Go-Blend Behavior and Affect

Matthew Barthet
Institute of Digital Games
University of Malta
Msida, Malta
matthew.barthet@um.edu.mt

Antonios Liapis
Institute of Digital Games
University of Malta
Msida, Malta
antonios.liapis@um.edu.mt

Georgios N. Yannakakis
Institute of Digital Games
University of Malta
Msida, Malta
georgios.yannakakis@um.edu.mt

***Abstract*—This paper proposes a paradigm shift for affective computing by viewing the affect modeling task as a reinforcement learning process. According to our proposed framework the context (environment) and the actions of an agent define the common representation that interweaves behavior and affect. To realise this framework we build on recent advances in reinforcement learning and use a modified version of the Go-Explore algorithm which has showcased supreme performance in hard exploration tasks. In this initial study, we test our framework in an arcade game by training Go-Explore agents to both play optimally and attempt to mimic human demonstrations of arousal. We vary the degree of importance between optimal play and arousal imitation and create agents that can effectively display a palette of affect and behavioral patterns. Our Go-Explore implementation not only introduces a new paradigm for affect modeling; it empowers believable AI-based game testing by providing agents that can blend and express a multitude of behavioral and affective patterns.**

***Index Terms*—Reinforcement Learning, Go-Explore, Arousal, Affective Computing, Artificial Agents, Gameplaying**

I. Introduction

Affective computing is traditionally viewed from an expert-domain and supervised learning lens through which manifestations of affect are linked to ground truth labels of affect that are provided by humans. Behavior and affect are either blended in the form of hand-crafted rules [1], [2] or machine learned via supervised learning methods [3]. While affect models designed or built this way are linked to the context of the interaction, they are often completely independent of the behavior of the involved actors.

A recent (non-deep) reinforcement learning (RL) algorithm, Go-Explore [4], showcased superb performance at hard exploration problems with many states—such as complex planning-based games—that most other deep learning methods struggled with. In its application to the game *Montezuma's Revenge* (Parker Brothers, 1984), Go-Explore reached super-human gameplaying performance. In part, this is achieved by storing all visited game states and exploring from such interim states rather than playing the game from the start [5]. Inspired by these recent breakthroughs in RL, we leverage the capacity of Go-explore to introduce a paradigm shift for affect modeling. We argue that viewing affect modeling as an RL process yields agents (or computational actors) that manage to reliably interweave behavior and affect without necessarily relying on affect corpora of massive sizes.

The proposed concept revolutionizes affective computing, which traditionally attempts to model human affect in the context of an interaction but largely ignores the affective response to the *actions* of the involved (inter-)actors. Both behavior and affect are blended in an internalised model that associates an agent's context (environment) and its actions to both its behavioral performance and its affective state. At the same time, we introduce a novel paradigm for RL where the rewards are not only tied to a user's behavior but combined with rewards from annotations provided by the users themselves (i.e. human affect demonstrations). According to our approach, both behavior and affect can form reward functions that can be experienced from RL agents that learn to behave and express affect in various ways. The proposed Go-Explore implementation is tested in a simple arcade game featuring a rich corpus of self-reported traces of arousal.

Our key findings suggest that agents can be trained effectively to behave in particular ways (e.g. play optimally with super-human performance) but also behave so as they *feel* as humans would in a particular game state. Beyond the proposed paradigm shift in affective computing, our Go-Explore agents offer insights on the relationship between affect and behavior through their RL trained models. Importantly, RL agents that blend behavior and affect can be used directly for believable testing as such agents can simulate and express simultaneously both behavioral and affective patterns of humans.

II. Background

This section provides a brief overview of the related domains of reinforcement learning, the Go-Explore algorithm, traditional affect modelling via imitation learning and affect modelling using reinforcement learning.

A. Reinforcement Learning and Go-Explore

Reinforcement learning approaches machine learning tasks from the perspective of behavioral psychology, mimicking the way animals and humans learn through receiving positive or negative rewards for their actions [6]. Exploring state spaces with sparse and/or deceptive rewards has been a core challenge for traditional RL algorithms, as they suffer from issues of detachment and derailment. *Detachment* occurs when an algorithm forgets how to return to previously visited promising

This project has received funding from the European Union's Horizon 2020 programme under grant agreement No 951911.

areas of the search space due to exploration in other areas. *Derailment* is a consequence of RL algorithms which do not separate returning to states from exploring the search space. This may result in potentially promising states that require a long sequence of precise actions unlikely to occur under exploratory conditions.

