High-Fidelity Simulated Players for Interactive Narrative Planning

Pengcheng Wang, Jonathan Rowe, Wookhee Min, Bradford Mott, James Lester
Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
{pwang8, jprowe, wmin, bwmott, lester}@ncsu.edu

Abstract

Interactive narrative planning offers significant potential for creating adaptive gameplay experiences. While data-driven techniques have been devised that utilize player interaction data to induce policies for interactive narrative planners, they require enormously large gameplay datasets. A promising approach to addressing this challenge is creating simulated players whose behaviors closely approximate those of human players. In this paper, we propose a novel approach to generating high-fidelity simulated players based on deep recurrent highway networks and deep convolutional networks. Empirical results demonstrate that the proposed models significantly outperform the prior state-of-the-art in generating high-fidelity simulated player models that accurately imitate human players' narrative interactions. Using the high-fidelity simulated player models, we show the advantage of more exploratory reinforcement learning methods for deriving generalizable narrative adaptation policies.

1 Introduction

Data-driven approaches to interactive narrative generation have been a topic of increasing interest. A broad range of machine learning methods have demonstrated promise in inducing interactive narrative planners to create tailored narrative scenarios and adaptive gameplay experiences. Dynamic Bayesian networks have been used to model director agents' decisions in interactive narrative generation [Lee *et al*., 2014], and prefix-based collaborative filtering has been utilized to predict players' preferences for particular story branches [Yu and Riedl, 2014]. Crowdsourcing has been employed to construct plot graphs for deriving new stories [Li *et al*., 2013], and reinforcement learning (RL) has been used to tailor interactive narratives to individual players in game-based learning environments [Wang *et al*., 2017a].

Despite significant promise, data-driven interactive narrative generation approaches often depend on high volumes of player interaction data, largely because of the significant uncertainty inherent in the complex interactions between narrative planners and human players. Because collecting high-quality player data is resource intensive, creating high-fidelity simulated player models that accurately imitate human player narrative interactions offers significant potential for data-driven interactive narrative planning.

In this paper, we address the problem of generating high-fidelity predictive simulated player models with deep neural networks (NNs). We introduce novel approaches to simulated player modeling using deep recurrent highway networks (RHNs) [Zilly *et al*., 2017] and deep convolutional neural networks (CNNs) [LeCun *et al*., 1989] for the tasks of player action prediction and player outcome prediction, respectively. Empirical results with an educational interactive narrative, CRYSTAL ISLAND, demonstrate that these deep NN-based simulated player models significantly outperform prior state-of-the-art shallow long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] network-based models on both player action prediction and player outcome prediction. Further, using high-fidelity simulated player models based on deep RHNs and CNNs, we investigate the effects of an RL-based narrative planner's exploration strategy on the generalizability of the derived narrative planning policies. Results indicate that more exploratory RL methods derive more generalizable narrative planning policies.

2 Related Work

Several families of computational approaches have been developed for interactive narrative generation, including STRIPS-planning [Porteous *et al*., 2015; Robertson and Young, 2015], adversarial search [Lamstein and Mateas, 2004] and machine learning [Nelson *et al*., 2006; Wang *et al*., 2016]. For data-driven approaches to interactive narrative generation, a variety of methods and data sources have been studied. Nelson *et al*. [2006] adopted temporal-difference learning to train a drama manager for a text-based interactive fiction system using assumption-based synthetic data. Roberts *et al*. [2006] proposed target-trajectory distribution Markov decision process to derive narratives following author-specified narrative distributions. With crowd-sourced data, Harrison and Riedl [2016] used RL to control virtual agents in an interactive narrative. Rowe *et al*. [2014] and Wang *et al*. [2016] applied modular RL to personalize interactive narratives using a corpus of human players'

gameplay data. Robin's laws have been utilized in player modeling to derive an interactive narrative framework that selects story elements on the basis of players' gameplay style [Thue *et al*., 2007]. Yu and Riedl [2014] also adopted Robin's laws-based simulated players in their collaborative filtering-based interactive narrative generator's verification. Zook *et al*. [2015] used automated planning agents to simulate humans to generate game playthroughs. Wang *et al*. [2017a] introduced a statistical predictive simulated player model to train deep Q-networks for tailoring interactive narratives to individual players. The advantage of utilizing simulated players for training interactive narrative planners by leveraging the intrinsic rules of a narrative environment has also been investigated [Wang *et al*., 2017b].

