Interactive Narrative Personalization with Deep Reinforcement Learning

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

Abstract

Data-driven techniques for interactive narrative generation are the subject of growing interest. Reinforcement learning (RL) offers significant potential for devising data-driven interactive narrative generators that tailor players' story experiences by inducing policies from player interaction logs. A key open question in RL-based interactive narrative generation is how to model complex player interaction patterns to learn effective policies. In this paper we present a deep RL-based interactive narrative generation framework that leverages synthetic data produced by a bipartite simulated player model. Specifically, the framework involves training a set of Q-networks to control adaptable narrative event sequences with long short-term memory network-based simulated players. We investigate the deep RL framework's performance with an educational interactive narrative, CRYSTAL ISLAND. Results suggest that the deep RL-based narrative generation framework yields effective personalized interactive narratives.

1 Introduction

Recent years have seen growing interest in data-driven approaches to interactive narrative generation. A broad range of machine learning techniques has shown promise for creating and tailoring interactive narratives, including prefix-based collaborative filtering techniques for narrative personalization [Yu and Riedl, 2014], dynamic Bayesian network models of directorial decisions [Lee et al., 2014], crowdsourcing approaches for automated story generation [Li et al., 2013], and reinforcement learning techniques for inducing interactive story planners from logs of past players' interactions [Rowe et al., 2014; Wang et al., 2016a].

Reinforcement learning (RL) techniques have shown particular promise because of their inherent support for sequential decision-making under uncertainty with delayed rewards. Computational models of interactive narrative generation should account for uncertainty because human players' actions are poorly understood and difficult to predict. Further, interactive narrative generation involves delayed rewards because player judgments about interactive narratives' quality are often deferred until after story episodes are complete. Despite this natural alignment, applications of RL to interactive narrative present significant challenges. Prior work on RL-based interactive narrative generation has often utilized low-dimensional state features and linear RL models to account for the limited availability of training data, particularly where interaction data from human players is involved [Rowe et al., 2014; Wang et al., 2016a]. Alternatively, synthetic data generated from simulations of human players have been utilized, but these models have been relatively simple and not validated [Nelson et al., 2006]. A limitation of these designs is they restrict RL-based narrative planners' abilities to identify complex non-linear player interaction patterns that are relevant to effectively personalizing interactive narratives.

In this paper, we investigate a deep RL framework for personalizing interactive narratives in stochastic open-world game environments. The contributions of our work include (1) investigating the effectiveness of a Q-network based deep RL framework for interactive narrative personalization, and (2) introducing a *bipartite player simulation model* that uses a pair of validated classifiers to generate synthetic data on player action sequences and player outcomes. By evaluating the Q-network based interactive narrative planner in a widely used educational interactive narrative, CRYSTAL ISLAND, we demonstrate that deep RL yields more effective policies for interactive narrative personalization than commonly utilized linear RL techniques.

2 Related Work

Several families of computational techniques have been investigated for interactive narrative generation, including adversarial search [Lamstein and Mateas, 2004], STRIPS-planning [Porteous et al., 2015; Robertson and Young, 2015], and RL [Riedl and Bulitko, 2013]. Nelson et al. (2006) utilized temporal-difference methods to train a drama manager for a text-based interactive fiction, Anchorhead, using synthetic data from user simulations. The simulation model assumed players interacted either cooperatively or adversarially with respect to the drama manager's decisions. This approach was extended by Roberts et al. (2006), which applied target-trajectory distribution Markov decision processes to generate interactive narratives following an

author-specified target trajectory distribution. Rowe et al. (2014) investigated a modular RL method for inducing an interactive narrative planner in an educational game. Wang et al. (2016a) studied alternative decompositional representations of an interactive narrative, demonstrating the decompositional representation's effect on the induced narrative planner's quality.

