Simulating Player Behavior for Data-Driven Interactive Narrative Personalization

Pengcheng Wang, Jonathan Rowe, Wookhee Min, Bradford Mott, James Lester

Department of Computer Science, North Carolina State University, Raleigh, NC 27695
{pwang8, jprowe, wmin, bwmott, lester}@ncsu.edu

Abstract

Data-driven approaches to interactive narrative personalization show significant promise for applications in entertainment, training, and education. A common feature of data-driven interactive narrative planning methods is that an enormous amount of training data is required, which is rarely available and expensive to collect from observations of human players. An alternative approach to obtaining data is to generate synthetic data from simulated players. In this paper, we present a long short-term memory (LSTM) neural network framework for simulating players to train data-driven interactive narrative planners. By leveraging a small amount of previously collected human player interaction data, we devise a generative player simulation model. A multi-task neural network architecture is proposed to estimate player actions and experiential outcomes from a single model. Empirical results demonstrate that the bipartite LSTM network produces the better-performing player action prediction models than several baseline techniques, and the multi-task LSTM derives comparable player outcome prediction models within a shorter training time. We also find that synthetic data from the player simulation model contributes to training more effective interactive narrative planners than raw human player data alone.

Introduction

Data-driven approaches to personalized interactive narrative generation have been the subject of growing interest. A broad range of machine learning techniques have shown considerable promise for improving interactive narrative planners' capacity to personalize stories and generate novel narrative scenarios. Yu and Riedl (2014) proposed a prefix-based collaborative filtering approach to predict player ratings of story branches to drive interactive narrative generation. Lee et al. (2014) used dynamic Bayesian networks to model director agent decisions in educational interactive narratives. Crowdsourcing approaches have also been examined to derive plot graphs for generating stories from prior players' input (Li et al. 2013; Harrison and Riedl 2016). More recently, reinforcement learning techniques have been used to personalize adaptable event sequences in narrative-centered learning environments (Rowe et al. 2014; Wang et al. 2016).

The limited availability of high-quality training data is a key gating factor in devising scalable data-driven interactive narrative generation techniques. High-quality data's limited availability is often an obstacle for developers seeking to use data-driven techniques. In many cases, collecting data by conducting large-scale studies with human players is resource-intensive. In these cases, utilization of player simulation models, which imitate human player behaviors to assist in training data-driven interactive narrative planners, shows particular promise.

In this paper, we present a long short-term memory (LSTM) network framework for devising simulated players that assist with training interactive narrative planners. We compare a multi-task neural network (NN) architecture with a bipartite architecture to predict player actions and experiential outcomes. LSTMs enable the player simulation model to compactly encode players' narrative interaction histories, and the multi-task NN architecture enables lower level feature sharing between player action and experiential outcome predictions tasks in the NNs. Experimental results indicate that bipartite LSTMs improve accuracy for predicting player actions relative to several competitive baselines and the multi-task NN architecture improves the training speed for LSTM-based player experiential outcome prediction using data from an educational interactive narrative, CRYSTAL ISLAND (Rowe et al. 2011). In addition, results suggest that the player simulation model assists in training better-performing reinforcement learning (RL)-based interactive narrative planners compared to RL-based interactive narrative planners trained with only human player data.

Related Work

A broad range of computational techniques and data sources have been utilized for training data-driven interactive narrative planners. Nelson et al. (2006) investigated temporal-difference methods for training a drama manager for a text-based interactive fiction, *Anchorhead*. Synthetic player data was utilized with the assumption that players would behave either cooperatively or adversarially with respect to the drama manager's decisions. Roberts et al. (2006) proposed target-trajectory distribution Markov decision process models for interactive narrative generation, which drive drama manager decisions toward an author-specified target distribution over narrative trajectories. Crowdsourcing techniques have also shown promise for training interactive narrative planners. Li et al. (2013) utilized crowdsourced data to generate plot-graph based narrative models. This framework was extended by Harrison and Riedl (2016) to learn reward functions for an RL system that controls virtual agents in the interactive narrative *Robbery World*.