Although NNs have been adopted in predictive simulated player modeling, prior state-of-the-art methods use shallow (1 hidden layer) network structures. Meanwhile, theoretical and empirical evidence indicates that deep NNs are more capable than shallow networks in learning meaningful representations [Bengio, 2009; Srivastava *et al*., 2015]. Deep recurrent and deep convolutional networks' success in tasks such as language modeling [Zilly *et al*., 2017] and image recognition [He *et al*., 2016] motivate the present work on deep neural network-based high-fidelity predictive simulated player modeling.

3 Data-Driven Interactive Narrative Planning

3.1 Interactive Narrative Testbed

We investigate deep neural network-based simulated player modeling with an interactive narrative-centered educational game, CRYSTAL ISLAND, which features a science mystery involving an infectious outbreak on a remote island. In the game, players explore the virtual environment by talking with non-player characters (NPCs), reading virtual books, conducting virtual laboratory tests, and completing an in-game diagnosis worksheet in the process of solving the mystery.

In CRYSTAL ISLAND, a narrative planner monitors players' gameplay and dynamically tailors the interactive experience when narrative adaptation opportunities arise. In this work, we examine four recurring adaptable events in CRYSTAL ISLAND, each of which is triggered by certain player actions. The four adaptable events include the following: (1) how an NPC, Teresa, describes her symptoms during an in-game dialogue with the player; (2) how an 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 NPCs deliver in-game quizzes for the player to take. An adaptable narrative planner should be able to tailor the narrative to each individual player by selecting proper narrative adaptation actions according to how the narrative unfolds so that the expected gameplay experience can be optimized. For example, when the player converses with Teresa, a sick NPC, the planner may direct Teresa to provide minimal symptoms-related detail if the player has not visited key areas of the game world and the narrative planning policy encourages game environment exploration before revealing symptoms.

In this work, the success of an interactive narrative planner is assessed in terms of player outcomes. Because CRYSTAL ISLAND has an educational focus, we adopt normalized learning gain (NLG) [Marx and Cummings, 2007] as a player outcome measure. NLG, a broadly used educational metric, captures a student's observed learning gain divided by their potential learning gain (i.e., (post - pre) / (max – pre)). Player outcomes are grouped into two categories, high NLG and low NLG, which are determined using players' NLG scores relative to the median NLG value.

The interactive narrative dataset was collected from two human subject studies with 453 players. In the studies, an interactive narrative planner utilizing a uniform random policy was deployed to broadly sample the planning policy space. All of the adaptable events within the narrative were designed in such a manner that story coherence is guaranteed under any narrative adaptation actions. In these two studies, players' gameplay action sequences, players' traits, players' interaction history with the narrative planner, and their pre- and post-test outcomes were collected. After eliminating incomplete records, data from 402 students were retained.

3.2 Simulated Player Modeling

Modeling simulated players entails developing two components of human player characteristics: players' actions in the interactive narrative (e.g., player interactions with the CRYSTAL ISLAND interactive narrative) and players' narrative outcomes (e.g., players' learning gains from interacting with CRYSTAL ISLAND). We model these two components with two modules. In CRYSTAL ISLAND, the player action prediction module is a classifier that predicts the player's next action at each gameplay time step among 15 possible player actions (including game-ending actions), which collectively captures the ways players explore CRYSTAL ISLAND's interactive narrative. The player outcome prediction module is a binary classifier, which distinguishes player experiences that yield high learning outcomes (high NLG) from those with low learning outcomes (low NLG). A set of 21 features is utilized to represent players' gameplay state for both action and outcome predictors. These features store cumulative player action history (14 features), player traits (3 features), and the narrative planner's past decisions (4 features).

High-fidelity simulated player models can be utilized to induce and evaluate interactive narrative planners. In these settings, the simulated player generates one player action at each gameplay time step by sampling from the softmax distribution in the player action prediction module's output. When the game-ending action is generated, a player outcome is sampled similarly from the player outcome prediction module. The simulated player thereby simulates the process of a human player interacting with the narrative.

3.3 Reinforcement Learning-Based Interactive Narrative Planning

For interactive narrative planning problems, RL provides a natural computational framework because of its capacity to

model sequential decision-making tasks in uncertain environments with delayed rewards, i.e., player outcomes. We represent interactions between a narrative planner and a player as a series of stochastic state transitions, which are influenced by the narrative planner's run-time adaptations that shape players' narrative outcomes.