Value-function-based RL has made great strides in recent years, originating from the success of applying Q-networks to solve high-dimensional complex sequential decision-making problems. The capacity of neural networks (NNs) to extract hierarchical features enables RL agents to learn better policies in large state and policy spaces. The feasibility of designing and implementing NN structures spurred the development of Q-network techniques. More diverse Q-networks have been adopted by applying different stabilization techniques [Mnih et al., 2015; Mnih et al., 2016], encoding observation histories with LSTMs [Hausknecht and Stone, 2015], and exploiting the advantage learning technique [Wang et al., 2016b]. These findings suggest that Q-networks show significant promise for solving high-dimensional complex interactive narrative generation tasks compared to traditional linear RL methods.

Training of RL interactive narrative planners requires a large amount of interaction data, particularly for deep RL. A similar problem arises in the domain of spoken dialogue systems, where user simulations serve an important role in training effective RL-based dialogue managers [Schatzmann et al., 2006; Young et al., 2013]. In this work, we adapt this simulated user approach by inducing a bipartite player simulation model for generating synthetic training episodes for deep RL-based interactive narrative personalization.

3 Deep RL Framework For Interactive Narrative Personalization

3.1 Interactive Narrative in CRYSTAL ISLAND

To investigate the performance of a Q-network-based deep RL interactive narrative personalization framework, we utilize CRYSTAL ISLAND, an educational interactive narrative featuring a science mystery about an infectious outbreak on a remote island (Figure 1). In CRYSTAL ISLAND, the player investigates a mysterious illness that is afflicting a team of scientists on the island. To solve the mystery, the player converses with non-player characters (NPCs), reads virtual books, conducts tests in a virtual laboratory, and completes an in-game diagnosis worksheet. The interactive narrative planner monitors the player's behavior and makes decisions about how *adaptable event sequence* (AES) should unfold. An AES is a recurring series of one or more adaptable story events that can unfold into several different possible narrative trajectories, each leading to potentially different player experiences and outcomes. A simplified example of an AES is the following: the player explores the game environment for a while, then talks with a sick NPC, Teresa, which triggers an adaptable event, for which the interactive narrative planner selects the narrative planning action of allowing Teresa to reveal limited information about her symptoms. Later the player character continues exploring the virtual world, conducting tests in the virtual laboratory, and submits a diagnosis worksheet of her anticipated solution to the science mystery, which unfortunately turns out to be incorrect. This triggers another adaptable event, leading the interactive narrative planner to select the adaptable narrative planning action of providing a detailed explanation of the player's errors. These adaptable events are recurring, i.e., each event can be triggered multiple times, and adaptable event occurrences are determined by player actions and narrative rules.

Figure 1. CRYSTAL ISLAND interactive narrative environment.

In this work we investigate four learning-related adaptable events in CRYSTAL ISLAND:

- **Teresa Symptoms Event** is triggered each time the player inquires about the sick scientist Teresa's symptoms. The interactive narrative planner selects one of three conversational responses for Teresa: providing minimal detail, moderate detail, or maximal detail about her symptoms.
- **Bryce Symptoms Event** is triggered each time the player inquires about the sick NPC Bryce's symptoms. The interactive narrative planner has two optional responses: providing minimal detail or moderate detail about his symptoms.
- **Diagnosis Feedback Event** is triggered when a player submits an incorrect diagnosis worksheet. The interactive narrative planner selects one of three options for the camp nurse's feedback: providing minimally detailed, moderately detailed, or maximally detailed feedback.
- **Knowledge Quiz Event** is triggered when a player converses with a subset of the NPCs. The interactive narrative planner optionally delivers, or does not deliver, an in-game quiz about relevant microbiology concepts.

Since CRYSTAL ISLAND is designed to achieve educational objectives, we adopt normalized learning gain (NLG) as a metric for gauging player experience. NLG is a normalized measurement of the difference between the player's post-test score and pre-test score. For analyses, players' narrative records are divided into two groups: records with NLG scores above the median (high NLG), and records with NLG scores below the median (low NLG). Although NLG is adopted here, other quantifiable metrics like engagement also fit into

the deep RL framework, and they may be more appropriate for interactive narratives that are primarily designed for entertainment purposes.