Player modeling is another approach for capturing the dynamics of player behavior to support data-driven interactive narrative planning. Thue et al. investigated PaSSAGE, an interactive narrative framework that dynamically selects interactive story content by estimating players' gameplay styles based upon Robin's Laws (Thue et al. 2007). Yu and Riedl (2014) also applied Robin's Laws to synthesize simulated users in the verification of collaborative filtering techniques for interactive narrative generation.

Raw player data—these typically consist of human players' game log files and questionnaire responses—have been utilized as training data sources in several interactive narrative planners. Lee et al. (2014) conducted Wizard-of-Oz studies to gather training data for devising dynamic Bayesian network-based interactive narrative planners. Rowe et al. (2014) investigated a modular RL framework for generating personalized interactive narratives using data from human players. More recently, the same dataset was used to investigate alternate decompositional representations for modular RL-based interactive narrative personalization (Wang et al. 2016).

Devising player simulations from human player data is a promising approach to training data-driven interactive narrative planners, but to date this has not been widely studied. An exception is recent work by Wang et al. (2017), which investigated a bipartite deep learning-based player simulation model to synthesize training data for a deep RL-based interactive narrative planner. We examine an alternate version of this framework by investigating a multi-task NN architecture to perform player action prediction and experiential outcome prediction simultaneously.

Multi-task NN architectures have not been previously investigated in player simulation or interactive narrative generation, but they have shown promise in other applications. For example, multi-task convolutional NNs have proven to be successful in multi-domain visual tracking (Nam and Han 2016) and automated image analysis (Fourure et al. 2017). Lample and Chaplot (2016) applied the same technique to enhance deep RL's ability to play FPS games. These examples motivate our hypothesis that multi-task deep NNs can be an effective approach for player simulation.

Data-Driven Interactive Narrative Personalization in CRYSTAL ISLAND

CRYSTAL ISLAND Testbed Environment

To investigate the performance of the player simulation models, as well as their effectiveness in training data-driven interactive narrative planners for story personalization, we utilize CRYSTAL ISLAND, a narrative-centered educational game for middle school science. CRYSTAL ISLAND features a science mystery about an infectious outbreak on a remote island (Figure 1). The player adopts the role of a medical detective who must determine the source and identity of the illness by exploring a virtual open world, conversing with virtual characters, reading virtual books, conducting tests in a virtual laboratory, taking in-game quizzes, and completing an in-game diagnosis worksheet to solve the mystery.

Figure 1. CRYSTAL ISLAND interactive narrative.

In CRYSTAL ISLAND, an interactive narrative planner can dynamically tailor a player's story experience at run-time in several ways, such as adapting in-game dialogues between the player and non-player characters (NPCs), providing feedback on player performance, or guiding the player toward in-game resources that might help with solving the mystery. These decisions can be made at run-time to personalize the gameplay experience to individual players' preferences and needs. In this work, following the experimental design in (Wang et al. 2017), we focus on four

recurring adaptable events in CRYSTAL ISLAND: (1) how the NPC Teresa describes her symptoms during an in-game dialogue with the player, (2) how the NPC Bryce describes his symptoms during an in-game dialogue with the player, (3) how much feedback the player receives after a failed attempt at diagnosing the outbreak, and (4) whether the NPC Kim delivers an in-game quiz for the player to take.

Because of the educational design objectives of CRYSTAL ISLAND, we utilize normalized learning gain (Marx and Cummings 2007) to assess the quality of players' interactive narrative experiences. Players typically complete pre- and post-tests when participating in classroom studies with CRYSTAL ISLAND. Normalized learning gain (NLG) is the normalized difference between player's post-test score and pre-test score. In our analysis, we group player experiences into two categories: high NLG and low NLG. Players with NLG scores above or equal to the median value are in the high NLG group, and players with NLG scores below the median are labeled with low NLG. Although NLG is adopted in this study, other experience metrics (e.g., engagement questionnaire scores) could also be employed.