Formally, when a player triggers an adaptable event e at interactive narrative planning time step t after conducting a series of player actions, the narrative planner chooses a narrative planning action a^{e_k} from the action set $A^e = \{a^{e_1}, a^{e_2}, \dots, a^{e_m}\}$ of event e. The narrative planner makes its decision following planning policy π_θ at the narrative interaction state $s_t \in S = (o_{t-n+1}, \dots, o_t)$, in which o_t is the observation at narrative planning time step t, and n is the number of observations encoded in the RL agent's state representation. Then the interactive narrative proceeds to the state s_{t+1} and a reward r_t is administered according to a narrative experience quality metric. Training an RL-based interactive narrative planner optimizes the interactive narrative planning policy π_θ so that the expected discounted cumulative reward $R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau$ can be maximized, where the discount factor $\gamma \in (0,1]$.

In CRYSTAL ISLAND, the RL narrative planner utilizes 25 features for each observation, of which 21 features are the ones used to represent a simulated player's gameplay state (described in Section 3.2), and 4 additional features form a one-hot encoding for the adaptable event index. Among the four adaptable events, in total 10 optional narrative adaptation actions exist. A discount factor $\gamma = 0.99$ is set to slightly encourage quick learning for players.

4 Simulated Player Modeling with Deep Neural Networks

Prior successes with shallow (1 hidden layer) LSTM networks in predictive simulated player modeling demonstrate the advantages of NN-based models in dealing with narrative interaction data [Wang *et al.*, 2017b]. Following this path, we present deep NN-based models in training player action and outcome predictors, with novel NN structures and effective regularization techniques.

4.1 Recurrent Highway Networks for Player Action Prediction

Recurrent highway networks (RHNs) construct recurrent NNs using highway layers [Zilly *et al.*, 2017]. A highway layer calculates its output as a weighted summation of an input's non-linear transformation and the input itself. This design has proven effective for building deep and trainable NNs [Srivastava *et al.*, 2015]. Specifically, RHNs increase network depth by utilizing a high recurrence depth in the recurrent step transition: at each transition time step, multiple highway layers (yellow blocks in Figure 1a) are connected. A 3-layer RHN with recurrence depth of 3 is depicted in Figure 1a. The output of the adopted RHN model at recurrence depth l and time step t on any RHN layer (the output of any yellow block in Figure 1a) is calculated as follows:

$$\mathrm{s}_l^t = \mathrm{h}_l^t \cdot \mathrm{r}_l^t + \mathrm{s}_{l-1}^t \cdot (1 - \mathrm{r}_l^t) \quad (1)$$

in which h_l^t is a nonlinear transformation of s_{l-1}^t with s_0^t being the recurrent output from time step t-1, and r_l^t is the transform gate that controls the contribution of h_l^t into the output s_l^t. $1 - \mathrm{r}_l^t$ calculates the weight for the prior recurrent state s_{l-1}^t into s_l^t. Calculation of h_l^t and r_l^t is as follows:

$$\mathrm{h}_l^t = \tanh\left(W_h x^t \mathbb{I}_{l=1} + R_{h^l} \mathrm{s}_{l-1}^t + b_{h^l}\right) \quad (2)$$

$$\mathrm{r}_l^t = \mathrm{sigmoid}\left(W_r x^t \mathbb{I}_{l=1} + R_{r^l} \mathrm{s}_{l-1}^t + b_{r^l}\right) \quad (3)$$

where x^t is input into one RHN layer at time step t, $\mathbb{I}$ is the identity function, and W, R, b are trainable weights and biases in RHN.

To achieve strong generalization, we adopt variational inference based dropout [Gal and Ghahramani, 2016] to train RHN-based player action predictors. This dropout technique utilizes a fixed dropout mask on each recurrent unit for each unique sequence. It resamples the dropout mask when processing the next sequence. So the same set of features will be kept for each sequence during training. This technique has shown better results than classic dropout techniques in recurrent NN regularization.

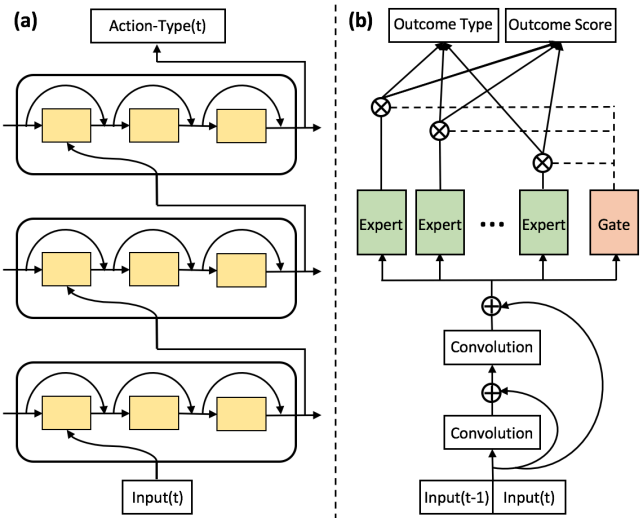

Figure 1: The NN architecture of (a) RHN player action predictor and (b) CNN player outcome predictor.