The human player interaction corpus for CRYSTAL ISLAND was generated from two human subject studies (one with 300 students and one with 153 students), which were conducted in two public middle schools. Students played the game until they solved the mystery, or 55 minutes had elapsed, whichever occurred first. During the studies, a uniform random narrative planning policy was deployed to interact with human subjects. The adaptable events in CRYSTAL ISLAND were designed to ensure the coherence of the narrative when the random policy was adopted. The game logged all player actions, triggered adaptable events, and random interactive narrative planner responses. In addition, several questionnaires were administered prior to, and immediately after, students' interactions with CRYSTAL ISLAND. These questionnaires were used to obtain information about students' individual characteristics, curricular knowledge, and engagement with the narrative environment.

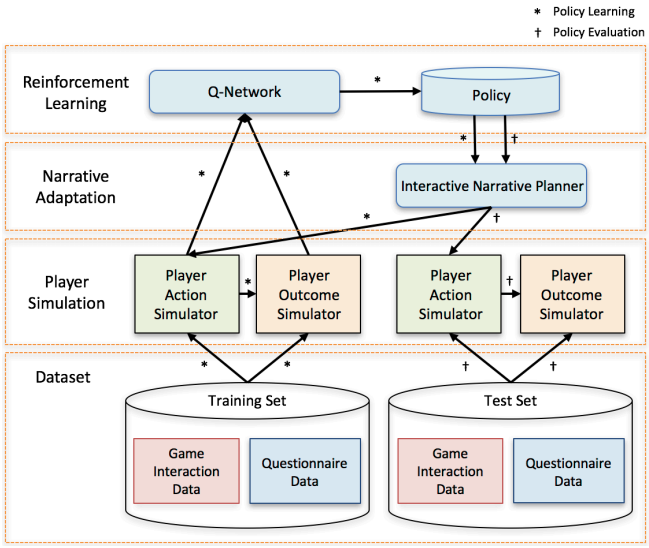

Figure 2. Q-network based deep RL interactive narrative personalization framework.

3.2 Deep RL Interactive Narrative Planner Architecture

As shown in Figure 2, the deep RL interactive narrative personalization framework has a 4-tier structure. We illustrate the architecture with respect to two principal phases: policy learning and policy evaluation. In the Dataset Tier, human player game interaction and questionnaire data are randomly divided into training (80%) and test sets (20%), which have 321 and 81 students' logs, respectively, after eliminating incomplete records. In the training set, 160 out of the 321 logs are from students with high NLG scores. In the test set, 39 out of 81 students have high NLG scores. In the bipartite model Player Simulation Tier, using features from players' gameplay behaviors and questionnaire results, one module predicts players' next action at each time step, and the other module predicts the players' learning outcomes. During training, the simulated player's behavior, as input, are sent into the RL Tier, where a Q-network improves the interactive narrative planning policy in an optimization process. In evaluation, the interactive narrative planner from the Narrative Adaptation Tier utilizes and assesses the optimized policy by interacting with simulated players from test set.

4 Player Simulation Model

In order to generate enough player interaction data for the deep RL interactive narrative planner to explore the vast state and policy space during training, we introduce a statistical, predictive bipartite model to simulate players' behavior in interaction with interactive narrative planners. In this model, players' behaviors are represented as a combination of player action and player outcome. The bipartite model uses two separate modules to predict and generate player actions and outcomes respectively.

The player action simulation module predicts the player's next action according to the interaction history. In CRYSTAL ISLAND, 15 discrete in-game player actions are modeled. In addition to the game-ending player action, which represents the termination of a game, the other 14 player actions encode how the player explores the game environment. These player actions include actions such as reading a virtual book or talking to NPCs. A subset of these player actions, under certain conditions, can trigger an adaptable event, which enables the interactive narrative planner to personalize the narrative by performing a narrative planning action that fits the individual player's interaction and experiencing patterns.