Go-Explore is a recent algorithm in the RL family [7] which is explicitly designed to overcome the two aforementioned challenges. The algorithm was introduced with the aim of improving RL performance in hard-exploration problems, which tend to contain sparse or deceptive rewards. Go-Explore has demonstrated previously unmatched performance in Atari games [5], highlighting its ability to thoroughly explore complex and challenging environments. In games with sparse rewards (such as *Montezuma's Revenge*), a large number of actions must be taken before a reward can be obtained, whereas deceptive rewards may mislead the agent and result in premature convergence and therefore poor performance [8]. Go-Explore has been used for text-based games, capable of outperforming traditional agents in *Zork1* [9] and is able to generalize to unseen text-based games more effectively [10]. The algorithm's capabilities have also been demonstrated in complex maze navigation tasks which could not be completed by traditional RL agents [11]. Beyond playing planning-based games with superhuman performance, Go-Explore has been used for autonomous vehicle control for adaptive stress testing [12], and as a mixed-initiative tool for quality-assurance testing using automated exploration [13]. While Go-Explore has proved to be a highly effective algorithm for behaviour policy search, it has never been tested on affect modeling tasks. This proof-of-concept paper introduces the first application of the algorithm for modeling affect as an RL process and blending it with behavior within a game agent.

B. Reinforcement Learning and Affective Computing

Traditionally, affect modelling [3] involves constructing a computational model of affect that takes as input the context of the interaction, such as pixels [14], [15], and multimodal information about a user—including physiological signals [16], facial expressions [17], [18] or speech [19]—and outputs a predicted corresponding emotional state (i.e. the ground truth of emotion). Given that affective computing relies on a provided ground truth of emotion that is human-annotated, affect detection is naturally viewed as a supervised learning task [3]. Traditionally a dataset of user state-affect pairs is used to train a model to predict affect [20]. Trained affect models are then used in conjunction with action selection methods for the synthesis, adaptation and affect-based expression of agents including virtual humans [21] and social believable agents [22].

Beyond the obvious uses of RL for learning a behavior policy, RL has been used as a paradigm for creative AI and, in particular, for the procedural generation of content (PCG) [23]. While the experience-driven PCG framework [24] considered the use of affect models beyond the behavior action space, its initial version never considered RL as a training paradigm for such generators. As a response, a recent study blended the frameworks of experience-driven PCG and PCG via reinforcement learning, namely ED(PCG)RL; EDRL in short [25] focuses on the use of RL for the algorithmic creation of content according to a surrogate model of player experience or affect.

Whilst there exist a variety of studies on the topic of agent emotion and reinforcement learning, literature on using human-annotated emotion as a training signal for learning is limited [26]. It has been shown that coupling an agent's simulated affect with its action-selection mechanism allows it to find its goal faster and avoid premature convergence to local optima [27]. Similarly, [28] showed that using affect as a form of social referencing is a simple method for teaching a robot tasks, such as obstacle avoidance and object reaching. Work on intrinsic motivation through the RL paradigm [29], [30] is also highly relevant to our aims. Intrinsic motivation studies by definition, however, ignore human demonstrations, behavioral and importantly affective [31]. A number of very recent studies (e.g. [32]) view the intrinsic motivation paradigm from an inverse RL lens through which reward functions are inferred from behavioral demonstrations.

The work in this paper expands upon the current state of the art by viewing affect modeling as an RL paradigm and explicitly blending agent behavior and affect using a cutting edge RL algorithm for hard exploration problems. The result is a set of agents which are tested in games in this initial study. The game agents trained to behave (i.e. play) optimally, even better than humans, and "feel" like a human would (via arousal imitation), or a blend of the two approaches with varying degrees of importance.

III. Blending Behaviour and Affect

This paper proposes combining rewards for good behavioral performance with rewards for affect matching in a reinforcement learning agent. We leverage the Go-Explore RL algorithm and describe our implementation in Section III-A and how it is enriched with affect information in Sections III-B and III-C.

A. Go-Explore Implementation

The Go-Explore algorithm builds on two phases to create a robust search policy that performs well under a specified reward scheme received from the environment. The first phase is the *exploration* phase, where a deterministic model of the environment is used to explore the search space thoroughly. During exploration an archive of the states encountered so far is used to ensure states are not forgotten, thus preventing the issue of derailment. Each state in the archive also contains the string of actions needed to return to it, addressing the issue of detachment and ensuring that all states can be visited. States are chosen using a selection strategy (e.g. randomly or through the UCB formula [33]), after which the algorithm returns to the state as described and begins exploring from there. At its simplest, exploration occurs by taking random actions and updating the cell archive with new states or updating existing