4.2 Convolutional Neural Networks for Player Outcome Prediction

Because player outcomes are typically measured only after the conclusion of an interactive narrative experience, outcome data tends to be more sparse than player action data. To address this problem we utilize convolutional NN-based (CNN) models for player outcome prediction. CNN-based player outcome prediction models can detect repeatable patterns from each gameplay time step and eliminate the constraint imposed by recurrent NN models of having to extract patterns using the exact input sequences.

CNN models can be leveraged in simulated player modeling because the same feature set is usually utilized at each gameplay time step to record the player's behaviors, thereby enabling convolution to be naturally applied along the time axis. This strategy has been used in machine

translation tasks [Gehring *et al*., 2017]. The CNN-based player outcome prediction model has two convolution layers stacked above the input layer (Figure 1b). To make the network easy to train, we adopt the residual learning method in [He *et al*., 2016]. Residual structure in NNs builds a direct path from input to output so that each layer with a residual connection learns a residual function with reference to the layer's input, which has proven effective in training deep CNN models. Specifically, we adopt a residual structure that directly adds raw input into the output of each of the two convolution layers. In this way, the output of convolution layer l is $o_l = \text{conv}(i_l) + i_1$ in which i_l represents the input into convolution layer *l*.

To improve the generalization of the player outcome predictor, two major regularization techniques are utilized. The first is multi-task learning. For NN models, when multiple tasks are correlated, building one NN model with multiple outputs can potentially improve its performance on each individual task. In this work, we construct a two-task NN model to predict the player outcome classification result (i.e., the outcome type in Figure 1b, high NLG or low NLG in CRYSTAL ISLAND) and regression result (i.e., the outcome score in Figure 1b, the exact NLG score in CRYSTAL ISLAND) together. Because these two tasks are highly correlated, adding a regression task can improve the performance of the outcome classification predictor. The second regularization technique utilized in the player outcome predictor is mixture of experts (ME) [Masoudnia and Ebrahimpour, 2014], which is an ensemble method employing multiple NNs to solve the same problem. Specifically, in our player outcome predictor, an ME module is utilized as a player outcome classifier and regressor, so the final output is the weighted summation of all expert modules (green blocks in Figure 1b) with a gate module (orange block in Figure 1b) assigning weights to each expert.

5 Interactive Narrative Planning with High-Fidelity Simulated Players

High-fidelity simulated player models enable the investigation of the generalizability of narrative planning policies, which measures the policy's effectiveness when interacting with unfamiliar player populations. A narrative planning policy's generalizability is essential to consider because the deployment environment of an interactive narrative planner is often challenging to control, and it can differ substantially from the training environment. For interactions with player populations who have different gameplay preferences, a generalizable narrative planning policy should generate meaningful and adaptive narratives. In this work we investigate the effects of an RL-based narrative planner's exploration-exploitation strategy on the narrative planning policy's generalizability. The exploration-exploitation strategy balances an RL agent's preference for gathering more information about the environment or utilizing its current knowledge to drive decision making. This problem is central in interactive narrative adaptation because of the pervasive high uncertainty in the interplay between human interaction and narrative adaptation. With insufficient exploration, the high variance of the state transition distribution and reward distribution may lead a narrative planner to learn a risky suboptimal policy. Without sufficient exploitation, convergence to (locally) optimal policies may not be achieved. This problem becomes even more important when narrative planners are expected to interact with unfamiliar player populations that exhibit different gameplay patterns. To explore this problem, we investigate three deep RL methods with different exploration-exploitation strategies: asynchronous one-step Q-networks (AQN), the asynchronous advantage actor-critic (A3C) method [Mnih *et al*., 2016], and the proximal policy optimization (PPO) method [Schulman *et al*., 2017].