The dataset used to investigate player simulation models for CRYSTAL ISLAND is from two human subject studies with 453 students from two public middle schools. During both studies, students either played the game until they solved the mystery, or 55 minutes had elapsed, whichever occurred first. An interactive narrative planner that followed a uniform random policy for controlling adaptable events in CRYSTAL ISLAND was deployed to broadly sample the space of planning policies; the adaptable events were designed in such a manner to avoid coherence conflicts in the generated narratives. From these two studies, we collected data on players' gameplay action sequences, players' traits, players' interaction history with the narrative planner, and their pre- and post-test outcomes.

Reinforcement Learning-Based Interactive Narrative Planning

Reinforcement learning provides a natural computational framework for modeling interactive narrative planning as a sequential decision-making task with delayed rewards, i.e., experiential outcomes. Utilizing the abstraction of *adaptable event sequences,* as in (Rowe et al. 2014), we represent interactions between the narrative planner and player as a series of stochastic state changes, which are influenced by the planner's run-time adaptations to CRYSTAL ISLAND's interactive narrative, and which drive players' experiential outcomes as measured by normalized learning gains.

More formally, when a player conducts a series of player actions and triggers an adaptable event *e* at interactive narrative planning time step *t*, the interactive narrative planner chooses an action a_t^e from a discrete action set $A^e = \{a^{e_1}, a^{e_2}, \dots, a^{e_m}\}$ of event *e.* Decisions about adaptable narrative events are driven by planning policy π and the current narrative interaction state $s_t \in S = (o_{t-n+1}, \dots, o_t)$, in which o_t is the observation at interactive narrative planning time step *t*, and *n* is the number of observations encoded in the state representation. The interactive narrative environment proceeds to the state s_{t+1} and reward signal r_t is administered according to a narrative experience quality metric. Training RL-based interactive narrative planners gradually adjusts the interactive narrative planning policy π in order to optimize the expected discounted cumulative reward $R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau$ obtained by the narrative planner, where the discount factor $\gamma \in [0,1]$. The output of RL training is an optimal policy π^*, which encodes the optimal narrative planning action to perform in each state s.

Player Simulation in Interactive Narrative

In interactive narratives with open-world virtual environments, such as CRYSTAL ISLAND, players actively drive the narrative forward by performing actions in the virtual world. However, open-world environments afford many possible narrative trajectories—players may follow different sequences of conversing with NPCs, completing in-game sub-tasks, or interacting with virtual objects—and each player is likely to experience his or her own unique trajectory, albeit with similarities to peers' experiences.

To simulate player behavior in CRYSTAL ISLAND, we devise a model that predicts the next player action at each time step using prior gameplay history and player trait data as input. In addition, the player simulation model predicts the player's experiential outcome at the conclusion of the narrative episode. Specifically, synthetic data is generated as follows: an initial simulation state is generated by sampling from the human player initial states' probability distribution. Next, the simulated player is used to determine the distribution over likely player actions at the next time step. One action is sampled according to this distribution, and the player simulation's current state is updated according to the effects of the synthetic player action. If the player action triggers an adaptable event sequence, then a narrative adaptation decision by the planner occurs, once again updating the simulation's current state. This process continues until a game-ending action is generated. Afterward, the simulated player's experiential outcome is predicted.

We frame both player action prediction and outcome prediction as classification problems. Because player actions in CRYSTAL ISLAND can be represented in terms of a discrete player action set, player action prediction can be formalized as a multi-class classification problem. In CRYSTAL ISLAND, we concentrate on 15 types of player actions (including the game-ending action), which collec-

tively capture the different ways players explore CRYSTAL ISLAND's interactive narrative. As described above, because the interactive narrative testbed was designed for educational purposes, we focus on two types of player outcomes: high learning outcomes (represented with high normalized learning gain) and low learning outcomes (represented with low normalized learning gain).

