are included in the experiment section.

Imitation Learning and Predictions

An immediate use of the action-value function is to derive the optimal policy which imitates the gameplay of the samples. That is, let $\pi^*(s)$ denote the action at state s

$$\pi^*(s) = \text{argmax}_{a'} Q^*(s,a') \tag{13}$$

is the policy function which predicts the players move. The intuitive understanding of our predicting power is if the players' reward system is available, we could easily predict the players' behavior. Moreover, the predictions of the user behavior are with a reason behind: While most of the classification models are just black boxes their outputs may not be corresponding to a clear intuitive. MMBM, instead, reveals the underlying system that drives the behavior before making predictions

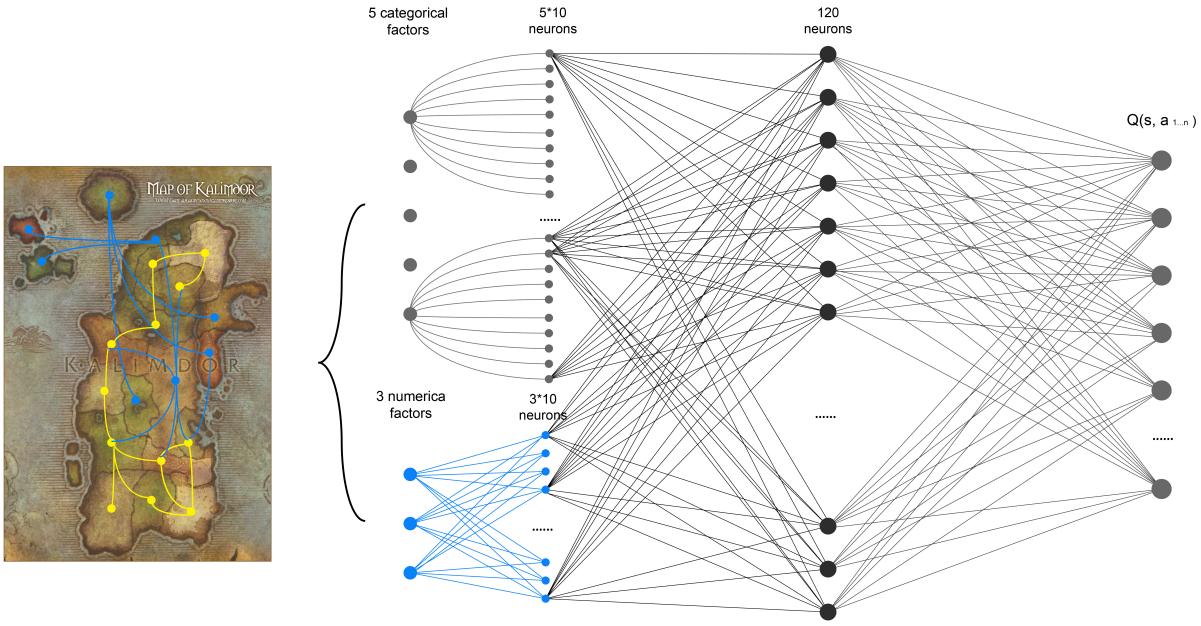

Figure 3. DQN architecture for Q^i training, $i = 1, \ldots, n$, on WoWAH

To make predictions, MMBM first solving LP (12), and get the combination of reward signal r_t. As the action-value function Q^i have been already learned, the action-value function $Q^*(s,a)$ becomes known by applying Eq. (10). To avoid the bias involved in the scalarization process, we learn the $Q^*(s,a)$ using the combination weights and the original dataset once more. The re-train of then action-value function can be generalized to a more complex combination, for example, Pareto combination of individual reward signals. Note that, $Q^*(s,a)$ and $\pi^*(s)$ are not corresponding to just advancement or fastest leveling up in the game. MMBM is beyond winning and losing: it models and predicts the actual actions that the humans would have conducted once they present at such a state.