AQN implements one-step Q-learning using deep NNs. As a value-based RL method, AQN learns the $Q(s_t, a_t; \theta)$ function, which represents expected discounted cumulative rewards by performing action a_t at state s_t following policy π_θ, towards an estimated target as shown in Equation 4:

$$y = r + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-) \quad (4)$$

so that $Q(s_t, a_t; \theta)$ can gradually approach the Q-values $Q(s_t, a_t; \theta^*)$ of an (locally) optimal policy π_{θ^*}. θ and θ^- are the parameters of the AQN model and its target model, respectively. The target model parameters θ^- are only updated with the training AQN parameters θ every certain number of steps. The other symbols in Equation 4 have the same meaning as in Section 3.3. As a value-based method, AQN usually adopts the exploratory strategy of ε-greedy. In ε-greedy, the RL agent picks the action $argmax_a Q(s, a; \theta)$ with probability $1 - \varepsilon$ for exploiting the learned knowledge, or it selects a random action with probability ε for exploring alternative choices. Usually, a decaying schedule is set for the ε parameter, so the agent explores the environment more often during its early learning stages, and it exploits learned knowledge more often later in learning as it has gained experiences.

A3C and PPO belong to the category of actor-critic RL methods. In contrast to value-based methods, which only maintain Q-values for state-action pairs, actor-critic RL methods maintain a state value function $V(s; \theta)$ measuring the expected discounted cumulative rewards from state *s* following policy π_θ, as well as an explicit policy representation $\pi(a|s; \theta)$ expressing the probability distribution of selecting each action at state *s* following policy π_θ when optional actions are discrete. In A3C, training focuses on increasing the value of a target function as defined in Equation 5, in which A_t^π is the advantage value function which can be estimated from samples using Equation 6. In contrast to A3C, PPO implements trust region update [Schulman *et al*., 2015] by utilizing a special target function as defined in Equation 7, so that policy updates are constrained within a threshold controlled by a hyper-parameter ϵ to improve training robustness. In Equation 7, $rt_t(\theta)$ represents the ratio of probability to obtain the transition sequence under the current policy π_θ and the probability under the sampling policy.

$$T_t = \log \pi(a_t|s_t; \theta)\mathrm{A}_t^\pi \quad (5)$$

$$\widehat{\mathrm{A}}_t^{\pi} = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau - \mathrm{V}(\mathrm{s}_t;\theta) \quad (6)$$

$$T_t = \mathbb{E}[\min(rt_t(\theta)\mathrm{A}_t^{\pi}, clip(rt_t(\theta),1-\epsilon,1+\epsilon)\mathrm{A}_t^{\pi})\,] \quad (7)$$

Because actor-critic methods keep an explicit representation of the policy $\pi(a|\mathrm{s};\theta)$, in standard use their exploration strategy is to sample actions following the probability distribution in π so that more promising actions will be selected more often for exploitation, while other actions also have chances to be adopted for exploration. The difference in the target functions of A3C and PPO not only distinguishes their updating processes, but it also affects their exploration strategies. Because PPO prevents its updating step from being very large, policies of PPO agents change more slowly, which offers PPO agents opportunities to exploit the already learned knowledge more often, and allows PPO polices to converge easier. On the other hand, because A3C's policy updating can be significantly influenced by each single noisy transition sequence in any training stage, A3C is encouraged to avoid early convergence to (local) optima and explore other choices more often, which holds promise for interactive narrative planning with unfamiliar players.

6 Evaluation

6.1 Simulated Player Modeling Evaluation

We evaluate the performance of simulated player models on the metrics of player action and outcome prediction accuracy rates, and macro-average F1 scores . In the CRYSTAL ISLAND dataset, 402 human players generated a total of 16,313 player actions spanning 15 player action types, of which the most commonly exhibited action type consists of 24.01% of all player actions. With a median split on the player learning outcome metric of NLG, 200 out of 402 players are labeled as high NLG players, and all the other players are labeled as low NLG players. For the following player action and outcome prediction experiments, Adam [Kingma and Ba, 2015] is employed for NN optimization. A dropout rate of 0.1 is adopted for all the models. Five-fold cross-validation is conducted for both prediction accuracy and macro-average F1 score based evaluations. The number of training epochs in each round of cross-validation is determined using a separate validation set, which is later merged back into the training set for the final evaluation. All the model size related hyperparameters are tuned using random search.