The input features to the player simulation model are designed to represent key player attributes, how the player interacts with the virtual environment, as well as how the interactive narrative planner adapts events to shape the player's experience. Accordingly, we design an input representation that consists of three groups of features. The first group contains features consisting of accumulated counts of player actions (except the game-ending action). The second group consists of player trait information, such as prior gameplay experience, prior content knowledge, and gender. The third group consists of details regarding the planner's most recent narrative adaptation decisions. In combination, these three groups comprise a 21-feature input vector, in which 14 features encode the player's action history, 3 features encode player traits, and 4 features encode the interactive narrative planner's past decisions.

In our work, we expect the simulated player to not only interact with the interactive narrative planner, but also to emulate how human players behave, providing abundant realistic data to enhance the training of an interactive narrative planner compared to training with a limited corpus of human player data. This objective can be achieved by leveraging logical rules from the interactive narrative environment during synthetic data generation.

To illustrate, we first consider an RL-based interactive narrative planner trained directly from human player data. Adaptable event sequences in CRYSTAL ISLAND are triggered when certain conditions are met in the virtual storyworld. Adaptable events, or actions taken by the interactive narrative planner, occur less frequently than player actions (Figure 2). If an RL-based interactive narrative planner is trained directly with human player data, the planner can only model the sequence of state transitions observed along the *Interactive Narrative Planner Action Timeline*, yielding a coarse-grained simulation that abstracts away details of how player action sequences form RL states and trigger adaptable events. This is illustrated in Figure 2: on the *Interactive Narrative Planner Action Timeline*, the state transition from t_1 to t_2 describes the effect of applying interactive narrative planner action $a_{t_1}^{e_{t_1}}$ in the narrative environment. An RL-based planner would fit a transition function consistent with the distribution $P(s_{t_2}|s_{t_1}, a_{t_1}^{e_{t_1}})$, either explicitly or implicitly depending on the RL algorithm. The series of events denoted in the *Player Action Timeline* u_2 to u_7 are ignored, losing contextual detail contained in the player action sequence, and the potential constraints in the logical rules of the interactive narrative environment.

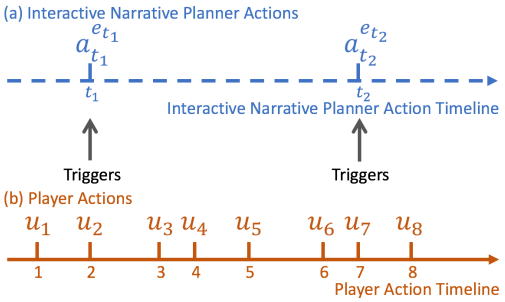

Figure 2. Example interactive narrative planner action timeline and player action timeline.

In contrast, in our player simulation model, when synthetic player behaviors and adaptable events are generated along the *Player Action Timeline*, the player simulation model not only generates behaviors following the patterns learned from the raw human player data, but it also exploits logical rules of the interactive narrative to avoid synthesizing impossible trajectories. By adopting an expressive player simulation model, extra prior knowledge from the rules of the virtual environment can be added in approximating the transition function $P(s_{t_2}|s_{t_1}, a_{t_1}^{e_{t_1}})$.

Another approach could be modeling narrative planner's behavior using RL along the *Player Action Timeline* by defining extra no-operation action for the narrative planner. However, this design can drastically delay the reward signal by increasing the interaction trajectory length, and result in a noisier state transition probability distribution from the additional less meaningful transitions.

LSTM Network-Based Player Simulation

Interactive narrative experiences are fundamentally sequential in nature, and this sequence data may contain useful information for predicting future player actions and experiential outcomes. To restrict the size of the feature space for player simulation, input feature sets in prior work have often not contained sequence information about narrative events. However, recurrent neural networks (RNNs) offer a convenient way to compactly encode a player's sequential interaction history and apply it for prediction.