EXPERIMENTAL ANALYSIS

WoWAH Player Behavior Dataset

We tested our MMBM in WoW, one of the most successful MMORPGs in the world with millions of active players. Like other MMORPGs, each player chooses a character avatar and control the avatar in third- or first-person view throughout the game. Players can explore the landscape, fight various monsters, complete quests individually or cooperatively, communicating and interacting with other players, or build their guilds (groups). As shown by Yee [22, 21], players' motivations are distinct and their actions and behavior are complicated. The WoWAH dataset [8] is an interesting dataset to investigate the behavior. It records a significant amount of gameplay data with over 70,000 players' movements (regarded as actions) from realm *TW-Light's Hope* every 10 minutes spanning for the 3-year period. Previous studies on this dataset are either based on descriptive statistics [8] or using simple classifiers or clustering [20, 5, 3, 4], and these methods fail to capture the rich and complex motivations of the players.

From the reinforcement learning perspective, we treat each player as a human agent who conducts an action at each time interval. All available data such as current level or joining the *guild* are regarded as observations. The players' trajectories are composed of a sequence of locations and observations, which partially reflect their playing strategies. [4, 18]. Even though Yee theorized ten different motivation for gameplay, we apply five of them that are frequently observed in the WoWAH dataset. Therefore, we compute five different kinds of motivations using the WoWAH dataset and let $n = 5$ in Alg. 1. The constructions of the motivation values $f_1, \ldots, f_5$ are illustrated in Tbl. 2, and are based on the Yee's research and other WoW case studies [6, 12]. With those value, we model the final reward function that each player tries to maximize during the gameplay.

Player Motivation Modeling

We present our experimental results on recovering the multi-motivation mechanism, which is the solution of LP (12). We use Tbl. 2 and solving LP (12) on trajectories that are randomly drawn from the WoWAH dataset. The underlying reward mechanism and value structure of the whole player community is

$$\phi = (0.40, 0.10, 0.21, 0.16, 0.12)^T. \qquad (14)$$

In other words, when choosing an action or conducting a behavior, the players' total motivation is composed of 40% their advancement, 10% their competition, 21% their relationship, 16% their teamwork, and 12% their escapism on average. The

Table 2. Different types of motivations in WoWAH and corresponding definitions

Motv.	**Category** & Definition
f^1	**Advancement** describes how fast the player levels up in the game. It's the speed the user levels up, divided by the averaged speed at the entire WoWAH.
f^2	**Competition** describes if the player joins *Battleground* or *Arena* and competing with human opponents. It equals the number of visits.
f^3	**Relationship** is linear to the duration that the player has been in the current *guild*.
f^4	**Teamwork** describes the intention of conducting teamwork, which is the number of recent zones with teamwork features visited. Zones with teamwork features include *Battleground*, *Arena*, *Dungeon*, *Raid*, or a zone controlled by *The Alliance*.
f^5	**Escapism** is the linear combination of the duration of the recent game session and the number of days the player continuously login to the game recently.

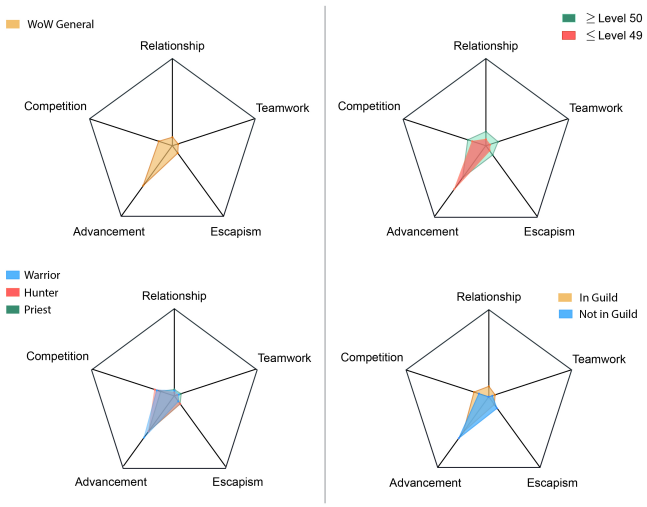

Figure 4. Spider maps to represent player reward mechanism or value structure. Top-left: the weights of different motivations for the entire WoW player community; top-right: different value structures between the players at higher level (≥ 50) and the players at lower level (≤ 49); bottom-left: different value structures between the players in different classes *Warrior*, *Hunter*, and *Priest*; bottom-right: comparison of different value structure of the players who are in *guild* and those who are not in a *guild*

results are illustrated as a spider map in Fig. 4. Note that the above ϕ is calculated based on the entire player database, and our MMBM can calculate the respective ϕ vector for an individual player or a player group.