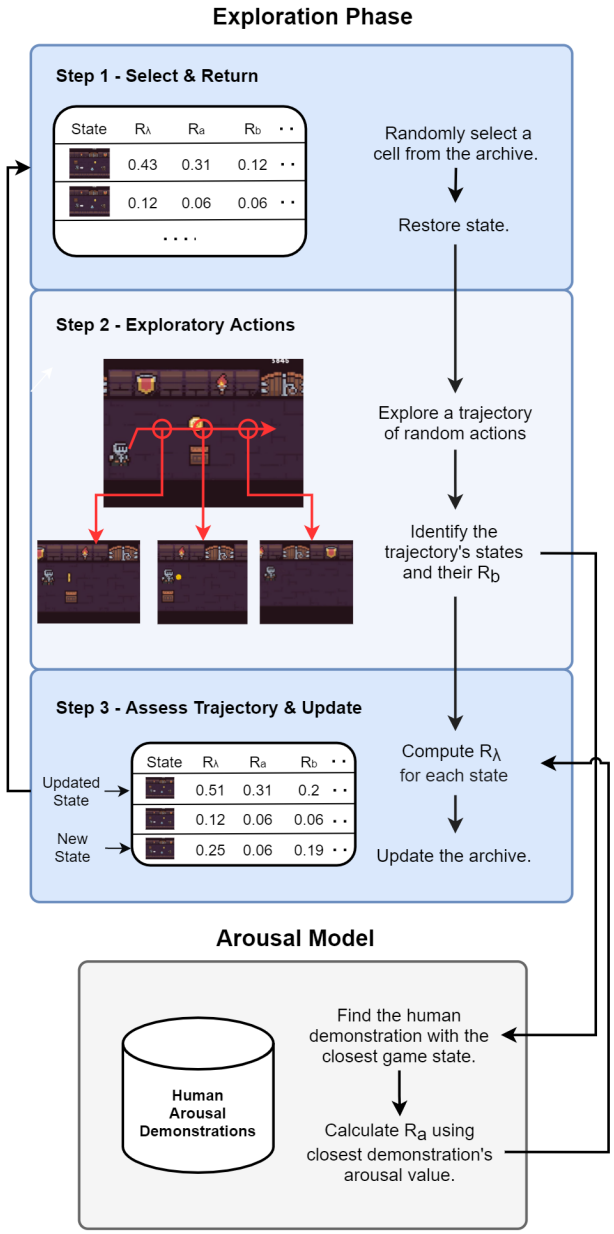


Fig. 1. A high-level overview of Go-Explore that blends agent behavior and affect.

ones with better reward values. The move selection strategy can be improved according to the nature of the environment being searched and through the use of expert knowledge.

The result of the exploration phase is a number of high performing trajectories using the deterministic model. If required, the *robustification* phase uses the "backward algorithm" [34] to train an agent to perform at the same level (or better) as the trajectories found in exploration, but in a stochastic setting. The backward algorithm is an RL technique used to learn from a given trajectory by decomposing the problem into smaller exploration tasks. It starts by placing the agent near the end of the trajectory and uses an off-the-shelf RL algorithm to train the agent to imitate its last segment. This is repeated several times, moving the starting point further back until the beginning of the trajectory is reached and the agent has been trained on the entire trajectory. To stabilize learning, Go-Explore extends this method to use multiple trajectories which are uniformly sampled at the beginning of each learning episode.

Our implementation of Go-Explore follows the original approach by Ecoffet *et al.* [5] (see Fig. 1). An archive of cells stores the game states that have been visited, with each cell representing a unique game state and containing the instructions needed to reach that point in the game. Each cell has an associated reward value, which is used to determine if the cell should be updated in case a similar state with a better score is found. Cells are chosen to explore from randomly, and the actions taken to build trajectories during exploration are also random. Along with the action trajectory to return to its state, each cell also contains trajectories for the state with accompanying cumulative behavioral and affect rewards per trajectory. This implementation of Go-Explore differs from the original version through the inclusion of affect (i.e. arousal in this study) in the reward function. Moreover, in this paper the robustification phase of Go-Explore is not carried out but will be explored in future work.

B. Arousal Model

A natural question arising when one is asked to blend behavior and affect within a learning process is how the two pieces of information will be considered and fused. An obvious requirement is that the human annotations of affect are time-continuous, thus providing moment-to-moment information about the change of affective states and aligning them with game states stored in playtraces.

One approach for calculating an affect reward would be to build *a priori* models of affect using supervised learning and use their predicted outcomes *indirectly* as affect-based reward functions. Instead, one could use the affect labels *directly* and build reward functions based on this information. Rather than relying on a trained surrogate model of arousal in a given state, our algorithm queries a dataset of human arousal demonstrations to find the arousal value of the human player closest to the current game state. We use the playtraces and their associated arousal traces directly to assess the player's arousal value in that state which, in turn, provides the intended arousal goal at this point in time.