In an RNN, the current hidden layer receives the output of the hidden layer in the preceding time step, and it propagates the current hidden layer output to the next time step. This property allows the player simulation model to represent the player's sequential behaviors using only the accumulated count information at each time step while still keeping the feature space compact. To overcome the vanishing gradient problem in standard RNNs, we adopt a broadly utilized RNN technique: long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997). Specifically, we investigate an LSTM architecture that utilizes three gating units following (Graves 2012). Input and output gates modulate the incoming and outgoing sig-

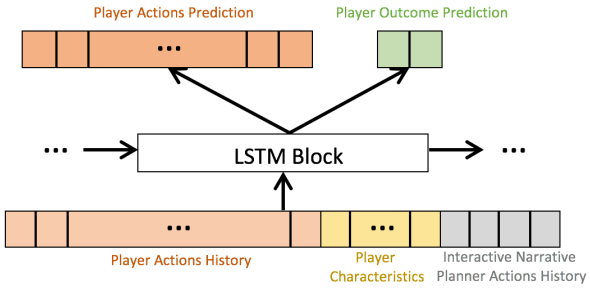

Figure 3. Multi-task LSTM network implementation of a player simulation for interactive narrative planning.

nals to the memory cell, and a forget gate controls whether the previous memory cell state is remembered or forgotten. In our player simulation model, player action prediction can be framed as a sequence-to-sequence LSTM, meaning that each player state input at one time step has an output of the next action. In contrast, player outcome prediction is formed as a sequence-to-one LSTM because predicted outcome is measured once at the end of each interaction sequence.

Multi-Task Neural Network Architecture for Player Simulation

Player action prediction and player outcome prediction are distinct but related tasks. In interactive narrative personalization, players with positive experiential outcomes may exhibit different patterns of actions than players with negative experiential outcomes. We exploit this relationship by utilizing a multi-task NN architecture for player simulation, which integrates player action prediction and player outcome prediction into a single model.

Deep neural networks provide a convenient structure for multi-task models. Because deep neural networks learn multi-level hierarchical features across several hidden layers, it is possible for multi-task deep neural networks to utilize shared lower layers with distinct output layers targeting each task. This architecture enables the model to leverage cross-task information by sharing abstract features. During training, errors from each output layer are backpropagated into the shared lower layers. Consequently, extraction of hierarchical features in the shared hidden layers is determined by multiple sources in a multi-task NN architecture. Figure 3 illustrates a multi-task LSTM model for predicting player actions and player outcomes simultaneously. For the sequence-to-one LSTM used for player outcome prediction, only the error from the last time step of one input sequence is backpropagated.

Evaluation

Two sets of experiments were conducted to evaluate how accurately the proposed player simulation models predict player actions and player outcomes, as well as how effectively the simulated players assist in training an RL-based interactive narrative planner. After removing incomplete records, data from 402 players were included in the corpus. In this dataset, each interactive narrative trajectory consisted of approximately 41 player actions centered around 15 player action types and 8 adaptable event occurrences on average. Among the 402 interactive narrative episodes, 200 were labeled as high NLG (i.e., high learning gains on educational outcomes).

We compare the performance of LSTM player simulation models with several baselines: logistic regression (Logistic), multi-layer perceptron (MLP), and mixture of experts (ME) (Masoudnia and Ebrahimpour 2014). For each of the NN models (MLP, ME, LSTM), they were implemented with both a bipartite architecture (denoted "-Bi") and a multi-task architecture (denoted "-Mul"). The bipartite networks used two separate models for player action prediction and outcome prediction (Wang et al. 2017), in contrast to the multi-task architectures that used a single model with two parallel output layers. For each MLP network, we designed a 2 hidden-layer structure, with the first hidden layer consisting of 32 neurons and the second hidden layer consisting of 24 neurons. The ME models consisted of four MLPs with a four-gate structure. The design of each MLP in the ME models is the same as the standalone MLP models. The LSTM models contain one hidden layer with 64 hidden units. The Adam (Kingma and Ba 2015) optimizer was utilized to train all NN models, and a dropout (Srivastava et al. 2014) rate of 0.3 was applied to avoid overfitting. Five-fold cross validations are conducted to evaluate player simulation models' prediction capabilities.