We show some comparison results for different player groups. Significantly value structures difference is observed between the players at a higher level (≥ 50) versus the players at a lower level (≤ 49), where the players at the lower level are much more motivated to advance as indicated by the bigger weight on the Advancement motivation. It also shows interesting difference among players in different classes, *Warrior*, *Hunter*, and *Priest*, where the *Warrior* players value more on Advancement and the *Priest* players value more about relationship. It agrees with the common knowledge in WoW that the spells of *Priest* focus on benefiting (healing, buffing, etc.) the team rather than those of *Hunder* and *Warrior* whose spells are more related about damage, and damage/tank, respectively. Lastly, the results also show that players in the *guild* value more about Teamwork and Relationship motivations as compared to the players that arenâĂŹt in a *guild*. The difference of th weights are distributed into *advancement* and *escapism* instead. Interestingly, those quantitative results agrees with previous knowledge-based studies on WoW [6, 12].

Predicting Players' Behavior

Once MMBM models the humans' motivation and value structure, it straightforwardly predicts the complex user behaviors. The prediction is made by the policy stated in Eq. (13). In WoW, at any given state, the player chooses its movements to stay in the same zone or move to another feasible zone. The action space is discrete, and depending on the players' level the size ranges from a few zones or over one hundred zones. Therefore, the chance of randomly guess players' next action is quite low. Our predictions are quite accurate considering the difficulties and action space size.

We evaluate the accuracy of the prediction, by comparing if the predicted action $\pi^*(s)$ agrees with the actual action a for the (s,a) pairs in the dataset. Experiments show that policies induced by a biased reward function underperform our π^*. That is computed by adding a disturb factor $\varepsilon \sim \mathcal{N}(0, 0.05)$ to the solution ϕ of LP (12), as shown in 3. We also compare our result with the policy only focuses on advancement, by setting $\phi = (1,0,0,0,0)^T$, and the results show that our approach predicts players' actions significantly better. This finding indicates that taking into consideration of multiple motivations of the user or player not only can reveal each player's different value structure but can also predict the complex user behavior more accurately. Then, we test the large-margin Q-learning and policy imitation and the results show that both these method are less accurate compared with our MMBM. Note that policy imitation via supervised learning is implemented by a multi-class support vector machine, mapping the state s to the action a.

A close examination of the errors that are made during the prediction yields some interesting understandings. As our MMBM model assumes that every player tries to maximize their cumulative reward in Eq. (4), i.e., everyone is regarded as a rational and optimal player. Unfortunately, our model would have some trouble to distinguish whether a particular action that deviates from an average one is caused by the player's actual intention or the player's sub-optimality during the gameplay. For instance, some of the players could spend hundreds of hours on solo *quest* but fail to level up quickly. This could be due to their intention of enjoying doing the quest repeatedly or the players not knowing the optimal strategy to level up. We will address this limitation of our method in the

Table 3. Accuracy of different approaches

Approach	Accuracy	Notes
π^*	56.5%	From Eq. (13)
Dist. π^*	52.5%	Use $\phi + \varepsilon$ instead
π^1	45.9%	Use $\phi = (1,0,0,0,0)^T$
LMQL	47.2%	Large margin Q-learning
PI	31.0%	Policy imitation via SL
Linear Q	29.5%	Replace DQN w/ linear

future work by considering humans' different ability or skill levels.

Dynamics of the Human Motivation

The motivation of gameplay may evolve. It can also be impacted by the new design or new versions of the game environment. How would a design update affect the users' motivations and behaviors and how we can quantify this impact? It's a very interesting question for every game designer to consider. We conduct an analysis of the dynamics of player motivations on the WoWAH dataset, i.e. how the underlying reward mechanism for players changes over time. To achieve this, the idea is that the set $\mathscr{T}$ in LP (12) may contain any numbers of trajectories. Randomly drawing (s_t, a_t) from the dataset where t is restricted to a specific time range yields the set $\mathscr{T}$ which illustrates the player's motivations during that time range. Taking the time range chronologically, we show the evolution of game motivation, characterized by the elements in ϕ. Fig. 5 illustrates the trend of *Advancement*, *Competition*, *Relationship*, *Teamwork*, and *Escapism*.[2].