C. Reward Function

The reward function used for this version of Go-Explore consists of two weighted functions for optimizing behavior and imitating human affect respectively. Both components are normalized within the range $[0, 1]$ to avoid uneven weighting between the two objectives. In particular the reward function used, R_λ, is as follows:

$$R_\lambda = \lambda \cdot R_a + (1 - \lambda) \cdot R_b \quad (1)$$

where R_a and R_b are the rewards associated with affect and behavior, respectively, and λ is a weighting parameter that

blends the two rewards. Formally, the reward associated to affect (i.e. arousal in this paper) is computed as follows:

$$R_a = \frac{1}{n}\sum_{i=0}^{n}(1 - |h(i) - a(i)|) \quad (2)$$

where i is a playtrace and affect annotation observation within a time-window; n is the number of observations made so far in this trajectory; $h(i)$ is the agent's estimated arousal value in its current game state; $a(i)$ is the arousal goal at this point in the game. In this paper, we derive $h(i)$ and $a(i)$ directly from human playtraces and their accompanied affect annotations (see Section III-B). Specifically we calculate $a(i)$ by first creating a *mean arousal trace* averaging all players' arousal values in the same timestamp: this creates a moment-to-moment arousal trace that captures the consensus of players (regardless of actual game context). $a(i)$ is then calculated by finding the arousal value of this mean arousal trace for that time window i. On the other hand, $h(i)$ is based on the agent's current game state, finding the annotated arousal value of a human playtrace at any timestamp which has an accompanying game state closest to the agent's game state.

R_a minimizes the absolute difference between the arousal value of a human player in a similar game state as the agent, and the mean annotated arousal value at this time window i. Since this difference is averaged across the number of observations made so far, it encourages trajectories with high imitation accuracy across the whole arousal trace generated.

The reward for the agent's behavior (R_b) depends on the game; in this paper we assume that the total score accumulated throughout the game is a sufficient reward for optimal behavior. This assumes that the environment follows arcade game tropes which are played for high-score, as is the case in our case study described in Section IV. In more complex games, or in games without an explicit score, the reward signal must be designed on an ad-hoc basis such as the reward function used in the original implementation of Go-Explore [5].

According to Eq. (1), if $\lambda = 0$ the reward function trains the agent to only maximise its score (i.e. optimize its behavior) and ignore its associated arousal trace. On the other hand, if $\lambda = 1$ the agent is trained to imitate human arousal and ignore its behavior.

IV. Case Study: Endless Runner

The proposed vision of blending arousal and performance rewards is tested in the "Endless Runner" game (hereafter *Endless*). Endless is a platformer game built using the Unity Engine and featured in the AGAIN dataset [35]. The game was chosen for its simple mechanics and objective, and for its accompanying dataset of 112 annotated human play sessions that can be easily used for the arousal model.

A. Game Description

In Endless, the player controls an avatar that constantly moves towards the right and must avoid or destroy obstacles that spawn in their path. The platform consists of two lanes (top/bottom) and the player's only controls is switching lanes by moving up or down (via keyboard input) and/or using a melee attack described below. Game objects are placed on one of two lanes (upper/lower) and are spawned at random intervals. Game objects include items that the player may collide with to improve their score (coins) or alter their movement speed (potions). Other game objects are obstacles (which include immobile enemies); the player must use their melee attack when in close proximity to the obstacle in order to clear it. Colliding with an obstacle results in a 10 point score penalty, destroys any nearby game objects on the screen, and resets the player's speed to the default value. Every 3 seconds the player is passively awarded a point to their game score on top of any bonus points they may receive for collecting coins. Every 10 seconds the speed of the player increases by a fixed amount, increasing the difficulty of the game. In theory, the game can be played for as long as the player wants. During data collection for the AGAIN dataset, an Endless session ended after exactly two minutes and the player has infinite lives. We follow the same duration in all experiments in this paper in order to leverage players' affect annotations and compare the agents' performance with human play.

Fig. 2. Endless Runner Game Layout

B. Go-Explore for Endless Runner

The game was converted into a deterministic environment to be compatible with the exploration phase of Go-Explore. The sequence of objects to be spawned and their spawn times was fixed to ensure the same sequence of game states are observed when replaying trajectories. Moreover, the game could start from any saved snapshot (i.e. any visited game state). This minimizes the time spent returning to a new cell's state and allows the algorithm to focus on exploration, an approach central to the Go-Explore paradigm [5]. To decide which cell the game state should be assigned to, the game state is mapped as an 8-parameter vector describing the player's current lane (two binary values, one per lane), and which game objects are on each lane at specific distance bands (near, mid-distance, and far). The possible values for these bands

are empty, item, or obstacle, and in case items and obstacles exist in the same band, it is treated as an obstacle band. The reward for optimal behavior (R_b) in Endless is the player's total score after an action is taken. This value is normalized between 0 and 1 with respect to the optimal score achievable in the play session. The optimal in-game score is calculated by summing two components. The first is the total amount of points awarded to the player passively over time for not dying during the game. The second is the maximum amount of bonus points achievable by picking up every coin in the deterministic environment.