Results from a comparison of these models are shown in Tables 1 and 2. For player action prediction, the Friedman statistical test finds significant differences in player action prediction accuracy rates, $\chi^2(6)=24.3$, $p < 0.001$, and prediction macro-average F1 scores, $\chi^2(6)=26.6$, $p < 0.001$, across the seven models. The best performing player action prediction model, LSTM-Bi, outperforms the other six models on the evaluations in each split of the five-fold cross validation with respect to both accuracy rate and macro-average F1 score metrics.

	Logistic	MLP-Bi	MLP-Mul	ME-Bi	ME-Mul	LSTM-Bi	LSTM-Mul
Prediction Accuracy	0.3135	0.3248	0.2859	0.3190	0.3190	**0.3304**	0.3160
Macro-Average F1 Score	0.1774	0.1698	0.1420	0.1588	0.1561	**0.2361**	0.1738

Table 1. Player action prediction performance of player simulation models.

	Logistic	MLP-Bi	MLP-Mul	ME-Bi	ME-Mul	LSTM-Bi	LSTM-Mul
Prediction Accuracy	0.5673	0.4952	0.5622	0.5723	0.5523	0.5747	**0.5871**
Macro-Average F1 Score	0.5712	0.5125	0.5459	0.5712	0.5529	0.5723	**0.5909**

Table 2. Player outcome prediction performance of player simulation models.

We further run a Wilcoxon post-hoc analysis. In a series of pairwise comparisons using the post-hoc statistical test, we derive the *p* value of 0.043 between LSTM-Bi and all other competitive models. An interesting finding from Table 1 is that LSTM-Bi outperforms the competitive baselines for the action prediction task with a sizable difference in the macro-average F1 score (33.1% improvement over the 2nd best). This result indicates LSTM-Bi predicts rarely occurred player actions much better than other models.

For player outcome prediction, the LSTM-Mul model achieves the highest averaged accuracy rate and macro-average F1 score. The Friedman tests indicate that there are no statistically significant differences across these models on both outcome prediction accuracy rate ($\chi^2(6)=12.5$, $p=0.053$) and macro-average F1 score ($\chi^2(6)=7.7$, $p=0.259$).

With regard to the training process, we find that the multi-task NN architecture usually speeds up the training time for LSTM and MLP-based player simulation models. On average, an LSTM-Mul model completes the training after 76 epochs, which is only 34.2% of the number of training epochs required for an LSTM-Bi model, where the stopping criterion is when they reach the highest predictive accuracy on a held-out validation set. This demonstrates that multi-task NN models can be a valuable option especially for player simulation problems with large dataset when training speed is of significant concern.

We also investigate the effectiveness of utilizing synthetic data from LSTM-Bi player simulation model to train an RL-based planner for personalizing interactive narrative in CRYSTAL ISLAND. In our experiment, we randomly select 80% of the data (321 students' records) to form a training set, and the remaining to form a test set. RL-based interactive narrative planners are trained using either (1) human player interaction data in the training set, or (2) synthetic data from the training set-based simulated player. We compare these two interactive narrative planners in terms of their estimated RL policy values, which represent the expected accumulated rewards the planners will obtain by following optimal policies during interactive narrative personalization.

To evaluate the two RL-based interactive narrative planners, we utilize an evaluation approach that leverages test set-based simulated players—they are generated from test set data following the same procedure as used to devise training set-based simulated players—to interact with the RL-based planner. According to the reward design in CRYSTAL ISLAND, valid RL policy values are in the range [-1,1], in which higher is better. Linear RL and Q-network RL techniques are employed to train the interactive narrative planners. The Q-network model has a 2-hidden layer structure with 64 and 32 hidden neurons, respectively. Other hyperparameters are set according to the experiment results reported in (Wang et al. 2017). As seen in Table 3, both linear RL models and Q-network RL models achieve better performance by utilizing the test set-based simulated players in the training process. The marginal improvement in normalized policy value by utilizing simulated players to train a linear RL interactive narrative planner is 5.6%. This marginal improvement is 4.1% when Q-network RL is utilized. For comparison, the uniform random policy's value is -0.0212.