First, we observed the dramatic increase in Advancement and Competition during the mid-to-late period on the graph. It happens at around the 150000th time interval, which coincides with the release of the patch *Wraith of the Lich King* on November 2008. Analyzing the game update patch, two primary reasons can explain the increased level of motivation on Advancement. First, the patch increased the maximum player level from 70 to 80. As a result, the players with level 70, the previous max level, were rushing to complete the remaining ten leveling ups to reach the new max level. Second, the patch introduced two new classes in the game, namely *Death Knight* and *Shaman*, and this gave incentives to many players to open the secondary accounts and to level up them is the first thing to do afterward. Meanwhile, the reason for more Competition is that many players tend to join player-versus-player (PvP) to compete with other human players to get more familiar with the mechanism of their new avatar. It's also noticed that the satisfactions are not independent of each other: players spend more time on advancement usually have insufficient time to complete tasks which require teamwork but provides no experience for leveling up. That's shown in Fig. 5 that the weight for teamwork decreases each time the weight for advancement increases, and vice versa.

We analysis the overall trend of the game during the three years when WoWAH was collected. It turns out that the game

[2]Note that at any time the weights of those elements sum to 1, representing how players value those satisfactions relatively.

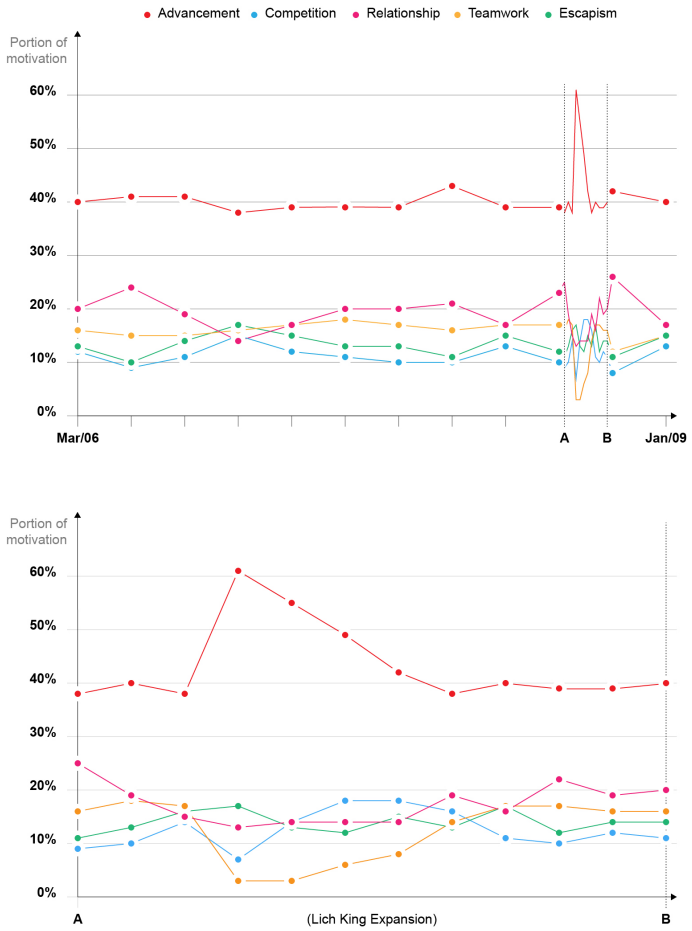

Figure 5. Top: trends of different kinds of motivations during from Mar 2006 to Jan 2009; Bottom: the enlargement of the top figure during around the release of patch *Wraith of the Lich King*

emphasis more on teamwork and relationship during the period, partially because the dataset was collected only two years after the game release, and the players are getting more and more involved in the game during that time. Apart from that, the weights of different kinds of motivations are under influences from both game patches and updates, and game user community. Overall, our MMBM model and analysis provides useful insights in Fig. 5 for game designers and researchers.