TABLE I
RESULTS FOR ENDLESS AVERAGED FROM 5 RUNS AND INCLUDING THE 95% CONFIDENCE INTERVALS.

Experiment Setup	**Performance Measures**		
	R_b	R_a	R_λ
$R_{0.0}$	0.79 ($\pm$0.0474)	0.72 ($\pm$0.0126)	0.79 ($\pm$0.0474)
$R_{0.25}$	0.73 ($\pm$0.0818)	0.73 ($\pm$0.0181)	0.73 ($\pm$0.0569)
$R_{0.5}$	0.74 ($\pm$0.0741)	0.74 ($\pm$0.0145)	0.74 ($\pm$0.0311)
$R_{0.75}$	0.69 ($\pm$0.0658)	0.76 ($\pm$0.0147)	0.74 ($\pm$0.0082)
$R_{1.0}$	0.25 ($\pm$0.1335)	0.79 ($\pm$0.0056)	0.79 ($\pm$0.0056)
Random	0.03 ($\pm$0.1012)	0.75 ($\pm$0.0074)	N/A
Human	0.70 ($\pm$0.0467)	0.77 ($\pm$0.0131)	N/A

C. Experimental Protocol

Reported results per method are averaged across five independent runs of the Go-Explore algorithm. Each run consists of the exploration phase of Go-Explore (there is no robustification phase in this first experiment), and the agent returns and explores 4,000 times before selecting the best trajectory and saving it. The agent explores a maximum of 20 actions before choosing a new state to explore from. The actions taken during exploration are chosen at random among the 6 possible options (move up or down, move up or down and attack, no action and attack). The new state to explore from is chosen at random among those already discovered: the reward of the state in the archive, or the number of times it has been visited is not considered. The best trajectories are saved and can be used for the robustification phase of Go-Explore in future work.

The λ parameter of Eq. 1 was varied to observe the relationship between learning to play the game optimally and learning to imitate human annotated arousal. Table I shows the five values used for the λ parameter, ranging from 0 to 1 in increments of 0.25. Recall that at $\lambda = 0$ and $\lambda = 1$ the agent tries to learn to solely behave optimally or to solely "feel" like a human respectively. As a baseline, an experiment with an agent that performed random actions was carried out and results are averaged from 5 independent runs. To estimate this random agent's arousal levels, a trace was generated based on the game states visited using the same approach as in the Go-Explore experiments.

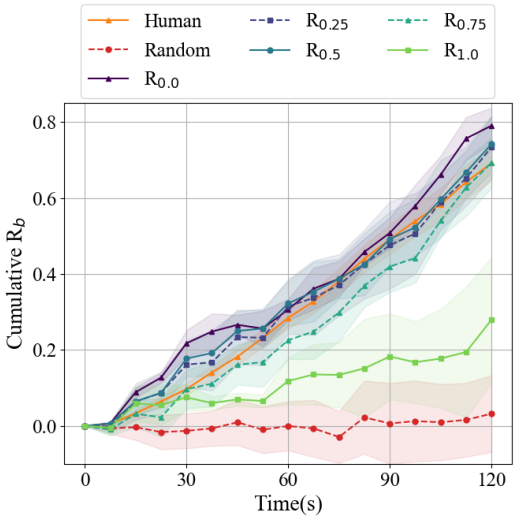
The results were compared to the average performance seen by humans in the dataset for both behavior and arousal reward functions. All results given are the average observed across the 5 runs of Go-Explore, paired with the 95% confidence interval.

D. Results

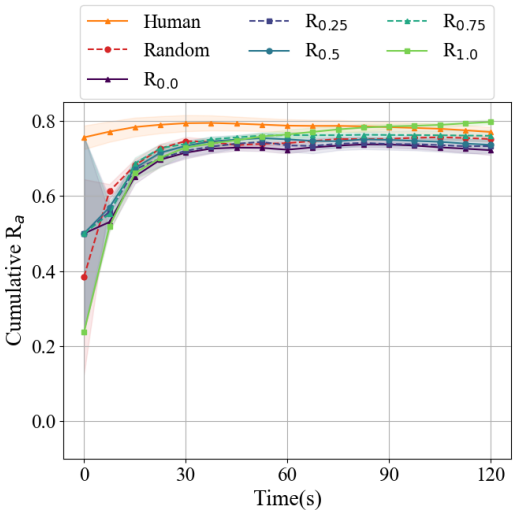
Table I shows the final values observed for the cumulative behavior (R_b) and arousal (R_a) components, as well as the overall reward function (R_λ) for each experiment. Note that the baseline agent and human entries are not included in the R_λ column as they were not trained using Go-Explore. Figure 3c illustrates how the agents' overall cumulative reward fluctuates over time for each Go-Explore configuration. Note that due to different λ values, the R_λ values across experiments are not comparable but the differences in how it fluctuates over time provides insight into the behavior of the algorithm. It is clear that agents with higher priority assigned to arousal imitation tend to converge to their maximum value quicker due to the nature of the arousal reward function. Since at $R_{0.0}$ the total reward amounts to a normalized measure of the agent's in-game score, it is not surprising that high scores are only attainable at late points in the game. Instead, states that match the mean arousal trace seem to be easily discovered even early in the game.