The player action prediction problem is formulated as a 15-class classification problem, in which a 21-feature input representation is constructed. The first 14 features represent player action histories (excluding game-ending action from the 15 player actions), which are accumulated counts of each player action until the current time step. The 15th to the 18th features store the interactive narrative planner's prior responses when one out of the four adaptable events was triggered. The 19th to the 21st features encode an individual player's information gathered from questionnaires, including the player's gender, prior game frequency, and pre-test score. The output of this module is the probability distribution of each player action the player might perform.

Similar to the player action simulation module, in the player outcome simulation module, the outcome prediction problem is formulated as a 2-class classification task (high NLG, low NLG). The same input feature set is adopted, and the player's outcome is estimated when the termination of each interaction trajectory is reached.

We select one type of recurrent neural network, long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] to construct the player simulation model. For both the player action prediction module and the player outcome prediction module, the same LSTM structure with one hidden layer leveraging 32 hidden units is utilized. Because in the player interaction corpus each discrete player action has been accurately labeled, the player action prediction module is a

sequence-to-sequence LSTM, in contrast to the sequence-to-one LSTM in player outcome prediction module, due to the fact that player outcome is only needed and accurately labeled at the end of each interaction sequence.

After training LSTMs for both the action and outcome prediction modules, we evaluated them on the test set. The top-1 player action prediction accuracy is 33.34%, the micro-average F1 score is 33.34%, and the top-8 player action prediction accuracy (the fraction of the correct label among the 8 labels considered most probable by the model) reaches 91.84%. The accuracy of player outcome prediction is 56.79%. The goal of the player simulation model is to imitate human player behaviors to train the RL interactive narrative planner, the result of which is reported in Section 6.

The simulated player behavior generation process proceeds as follows: First, we randomly sample the player's initial state and first time step player action from the training set. After that, at each time step, we update the player state by applying the player action effect, and then use the updated player state value as input into the action prediction LSTM. The user simulation model then selects the next player action by sampling from the optional actions according to the softmax output of the action prediction model. Whenever an adaptable event is triggered, the interactive narrative planner's response will be stored in the 15th to the 18th state features. When a game-ending player action is generated, the player outcome simulation module makes a prediction and returns it as a measure of this interactive narrative's quality. On average, the training set-based simulated player triggers 7.26 adaptable events while interacting with random policy. This number is 7.69 in observed trajectories in training set.

5 Q-network Based Interactive Narrative Planner Derivation

The AES unfolding process is modeled as a sequential decision-making problem in which the interactive narrative planner interacts with players over discrete interactive narrative planning time steps using RL. We use interactive narrative planning time step to represent the time point when adaptable events are triggered in the narrative, which makes it a more coarsely grained process than player action prediction and simulation. This interactive narrative planning time step design follows the convention in RL-based interactive narrative personalization in previous work [Rowe et al., 2014, Wang et al., 2016a]. A formal RL-based player–interactive narrative planner interaction is represented as follows: when a player triggers an adaptable event of type *c* at interactive narrative planning time step *t*, after viewing interaction state $s_t \in S = (o_{t-n+1}, \cdots, o_t)$, in which o_t is the interaction observation at interactive narrative planning time step *t*, and *n* is the length of observations encoded into the state representation, the interactive narrative planner (i.e., the RL agent) chooses an action a_t^c from a discrete action set A^c of type *c* to perform by following the interactive narrative planning policy π. The interactive narrative then proceeds to the state s_{t+1} and receives a reward signal r_t generated according to the narrative evaluation metric. Training of the RL agent optimizes the interactive narrative planning policy π, so that the expected discounted cumulative rewards gained by the interactive narrative planner as $R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau$ can be maximized, in which $\gamma \in [0,1]$ is a discount factor trading off the importance of future rewards and immediate rewards. The expected output of RL training is an optimal policy π^*, indicating the best action to be taken by the interactive narrative planner in each state.