CONCLUSIONS AND FUTURE WORK

We present MMBM, a general RL model that takes multi-faceted human motivations into consideration. MMBM conducts the IRL task, while not relying on the access of policy function nor the dynamics of the environment. Hence, MMBM can be applied to study complex, interactive environments with its historical dataset. Our experiment results on the WoWAH dataset shows that MMBM recovers reasonable reward mechanism of the players. On top of that, it predicts human playersâĂŹ behaviors accurately, shows how different group of players have respective value structure, and provides a quantitative assessment of how the redesign of the game environment impacts players' behaviors.

We view our work as one of the first that can combine the richness of psychological and game research theories with the rigorousness of RL models. Our goal is beyond winning and losing: not simply to create software agents that beat human in various games or competitions, but to propose methods that can help to understand the intricacy and complexity of human motivations and their behaviors. We hope to inspire more researchers to investigate this topic further.

REFERENCES

1. Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 1.
2. JAMES TYRONE ORPILLA ALVARADO. 2005. *Playing with Power: An Examination of a Massive Multiplayer Online Role Playing Game*. Ph.D. Dissertation. Alliant International University.
3. Christian Bauckhage, Anders Drachen, and Rafet Sifa. 2015. Clustering game behavior data. *IEEE Transactions on Computational Intelligence and AI in Games* 7, 3 (2015), 266–278.
4. Jonathan Bell, Swapneel Sheth, and Gail Kaiser. 2013. A large-scale, longitudinal study of user profiles in world of warcraft. In *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 1175–1184.
5. Anders Drachen, Christian Thurau, Rafet Sifa, and Christian Bauckhage. 2014. A comparison of methods for player clustering via behavioral telemetry. *arXiv preprint arXiv:1407.3950* (2014).
6. Nicolas Ducheneaut, Nicholas Yee, Eric Nickell, and Robert J Moore. 2006. Alone together?: exploring the social dynamics of massively multiplayer online games. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 407–416.
7. Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, and others. 2018. Deep Q-learning from Demonstrations. *Association for the Advancement of Artificial Intelligence (AAAI)* (2018).
8. Yeng-Ting Lee, Kuan-Ta Chen, Yun-Maw Cheng, and Chin-Laung Lei. 2011. World of Warcraft avatar history dataset. In *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 123–128.
9. T Mikolov and J Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* (2013).
10. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and others. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
11. Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. *arXiv preprint arXiv:1701.01724* (2017).
12. Bonnie Nardi and Justin Harris. 2006. Strangers and friends: Collaborative play in World of Warcraft. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. ACM, 149–158.
13. Andrew Y Ng, Stuart J Russell, and others. 2000. Algorithms for inverse reinforcement learning.. In *Icml*. 663–670.
14. Shibin Parameswaran and Kilian Q Weinberger. 2010. Large margin multi-task metric learning. In *Advances in neural information processing systems*. 1867–1875.
15. Bilal Piot, Matthieu Geist, and Olivier Pietquin. 2013. Learning from demonstrations: Is it worth estimating a reward function?. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 17–32.
16. Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. 2006. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 729–736.
17. Daniel Schultheiss. 2007. Long-term motivations to play MMOGs: A longitudinal study on motivations, experience and behavior. In *DiGRA*. 344–348.
18. Siqi Shen, Niels Brouwers, Alexandru Iosup, and Dick Epema. 2014. Characterization of human mobility in networked virtual environments. In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. ACM, 13.
19. David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and others. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.
20. Mirko Suznjevic, Ivana Stupar, and Maja Matijasevic. 2011. MMORPG player behavior model based on player action categories. In *Proceedings of the 10th Annual Workshop on Network and Systems Support for Games*. IEEE Press, 6.
21. Nick Yee. 2006a. The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. *Presence: Teleoperators and virtual environments* 15, 3 (2006), 309–329.
22. Nick Yee. 2006b. Motivations for play in online games. *CyberPsychology & behavior* 9, 6 (2006), 772–775.
23. Martin Zinkevich, Michael Johanson, Michael H Bowling, and Carmelo Piccione. 2007. Regret Minimization in Games with Incomplete Information.. In *NIPS*. 1729–1736.