Looking at the results for the behavioral component (i.e. the total game score normalized to the absolute best possible score), the random agent shows the worst performance as one would expect when playing most games. While the exploration phase of Go-Explore relies on a random sequence of actions, the discovery of interim states (cells) to explore from and the optimization of these states based on R_λ clearly leads to a more efficient playstyle than random. For $R_{1.0}$, the agent still manages to produce a better score than the random agent but remains significantly lower than the average human player. Random and $R_{1.0}$ also display a wider confidence interval compared to the rest of the experiments, pointing to an inconsistent behavior. When the behavior component is introduced with a small weight (e.g. $R_{0.75}$), the score immediately matches that of the average human demonstration. As λ is lowered to zero, the agent's score improves and surpasses human levels of performance. Figure 3a illustrates how the agents' cumulative behavior reward changes over time for each configuration. As noted above, the cumulative behavior reward is very time-dependent by design (players reach higher scores the longer they play) but clearly the random agent (and $R_{1.0}$ to a degree) tends to lose score by hitting obstacles which seems to perfectly offset passive score gains.

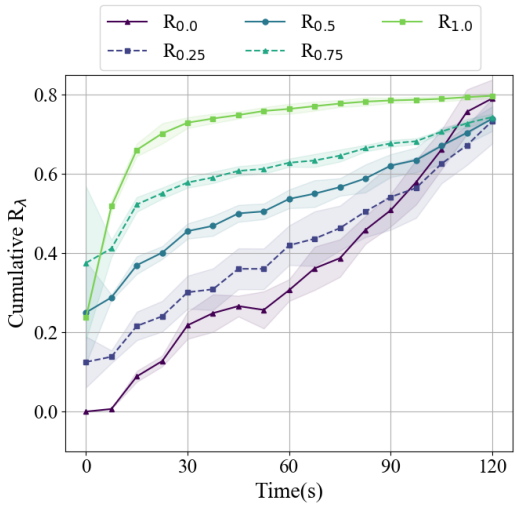
The results for the arousal component tell a similar story to the results for behavior, with the exception of the random agent. Unsurprisingly, the arousal score increases as λ increases from 0 to 1. What is surprising however is the arousal scores attained by the random agent, which seem to be almost at the same levels as the human trace and is only significantly surpassed by $R_{1.0}$. The potential reasons for this are discussed in section V. Figure 3b illustrates how the agents' cumulative arousal reward changes over time for each configuration. It is evident that unlike R_b which is tied to the game score, it is easy to attain high values in R_b early on, and it is also easy to maintain the same levels throughout the game even when performing random actions.



(a) Cumulative Behavior Reward (R_b)



(b) Cumulative Arousal Reward (R_a)



(c) Cumulative Total Reward (R_λ)

Fig. 3. Cumulative rewards averaged from 5 independent runs. Shaded areas denote the 95% confidence interval.

V. Discussion

This paper envisions how affect modeling and expression can be realised thought the RL paradigm. In particular, we investigate how arousal traces can be used as human demonstrations that train a gameplaying agent to learn how to feel like a human. In the simple testbed of Endless Runner, the large number of annotated playtraces allowed us to match an agent's game state to a human player's game state and use the player's annotated arousal level directly. Results indicate that, as expected, updating the cells of Go-Explore based on the agent's in-game performance (a normalized version of the game score) leads to optimal behavior policies that surpass the average human scores. Combining this performance-based reward with an arousal-based reward that aimed to mimic human annotations resulted in a minor drop of performance which, nevertheless, remains human-competitive. Evidently, using this arousal-based policy alone was detrimental to gameplay performance and points to some limitations of the current way that R_a of Eq. 2 is calculated.

The fact that for all agents, including the random action baseline, the cumulative arousal reward swiftly reached high values points to a task that is overly easy. It seems that deriving a policy only based on R_a does not motivate the agent to explore many different states, although the number of updates or new cells encountered in Go-Explore has not been studied sufficiently to verify this hypothesis. Moreover, it should be noted that the human annotations of arousal were processed in an unbounded, ordinal fashion and normalized after the fact. While most players follow a similar pattern of increasing arousal as the game goes on, using the numerical difference between one human's arousal value (closest to the agent's game state) and the mean could reintroduce subjectivity biases due to the normalization applied. Designing another reward function for arousal that better matches the ordinal nature of affect [36], [37] would be an important direction for future work. Finally, both performance and affect rewards are measured cumulatively, in part due to the fact that the former is the player's score. Exploring different variants by e.g. averaging either score increase or arousal similarity across a narrower time window is expected to have an effect on agents' performance.