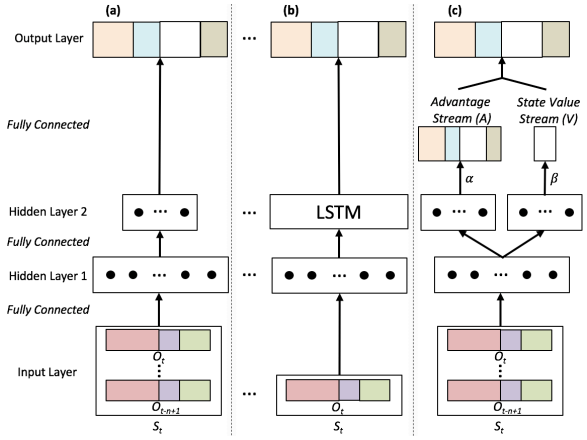

Figure 3. Q-network structures of (a) Q-net and Async Q-net, (b) Recurrent Q-net, and (c) Dueling Q-net and Async Dueling Q-net. The structure of the Recurrent Dueling Q-net is the same as the Dueling Q-net except for the substitution of the dual fully connected hidden layers in hidden layer 2 below *A* and *V* with dual LSTMs. In all these Q-networks, ReLU is the adopted non-linearity.

A Q-network is a NN implementation of the Q-learning algorithm. Under policy π, the Q value of a state–action pair (s_t, a_t) represents the expected cumulative reward the agent can receive by taking action a_t at state s_t and then following policy π till the end of the interaction. In CRYSTAL ISLAND, the interaction observation o_t is a set of 25 features, in which the first 21 features are identical to the player simulation model features containing both gameplay and questionnaire information of a simulated player when an adaptable event is triggered. The other 4 features are a one-hot encoding of adaptable event type. Interactive narrative planner action a_t^c has the definition specified in Section 3.1, and the reward signal *r* at the end of each interactive narrative is either 1 for high NLG trajectories or -1 for low NLG trajectories. The input into the CRYSTAL ISLAND Q-networks is the state s_t. The output is the Q values for 10 interactive narrative planning actions belonging to four adaptable events.

5.1 Q-network Stabilization

Utilizing NNs enables RL interactive narrative planners to extract complex nonlinear interaction patterns. However, the training process of Q-networks can be unstable, or even diverge [Tsitsiklis and Roy, 1997] because of the correlations in the sequence of neighboring interactions [Mnih et al., 2015]. To stabilize the Q-network interactive narrative planning policy training, we investigate three techniques. The first stabilization method is *experience replay* [Lin, 1993]. By applying *experience replay*, instead of updating Q functions

directly using the instant experience $e_t = (s_t, a_t, r_t, s_{t+1})$, all interaction experiences are stored into a dataset, then a minibatch of experiences are randomly sampled from the dataset at each training step. This approach breaks the correlations in neighboring interactions by rearranging sequences of experiences used in Q-learning.

The second stabilization technique we applied is *separate target network* [Mnih et al., 2015]. With this method, instead of employing a single NN to calculate both $Q(s_t, a_t)$ and the temporal difference target $r_t + max_{a'}Q(s_{t+1}, a')$ in Q-learning, a separate target network is copied from the training Q-network every certain steps and held fixed between individual updates. The target network is only used in temporal difference target estimation. By freezing the target network for certain steps, a less frequently changing target function can be achieved, so training can be stabilized. Using θ and θ^- to represent the training Q-network parameters and separate target Q-network parameters respectively, Q-learning can be executed with any gradient descent-based optimization method using the gradient of the loss function $L(\theta)$ as shown in Equation 1 and 2. Note that we remove the subscript for time step *t*, in s, a and *r*, to make the expectation not specific to to a certain time.