Policy Value	Linear RL	Q-network RL
Train with Raw Data	0.0408	0.1234
Train with Simulated Players	0.0992	**0.1698**

Table 3. Interactive narrative planning policy values.

Conclusion

We have presented an LSTM-based neural network framework for devising player simulation models to assist in training data-driven interactive narrative planners. We utilize LSTMs to encode sequential information about player behavior to perform player action prediction and player outcome prediction. We propose the utilization of a multi-task neural network architecture to predict player action and outcome from a single model. Empirical results demonstrate that LSTM bipartite networks yield improved performance for player action prediction over several competitive baselines. The multi-task NN architecture trains comparable player outcome prediction models much faster than bipartite models using LSTM. Further, the resulting player simulations enhance the quality of interactive narrative planning policies induced under both linear RL and Q-network RL methods. In future work, it will be important to investigate alternate deep learning techniques, such as generative adversarial networks, for devising effective player simulation models, as well as investigate the runtime impacts of RL-based interactive narrative planners trained with simulated player data.

References

Fourure, D., Emonet, R., Fromont, E., Muselet, D., Neverova, N., Trémeau, A., and Wolf, C. 2017. Multi-Task, Multi-Domain Learning: Application to Semantic Segmentation and Pose Regression. *Neurocomputing*, 251: 68-80.

Graves, A. 2012. *Supervised sequence labelling with recurrent neural networks*. Vol. 385. Springer.

Harrison, B., and Riedl, M. O. 2016. Learning from Stories: Using Crowdsourced Narratives to Train Virtual Agents. In *Proceeding of the Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 183-189.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*.

Lample, G., and Chaplot, D. S. 2016. Playing FPS games with deep reinforcement learning. *arXiv*:1609.05521.

Lee, S., Rowe, J., Mott, B., and Lester. J. 2014. A Supervised Learning Framework for Modeling Director Agent Strategies in Educational Interactive Narrative. *IEEE Transactions on Computational Intelligence and AI in Games*, 6: 203–215.

Li, B., Lee-Urban, S., Johnston, G., and Riedl, M. 2013. Story Generation with Crowdsourced Plot Graphs. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, 598–604.

Marx, J. D., and Cummings, K. 2007. Normalized Change. *American Journal of Physics*, 75(1): 87-91.

Masoudnia, S., and Ebrahimpour, R. 2014. Mixture of Experts: A Literature Survey. *Artificial Intelligence Review*, 42(2): 275-293.

Nam, H., and Han, B. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4293-4302.

Nelson, M., Roberts, D., Isbell, C., and Mateas, M. 2006. Reinforcement Learning for Declarative Optimization-Based Drama Management. In *Proceedings of the 5th International Conference on Autonomous Agent and Multi-Agent Systems*, 775–782.

Roberts, D., Nelson, M., Isbell, C., Mateas, M., and Littman, M. 2006. Targeting Specific Distributions of Trajectories in MDPs. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, 1213-1218.

Rowe, J., Mott, B., and Lester, J. 2014. Optimizing Player Experience in Interactive Narrative Planning: A Modular Reinforcement Learning Approach. In *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 160–166.

Rowe, J., Shores, L., Mott, B., and Lester, J. 2011. Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *International Journal of Artificial Intelligence in Education*, 21(1-2), 115-133.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1), 1929-1958.

Thue, D., Bulitko, V., Spetch, M., and Wasylishen, E. 2007. Interactive Storytelling: A Player Modelling Approach. In *Proceedings of the 3rd AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 43-48.

Wang, P., Rowe, J., Min, W., Mott, B., and Lester, J. 2017. Interactive Narrative Personalization with Deep Reinforcement Learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*.

Wang, P., Rowe, J., Mott, B., and Lester, J. 2016. Decomposing Drama Management in Educational Interactive Narrative: A Modular Reinforcement Learning Approach. In *Proceedings of the Ninth International Conference on Interactive Digital Storytelling*, 270-282.

Yu, H., and Riedl, M. 2014. Personalized Interactive Narratives via Sequential Recommendation of Plot Points. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2): 174–187.