It is also worth noting that the Endless Runner testbed has a low branching factor and a deterministic game state. Therefore each experiment was subjected to a very similar sequence of game states. While the simple game still showed that performance-based optimization via Go-Explore is vastly superior to a random agent, it may have affected the arousal model in unexpected ways. Due to the few visited game states, it is likely that the range of values that could be returned by the arousal model was small, which is a likely cause for the agents' similar arousal accuracy and small confidence intervals across the board. Furthermore, when identifying the closest human for the arousal model, a relatively small subset of sessions are used for computational efficiency which further limits the range of arousal values that could be observed. Changing the

approach for deriving an arousal value for a given state with this limitation in mind would help generate more diverse traces and allow the differences in the reward functions to become more pronounced. A more complex game where the agent has more degrees of freedom and more arousing stimuli for the human playtesters will also likely illuminate the strengths and weaknesses of this approach.

This proof of concept opens up several avenues for future work to further explore the relationship between behavior and affect in the context of reinforcement learning. Obvious next steps have been highlighted above in terms of refining the arousal reward function and testing the approach in more complex, more stimulating games. Another direction is testing machine-learned predictors of affective states rather than the direct mapping to the closest human trace performed currently. While surrogate models are often inexact, it may counteract the sparse game states encountered by human players when matching an unseen state. More importantly, incorporating the robustification phase in the Go-Explore algorithm is expected to lead to new insights on the impact of affect-based rewards, especially since the environment will no longer be deterministic and thus many more game states are likely to be visited. Finally, imitating human behavior (as a form of reward function) can reveal interesting new relationships between the human-like behavior and affect and optimal play; such derived policies would likely allow agents to play (near) optimally, whilst attempting to imitate both human behavior and human affect.

VI. Conclusion

This paper presents a proof of concept implementation of a new reinforcement learning paradigm for affective computing where behavioral and affective goals are interwoven. We leverage the Go-Explore algorithm due to its cutting edge ability to solve hard exploration problems, and we pair it with a set of reward functions that blend optimal behavior with arousal imitation to different degrees. Using the Endless Runner game as a platform to test the implementation, we were able to make use of an extensive dataset of human play sessions and accompanying arousal demonstrations that guided the agent's policy. While this initial study focused on a single, simple game, the next steps of our investigations include the enhancement of the Go-Explore approach to cater for its robustification phase, the introduction of ordinal reward functions, and the extension of the approach to accommodate more complex environments within and beyond games.

References

[1] Stacy Marsella, Jonathan Gratch, Paolo Petta, et al., "Computational models of emotion," *A Blueprint for Affective Computing-A sourcebook and manual*, vol. 11, no. 1, pp. 21–46, 2010.

[2] Stacy C Marsella and Jonathan Gratch, "Ema: A process model of appraisal dynamics," *Cognitive Systems Research*, vol. 10, no. 1, pp. 70–90, 2009.

[3] Rafael A Calvo and Sidney D'Mello, "Affect detection: An interdisciplinary review of models, methods, and their applications," *IEEE Transactions on Affective Computing*, vol. 1, no. 1, pp. 18–37, 2010.

[4] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune, "Montezuma's revenge solved by go-explore, a new algorithm for hard-exploration problems (sets records on pitfall, too)," *Uber Engineering Blog*, 2018.

[5] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune, "First return, then explore," *Nature*, vol. 590, no. 7847, pp. 580–586, 2021.

[6] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[7] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune, "Go-explore: a new approach for hard-exploration problems," *arXiv preprint arXiv:1901.10995*, 2019.

[8] Georgios N. Yannakakis and Julian Togelius, *Artificial Intelligence and Games*, Springer, 2018, http://gameaibook.org.

[9] Prithviraj Ammanabrolu, Ethan Tien, Zhaochen Luo, and Mark O. Riedl, "How to avoid being eaten by a grue: Exploration strategies for text-adventure agents," *arXiv preprint arXiv:2002.08795*, 2020.

[10] Andrea Madotto, Mahdi Namazifar, Joost Huizinga, Piero Molino, Adrien Ecoffet, Huaixiu Zheng, Alexandros Papangelis, Dian Yu, Chandra Khatri, and Gokhan Tur, "Exploration based language learning for text-based games," *arXiv preprint arXiv:2001.08868*, 2020.

[11] Guillaume Matheron, Nicolas Perrin, and Olivier Sigaud, "PBCS: Efficient exploration and exploitation using a synergy between reinforcement learning and motion planning," in *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 2020, pp. 295–307.