$$\nabla_\theta L(\theta) = \mathbb{E}_{s,a,r,s'}[(Q(s,a;\theta) - y)\nabla_\theta Q(s,a;\theta)] \quad (1)$$

$$y = r + \gamma max_{a'}Q(s', a'; \theta^-) \quad (2)$$

The third way we exploit to stabilize Q-network training is the *asynchronous gradient descent optimization* [Mnih et al., 2016]. The core idea of implementing an asynchronous RL interactive narrative planner architecture is that, instead of using one interactive narrative planner to interact with one player and learn from the interactions, multiple RL interactive narrative planners can be set up to interact with multiple players in parallel. By accumulating gradients of loss from each RL interactive narrative planner with respect to the shared Q-network parameters, the training of the Q-network can be stabilized using all current experiences without the need to build the experience replay dataset.

5.2 Recurrent Q-network

Although the player–interactive narrative planner interaction environment has been often assumed to be Markovian in previous RL-based interactive narrative personalization works [Rowe et al., 2014; Wang et al., 2016a], the assumption does not hold in practice in interactive narrative generation. To model interactions in a partially observable environment, a common solution is to stack a fixed length of observation history into the RL state representation. However, there are other ways to concisely encode long-term history, such as the adoption of RNNs. Specifically, the recurrent Q-network embeds an LSTM layer within the Q-network [Hausknecht and Stone, 2015], which enables the Q-network to effectively preserve a long-term memory in the narrative but maintains a compact state representation as $s_t = o_t$ (Figure 3b). This structure may help with situations when long interaction patterns influence interactive narrative quality severely, i.e., late stage interactive narrative planning decisions being strongly affected by early stage events.

5.3 Dueling Q-network

Because state and policy spaces in interactive narrative personalization problems are often vast, utilizing advantage learning [Harmon and Baird, 1995] by evaluating state values independently can be valuable. State values give the RL agent the estimation of the "goodness" of a state, even without all actions being thoroughly explored. Thus, in early stages in interactive narrative planner training, advantage learning might be able to guide the interactive narrative planner to avoid "bad states" early and explore "good states" more thoroughly.

Under the Q-network structure, this notion can be implemented using a dueling Q-network [Wang et al., 2016b]. As shown in Equation 3, an action's *Q* values can be represented by the summation of a state value *V* and an action advantage value *A*. Corresponding to Figure 3c, θ' represents parameters of the Q-network below the dueling layer, and α, β represent parameters in the advantage stream and state stream, respectively. A merging module sums the output from these two streams to obtain the same output format as other Q-networks.

$$Q(s,a;\theta',\alpha,\beta) = V(s;\theta',\beta) + A(s,a;\theta',\alpha) \quad (3)$$

6 Evaluation

To evaluate the performance of interactive narrative planning policies, an evaluation was conducted with interactive narrative planning Q-networks distinctly configured in three dimensions. The first dimension is the choice of stabilization techniques. Because both *experience replay* and *asynchronous gradient descent optimization* methods are compatible with the *separate target network* method in our implementation, we either combine *experience replay* with *separate target network* (Q-net) or integrate *asynchronous gradient descent optimization* with *separate target network* (Async Q-net). The second configuration dimension is based on the manner in which observation history is embedded into RL state representations. The interactive narrative planning Q-networks either take a fixed length observation in the state representation, or utilize the recurrent Q-network structure by substituting one hidden layer with an LSTM layer. This configuration is labeled History Length. The third dimension of configuration is exploiting (Dueling Q-net, Async Dueling Q-net) or not exploiting the dueling structure. Structures of these Q-networks are shown in Figure 3. For all of these structures, Hidden Layer 1 has 64 neurons, and Hidden Layer 2 (either fully connected layer or LSTM) has 32 neurons. The advantage stream and state value stream both contain 32 neurons in dueling structures. We use Adam, a first-order gradient-based optimization method, to train the weights for all of the Q-networks [Kingma and Ba, 2015]. As a baseline, we also train a linear RL interactive narrative planning model (Linear). All of these RL models are trained with the same training set-based simulated players, and evaluated by interacting with the same test set-based simulated players for 10,000 episodes. Each RL model has been trained until they converge to optimized narrative planning policies (Table 1). Training the Q-net and Dueling Q-net requires approximately

8 hours without GPU acceleration. The Asyn Q-net and Async Dueling Q-net require approximately 2 hours to converge on 4 threads.