In our evaluation of player action predictors (Table 1), a logistic regression model serves as a baseline, a shallow (1-layer) LSTM model with 64 hidden neurons is the prior state-of-the-art [Wang *et al*., 2017b], and three deep recurrent NN models are considered. The RHN model has 3 RHN layers with each layer maintaining a recurrence depth of 3 (as in Figure 1a). A 4-layer deep LSTM network and a 4-layer Grid LSTM [Kalchbrenner *et al*., 2016] model are compared to consider the effectiveness of different recurrent unit structures. The input layer and all the recurrent layers in the three deep recurrent NNs have 21 neurons that reflect the gameplay feature set size. A CNN action predictor with one convolution layer of size 21 using a 1×21 sized convolution kernel, convolution step size of 1, and 2 time-frame inputs is also utilized.

Using results from five-fold cross-validation, the Friedman statistical test finds significant differences in player action prediction accuracy rates, $\chi^2(5)=25.0$, $p<0.001$, and macro-average F1 scores, $\chi^2(5)=23.6$, $p<0.001$, across the six models. The deep RHN model outperforms all the other action prediction models, including the prior state-of-the-art (shallow LSTM) by large margins, with the Wilcoxon post-hoc analysis yielding a p value of 0.043 for each pairwise comparison on both prediction accuracy and macro-average F1 score metrics. The fact that all three deep recurrent NN models (RHN, deep LSTM, and Grid LSTM) outperform the shallow LSTM model on both metrics with p values of 0.043 under pairwise Wilcoxon tests suggests that depth in recurrent NNs is an important factor in player action predictors. Further, all three deep recurrent NNs outperform the CNN model with p values of 0.043 from pairwise Wilcoxon tests, which indicates that deep recurrent NNs provide an advantage over CNNs in player action prediction. A possible explanation may be that the exact gameplay history sequence provides considerably more information, which contributes to accurate player action prediction. To investigate the effects of regularization technique on training the deep RHN model, we train other RHNs either without dropout or only adopting a normal dropout layer on the output of the RHN model. In these cases, the action prediction accuracy rates drop from 0.4435 to 0.4112 and 0.4137, respectively.

	Logistic Regress.	Shallow LSTM	Deep LSTM	Grid LSTM	RHN	CNN
Accu.	0.3135	0.3304	0.4306	0.4126	**0.4435**	0.3605
Mac. F1	0.1774	0.2361	0.3108	0.3065	**0.3218**	0.2655

Table 1: Player action prediction model performance.

	Logistic Regress.	LSTM-Mul	RHN	CNN	CNN-noME	CNN-noMul
Accu.	0.5673	0.5871	0.5997	**0.6096**	0.5772	0.5971
Mac. F1	0.5712	0.5909	0.5997	**0.6121**	0.5798	0.5974

Table 2: Player outcome prediction model performance.

In a similar evaluation process, we assess the following player outcome prediction models. Logistic regression serves as a baseline. LSTM-Mul, a multi-task structured shallow (1-layer) LSTM model, is the prior state-of-the-art [Wang *et al*., 2017b]. A 1-layer RHN with recurrence depth of 5 is a deep recurrent model. A 2-layer CNN model with mixture of experts (using 23 experts) and multi-task regularization (labeled as CNN in Table 2) is implemented as shown in Figure 1b. CNN-noME is the same network as the CNN model but without mixture of experts regularization, and CNN-noMul eliminates the multi-task output module from the CNN model. All CNN-based outcome predictors have convolution layers of size 21 with convolution kernels of size 1×21, convolution step size of 1, and 2 time-frame inputs.

CNN models outperform all the other models on both metrics of player outcome prediction accuracy and macro-average F1 score with five-fold cross-validation (Table 2). The Friedman test indicates that there are no statistically

significant differences between these models in the outcome prediction accuracy ($\chi^2(5)=9.0$, p=0.110) and macro-average F1 score ($\chi^2(5)=7.8$, p=0.167). However, we find that CNN models outperform the logistic regression baseline in all the five rounds of evaluations in cross-validation on both metrics, which is not achieved by the LSTM-Mul or RHN models. This observation indicates that in contrast to player action prediction, keeping track of the exact sequence of gameplay history may not be necessary in outcome prediction, which allows CNN models to perform better than recurrent neural networks. The result that CNN models achieve higher average performance over CNN-noME and CNN-noMul reveals that both mixture of experts and multi-task learning techniques have good regularization effects on training CNN-based player outcome predictors.