[12] Mark Koren and Mykel J. Kochenderfer, "Adaptive stress testing without domain heuristics using go-explore," *arXiv preprint arXiv:2004.04292*, 2020.

[13] Kenneth Chang, Batu Aytemiz, and Adam M Smith, "Reveal-more: Amplifying human effort in quality assurance testing using automated exploration," in *Proceedingos of the IEEE Conference on Games*, 2019.

[14] Konstantinos Makantasis, Antonios Liapis, and Georgios N. Yannakakis, "From pixels to affect: A study on games and player experience," in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 2019.

[15] Konstantinos Makantasis, Antonios Liapis, and Georgios N Yannakakis, "The pixels and sounds of emotion: General-purpose representations of arousal in games," *IEEE Transactions on Affective Computing*, 2021.

[16] Hector P Martinez, Yoshua Bengio, and Georgios N Yannakakis, "Learning deep physiological models of affect," *IEEE Computational intelligence magazine*, vol. 8, no. 2, pp. 20–33, 2013.

[17] Adria Ruiz, Ognjen Rudovic, Xavier Binefa, and M. Pantic, "Multi-instance dynamic ordinal random fields for weakly supervised facial behavior analysis," *IEEE Transactions on Image Processing*, vol. 27, pp. 3969–3982, 2018.

[18] R. Walecki, Ognjen Rudovic, V. Pavlovic, B. Schuller, and M. Pantic, "Deep structured learning for facial action unit intensity estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5709–5718.

[19] George Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, Mihalis A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5200–5204.

[20] Sander Koelstra, C. Mühl, M. Soleymani, Jong-Seok Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis ;using physiological signals," *IEEE Transactions on Affective Computing*, vol. 3, pp. 18–31, 2012.

[21] William R Swartout, Jonathan Gratch, Randall W Hill Jr, Eduard Hovy, Stacy Marsella, Jeff Rickel, and David Traum, "Toward virtual humans," *AI Magazine*, vol. 27, no. 2, pp. 96–96, 2006.

[22] W Scott Reilly, "Believable social and emotional agents.," Tech. Rep., Carnegie-Mellon Univ Pittsburgh pa Dept of Computer Science, 1996.

[23] Ahmed Khalifa, Philip Bontrager, Sam Earle, and Julian Togelius, "Pcgrl: Procedural content generation via reinforcement learning," *arXiv preprint arXiv:2001.09212*, 2020.

[24] Georgios N Yannakakis and Julian Togelius, "Experience-driven procedural content generation," in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*. IEEE, 2015, pp. 519–525.

[25] Tianye Shu, Jialin Liu, and Georgios N Yannakakis, "Experience-driven PCG via reinforcement learning: A Super Mario Bros study," in *Proceedings of the IEEE Conference on Games*, 2021.

[26] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker, "Emotion in reinforcement learning agents and robots: a survey," *Machine Learning*, vol. 107, no. 2, pp. 443–480, 2018.

[27] Joost Broekens, Walter A Kosters, and Fons J Verbeek, "On affect and self-adaptation: Potential benefits of valence-controlled action-selection," in *International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer, 2007, pp. 357–366.

[28] Cyril Hasson, Philippe Gaussier, and Sofiane Boucenna, "Emotions as a dynamical system: the interplay between the meta-control and communication function of emotions," *Paladyn*, vol. 2, no. 3, pp. 111–125, 2011.

[29] Satinder Singh, Andrew G Barto, and Nuttapong Chentanez, "Intrinsically motivated reinforcement learning," Tech. Rep., Massachusetts University, Amherst Department of Computer Science, 2005.

[30] Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 70–82, 2010.

[31] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas, "Social influence as intrinsic motivation for multi-agent deep reinforcement learning," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2019, pp. 3040–3049.

[32] Léonard Hussenot, Robert Dadashi, Matthieu Geist, and Olivier Pietquin, "Show me the way: Intrinsic motivation from demonstrations," *arXiv preprint arXiv:2006.12917*, 2020.

[33] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[34] Tim Salimans and Richard Chen, "Learning montezuma's revenge from a single demonstration," *arXiv preprint arXiv:1812.03381*, 2018.

[35] David Melhart, Antonios Liapis, and Georgios N. Yannakakis, "The Affect Game AnnotatIoN (AGAIN) dataset," *arXiv preprint arXiv:2104.02643*, 2021.

[36] Georgios N Yannakakis, Roddy Cowie, and Carlos Busso, "The ordinal nature of emotions," in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*. IEEE, 2017, pp. 248–255.

[37] Georgios N Yannakakis, Roddy Cowie, and Carlos Busso, "The ordinal nature of emotions: An emerging approach," *IEEE Transactions on Affective Computing*, vol. 12, no. 1, pp. 16–35, 2018.