For measuring derived interactive narrative planning policies, we tried *importance sampling* and *weighted importance sampling*, as used in [Wang et al. 2016a]. However, both of the methods generate highly skewed estimations. Thus, we adopt the method of evaluating policies on test set-based simulated players, which follows the evaluation convention in the field of spoken dialogue systems when simulated user models are utilized [Henderson et al., 2005; Schatzmann et al., 2006]. The test set-based player simulation model is constructed and trained in the same way as it is for the training set-based player simulation model.

Hist. Length	Linear	Q-net	Dueling Q-net	Async Q-net	Async Dueling Q-net
1	0.0540	0.0586	**0.1698**	0.0942	**0.1042**
2	0.0764	0.0880	0.1262	0.1114	0.0860
3	**0.0992**	0.0770	0.1320	0.0748	0.0958
4	0.0788	0.1066	0.1272	**0.1128**	0.0740
LSTM	–	**0.1294**	0.1252	0.0874	0.0976

Table 1. Evaluated policy values of interactive narrative planner trained with each type of Q-network structures on 10,000 interactions with test set-based simulated players. According to reward signal design, policy's value is in range of [-1,1], where a uniform random policy has value of 0.0482. Discount factor γ is set to 1.

As seen in Table 1, Q-network based RL interactive narrative planners derive better-performing policies than the linear RL interactive narrative planner does. Although all of the Q-networks converge to locally optimal policies in training, we have observed that the stabilization techniques affect policy quality for interactive narrative personalization. An interesting finding is that the performance of the *asynchronous gradient descent optimization* method is correlated with the number of trainable parameters in Q-networks (as shown in Table 2). In our experiment, when the trainable parameter number is relatively small (5,674 for Q-net and Async Q-net), the asynchronous method generates better or comparable policies (Table 1 rows 2–5 in 5th column) than Q-net policies (3rd column of Table 1). However, when either a recurrent structure or dueling structure is added, which dramatically increases the number of trainable parameters in Q-networks, the asynchronous method diminishes the interactive narrative planning policy's performance in CRYSTAL ISLAND.

Hist. Length	Q-net	Dueling Q-net	Async Q-net	Async Dueling Q-net
1–4	5674	7787	5674	7787
LSTM	14410	26859	14410	26859

Table 2. Number of trainable parameters in Q-networks.

From Table 1, we find that utilizing the dueling structure significantly improves the Q-network's performance when *experience replay* and *separate target network* are used for stabilization with fixed-length observation RL states (4th column). Dueling Q-networks achieve best performance with a short observation history encoded into the RL state representation (4th and 6th columns of Table 1). Q-networks without a dueling structure achieve their best performance with long observation histories (3rd and 5th columns of Table 1). Because dueling Q-networks use a separate stream to estimate RL state values, results suggest that more compact narrative representations may be more effective for estimating state values when training an interactive narrative planner for CRYSTAL ISLAND.

We also investigate the effectiveness of incorporating recurrent structures into the Q-networks (bottom row of Table 1) to extract representative RL states. The LSTM structure yields better performance for the Q-net planner (3rd column of Table 1) than all other fixed-length state representations. Overall, fixed-length state representations yield better performance for the other types of Q-networks (4th through 6th columns of Table 1), but the LSTM structure does produce policy values that are comparable to several fixed-length state representations. Given these mixed results, additional investigation of recurrent Q-network structures for interactive narrative personalization is merited.