6.2 Interactive Narrative Planning Evaluation

Building on the results described in Section 6.1, we train high-fidelity deep RHN-based and deep CNN-based simulated player models to evaluate the effectiveness and generalizability of narrative adaptation policies derived by RL methods. Three experimental conditions are employed. In Condition 1, RL narrative planners are trained using the full CRYSTAL ISLAND corpus-based simulated player model and then evaluated on the same simulated player model. In Condition 2, the CRYSTAL ISLAND corpus is randomly divided into two halves, and a simulated player model is trained on each half. Then two RL narrative planners are trained using each half corpus-based simulated player model and evaluated on the full corpus-based simulated player model. In Condition 3, a two-fold cross-validation-like evaluation is conducted, in which RL narrative planners are trained using a training set-based simulated player model and evaluated on a test set-based simulated player model. In this way, the evaluation narrative planning environment in Conditions 1, 2 and 3 becomes increasingly unfamiliar to the trained narrative planners. For each interactive narrative interaction step, a reward of 1 is assigned only at the conclusion of the narrative if the simulated player reaches a high NLG outcome; otherwise, the reward is always 0. All evaluations are conducted by allowing the derived narrative planners to interact with high-fidelity simulated player models for 5,000 episodes. Policies are evaluated by policy value, which is the average score the narrative planner receives for each narrative adaptation episode, ranging in [0,1]. For each RL method in each condition, three policies are generated with different training steps after training converges. In evaluation, averaged policy values are reported with the largest coefficient of variation of 5.5% for all entries in Table 3. For A3C and PPO methods, a stochastic policy using the direct output of the methods and a greedy policy, which always adopts the action with the largest probability from $\pi(a|s;\theta^*)$, are both evaluated. The ε value in ε-greedy of AQN is set to decay from 1 to 0.01 linearly in 75% of the training steps, which makes AQN operate in an exploration mode initially and in an exploitation mode in the later stages of training. Because of PPO's optimization, it also takes on an exploitation mode more prominently than A3C. A3C is the most exploratory RL method in this experiment.

We find that the exploratory strategy of RL methods heavily affects the effectiveness, and particularly the generalizability, of interactive narrative planners (Table 3). When the evaluation environment is the same as the training environment (Condition 1), all RL methods derive effective narrative planning polices, and the more exploitative methods (AQN and PPO) are usually superior to the more exploratory method (A3C). These behaviors are what one would expect because exploitation helps the narrative planner to converge to (local) optimal policies easier. When the evaluation environment becomes more challenging and more unfamiliar to the RL narrative planners (Conditions 2 and 3), their performance drops with the degradation speed reflecting their degree of exploration. For less exploratory methods (AQN and PPO), performance drops faster, and they even derive policies worse than random in Condition 3. Meanwhile, the more exploratory method (A3C) is able to generate effective narrative planning polices (better than random policy), even when the evaluation simulated player model is quite different from the training simulated player model. These results indicate that the greater emphasis on exploration helps A3C avoid relying too much (and too early) on the highly noisy early narrative adaptation experiences, and A3C's preference for exploring more thoroughly in uncertain environments boosts its ability to derive more generalizable narrative adaptation policies for unfamiliar players.

Cond.	Random	AQN	A3C	A3C-Greedy	PPO	PPO-Greedy
1	0.5531	**0.6255**	0.6064	0.6031	0.6061	0.6195
2	0.5531	**0.5864**	0.5769	0.5802	0.5653	0.5697
3	0.5315	0.5038	0.5510	**0.5516**	0.4756	0.4674

Table 3: Narrative adaptation policies evaluation.

7 Conclusion

High-fidelity simulated player models can play a central role in data-driven adaptable interactive narrative planning. We have presented a high-fidelity simulated player model based on deep recurrent highway networks and deep convolutional neural networks that effectively leverages the recurrent and convolutional structures to predict players' actions and outcomes. Empirical results demonstrate that the simulated player model achieves significant improvements over the prior state-of-the-art in predicting players' behaviors. With this high-fidelity simulated player model, we have also investigated the effects of reinforcement learning's exploration strategy on the effectiveness of narrative planning policy learning. Results indicate that more exploratory RL methods derive narrative planners with better generalizability. In future work, it will be important to investigate the performance of simulated player modeling on a broader range of interactive narratives, and investigate the effectiveness of data-driven interactive narrative planners induced with high-fidelity simulated player models in run-time settings with human players.

References

[Bengio, 2009] Y. Bengio. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1): 1-127, 2009.

[Gal and Ghahramani, 2016] Y. Gal, and Z. Ghahramani. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *NIPS*, pages 1019–1027, 2016.

[Gehring *et al*., 2017] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. Dauphin. Convolutional Sequence to Sequence Learning. *arXiv:1705.03122*, 2017.