The intuition for the quality of derived interactive narrative planning policies is as follows: out of the 10,000 interactions between the derived interactive narrative planner and test set-based simulated players, the best-performing Q-network (with policy value of 0.1698) agent led 5,849 players to reach to a high NLG (high learning gain), compared to the best performing linear RL agent with 5,496 players reaching high NLG (an evaluation value of 0.0992). It should be noted that the random policy guided only 5,241 players towards a high NLG. Normalizing the policy values into the scope of [0,1], the performance improvement of the optimally configured Q-network interactive narrative planner over the best linear RL agent is 6.42%. Given the challenges presented by modeling human players' behaviors in interactive narratives, this improvement is significant.

7 Conclusion

Data-driven approaches to interactive narrative planner offer considerable promise for generating personalized interactive narratives. We have presented a Q-network based deep RL framework that features a bipartite player simulation model. Utilizing a Q-network improves an interactive narrative planner's ability to extract non-linear complex player interaction patterns, and the long short-term memory network-based player simulation model supplies Q-networks with unlimited training and testing data by synthesizing sequential player actions and outcomes separately. Results of an evaluation suggest that a properly configured Q-network RL interactive narrative planner can significantly outperform a linear RL-based interactive narrative planner. In future work, it will be important to investigate RL-based interactive narrative planners for both education and entertainment to further explore their potential to create effective interactions to support a broad range of player populations and a wide array of genres of interactive narrative.

References

[Harmon and Baird, 1995] M. Harmon, L. Baird, and A. Klopf. Advantage updating applied to a differential game. In *Proceeding of Advances in Neural Information Processing Systems*, pages 353–360, 1995.

[Hausknecht and Stone, 2015] M. Hausknecht, and P. Stone. Deep Recurrent Q-Learning for Partially Observable MDPs. In *AAAI Fall Symposium Series*, 2015.

[Henderson et al., 2005] J. Henderson, O. Lemon, and K. Georgila. Hybrid reinforcement/supervised learning for dialogue policies from communicator data. In *IJCAI Workshop on Knowledge And Reasoning in Practical Dialogue Systems*, pages 68–75, 2005.

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

[Kingma and Ba, 2015] D. Kingma, and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference for Learning Representations*, 2015.

[Lamstein and Mateas, 2004] A. Lamstein, and M. Mateas. Search-based drama management. In *2004 AAAI Workshop on Challenges in Game Artificial Intelligence*, pages 103–107, 2004.

[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. *IEEE Transactions on Computational Intelligence and AI in Games*, 6: 203–215, 2014.

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

[Lin, 1993] L.J. Lin. Reinforcement learning for robots using neural networks. Technical Report, *DTIC* Document, 1993.

[Mnih et al., 2015] V.Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518 (7540): 529–533, 2015.

[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 *Proceedings of 33rd International Conference on Machine Learning*. 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 *Proceedings of the 5th International Conference on Autonomous Agent and Multi-Agent Systems*, 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 *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, 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 *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, pages 1213-1218, 2006.

[Robertson and Young, 2015] J. Robertson, and R. Young. Automated Gameplay Generation from Declarative World Representations. In *Proceedings of the 11th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 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 *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 160–166, 2014.

[Schatzmann et al., 2006] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2): 97–126, 2006.

[Tsitsiklis and Roy, 1997] Tsitsiklis, J. and Roy, B. V. An Analysis of Temporal-Difference Learning with Function Approximation. IEEE Trans. Automat. Contr. 42, 674–690, 1997.

[Wang et al., 2016a] P. Wang, J. Rowe, B. Mott and J. Lester. Decomposing Drama Management in Educational Interactive Narrative: A Modular Reinforcement Learning Approach. In *Proceedings of the 9th International Conference on Interactive Digital Storytelling*, pages 270–282, 2016.

[Wang et al., 2016b] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas. Dueling Network Architectures for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1995-2003, 2016.

[Young et al., 2013] S. Young, M. Gašić, B. Thomson, and J. Williams. Pomdp-based Statistical Spoken Dialog Systems: A Review. Proceedings of IEEE, 101(5): 1160-1179, 2013.

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