[Harrison and Riedl, 2016] B. Harrison, and M. Riedl. Learning from Stories: Using Crowdsourced Narratives to Train Virtual Agents. In *AIIDE*, pages 183-189, 2016.

[He *et al*., 2016] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770-778, 2016.

[Hochreiter and Schmidhuber, 1997] S. Hochreiter, J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 1997.

[Kalchbrenner *et al*., 2016] N. Kalchbrenner, I. Danihelka, and A. Graves. Grid Long Short-term Memory. In *ICLR*, 2016.

[Kingma and Ba, 2015] D. Kingma, and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[Lamstein and Mateas, 2004] A. Lamstein, and M. Mateas. Search-Based Drama Management. In *AAAI Workshop on Challenges in Game AI*, pages 103–107, 2004.

[LeCun *et al*., 1989] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4): 541-551. 1989.

[Lee *et al*., 2014] S. Lee, J. Rowe, B. Mott, and J. Lester. A Supervised Learning Framework for Modeling Director Agent Strategies in Educational Interactive Narrative. *T-CIAIG*, 6(2): 203–215, 2014.

[Li *et al*., 2013] B. Li, S. Lee-Urban, G. Johnston, and M. Riedl. Story Generation with Crowdsourced Plot Graphs. In *AAAI*, pages 598–604, 2013.

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

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

[Mnih *et al*., 2016] V. Mnih, A. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *ICML*, pages 1928–1937, 2016.

[Nelson *et al*., 2006] M. Nelson, D. Roberts, C. Isbell, and M. Mateas. Reinforcement Learning for Declarative Optimization-Based Drama Management. In *AAMAS*, pages 775–782, 2006.

[Porteous *et al*., 2015] J. Porteous, A. Lindsay, J. Read, M. Truran, M. Cavazza. Automated Extension of Narrative Planning Domains with Antonymic Operators. In *AAMAS*, pages 1574–1555, 2015.

[Riedl and Bulitko, 2013] M. Riedl, and V. Bulitko. Interactive Narrative: An Intelligent Systems Approach. *AI Magazine*, 34(1): 67–77, 2012.

[Roberts *et al*., 2006] D. Roberts, M. Nelson, C. Isbell, M. Mateas, and M. Littman. Targeting Specific Distributions of Trajectories in MDPs. In *AAAI*, pages 1213-1218, 2006.

[Robertson and Young, 2015] J. Robertson, and R. Young. Automated Gameplay Generation from Declarative World Representations. In *AIIDE*, pages 72–78, 2015.

[Rowe *et al*., 2014] J. Rowe, B. Mott, and J. Lester. Optimizing Player Experience in Interactive Narrative Planning: A Modular Reinforcement Learning Approach. In *AIIDE*, pages 160–166, 2014.

[Schulman *et al*., 2015] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust Region Policy Optimization. In *ICML*, pages 1889-1897, 2015.

[Schulman *et al*., 2017] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *arXiv:1707.06347*, 2017.

[Srivastava *et al*., 2015] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training Very Deep Networks. In *NIPS*, pages 2377-2385, 2015.

[Thue *et al*., 2007] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen. Interactive Storytelling: A Player Modelling Approach. In *AIIDE*, pages 43-48, 2007.

[Wang *et al*., 2016] P. Wang, J. Rowe, B. Mott and J. Lester. Decomposing Drama Management in Educational Interactive Narrative: A Modular Reinforcement Learning Approach. In *ICIDS*, pages 270–282, 2016.

[Wang *et al*., 2017a] P. Wang, J. Rowe, W. Min, B. Mott and J. Lester. Interactive Narrative Personalization with Deep Reinforcement Learning. In *IJCAI*, pages 3852-3858, 2017.

[Wang *et al*., 2017b] P. Wang, J. Rowe, W. Min, B. Mott and J. Lester. Simulating Player Behavior for Data-Driven Interactive Narrative Personalization. In *AIIDE*, pages 255–261, 2017.

[Yu and Riedl, 2014] H. Yu, and M. Riedl. Personalized Interactive Narratives via Sequential Recommendation of Plot Points. *T-CIAIG*, 6(2): 174–187, 2014.

[Zilly *et al*., 2017] J. Zilly, R. Srivastava, J. Koutník, and J. Schmidhuber. Recurrent Highway Networks. In *ICML*, pages 4189-4198, 2017.

[Zook *et al*., 2015] A. Zook, B. Harrison, and M. Riedl. Monte-Carlo Tree Search for Simulation-based Strategy Analysis. In *FDG*, 2015.