Intelligent Tutoring Strategies for Students with Autism Spectrum Disorder: A Reinforcement Learning Approach

Stephanie Milani*[1], Zhou Fan*[2], Saurabh Gulati[3], Thanh Nguyen[4], Fei Fang[1], and Amulya Yadav[5]
[1]Carnegie Mellon University [2]Shanghai Jiao Tong University [3]ZS Associates
[4]University of Oregon [5]Pennsylvania State University

Abstract

Due to their socialization and communication difficulties, young children with autism spectrum disorder (ASD) face many challenges in traditional school environments. Intelligent tutoring systems (ITS) are a promising technology for supplementing instruction for children with ASD. However, previous work in the design of such systems often does not consider crucial cognitive features of students, such as working memory deficits and the zone of proximal development (ZPD), which, at a high level, refers to the difference between what a student can accomplish on their own and what they can accomplish with the help of a knowledgeable other. Furthermore, these systems are frequently not tailored for use by children with ASD, potentially causing significant shortcomings in the effectiveness of their learning. Aiming to address these issues, we propose RELETAS, an intelligent tutoring system which uses a state-of-art reinforcement learning algorithm to learn effective, customized tutoring policies for children with ASD. RELETAS also incorporates working memory deficits and ZPD in its student models. We provide initial results of our system on a digit identification task and discuss potential directions for future work.

Introduction

Autism spectrum disorder (ASD) is one of the most severe childhood neuro-psychiatric disorders, and it has emerged as a major public health concern in the United States. In 2018, one in every 59 children suffered from ASD (Baio et al. 2018). The number of children with ASD diagnoses has markedly increased over the last decade, as has the demand for services. ASD is characterized by resistance to change, ritualistic and repetitive behavior, and problems in social development, communication, executive function, and generalization (American Psychological Association 2018; de Marchena, Eigsti, and Yerys 2015). As a result, in comparison with their peers, students with ASD can experience significant challenges from many aspects of classroom learning, such as those relating to socialization, communication, and generalization (Gobbo and Shmulsky 2012).

*Equal contribution

Intelligent tutoring systems (ITS) are a promising means of teaching children with ASD (Mondragon et al. 2017). However, most existing ITS are not directly applicable for children with ASD for several reasons. First, most existing ITS are designed for use by neuro-typical school-going children and are not tailored for children with ASD. For example, existing ITS do not account for the working memory deficits that are often seen among children with ASD (Barendse et al. 2013). Similarly, they do not consider the zone of proximal development (ZPD) of children with ASD. The ZPD of children with ASD is important to consider, as a child with ASD typically experiences a number of challenges when in the ZPD, including, but not limited to, becoming overwhelmed with new experiences (Sahin et al. 2018). By not considering the ZPD of these students, an ITS may pose questions that cause the student to either become bored or discouraged. Second, most existing ITS output static tutoring strategies, where each child is presented with the same sequence of lessons and examples irrespective of their performance on previous lessons. Customized tutoring strategies are especially desirable for students with ASD, as the challenges each individual faces differ widely; thus, teaching policies must be tailored accordingly. Reinforcement learning (RL) offers one way of customizing these tutoring strategies; however, there is little existing work on RL-based ITS for children with ASD, with one exception discussed in detail in Related Work.

This paper presents Reinforcement Learning for Effective Tutoring of Autistic Students (RELETAS), a novel ITS which uses RL to learn adaptive long-term tutoring policies for teaching children with ASD. In the development of RELETAS, we address the shortcomings in previous ITS with the following contributions. First, we propose a parameterized learning model for the student's learning process which explicitly takes into account the varying ZPD, memory capabilities, and learning capability of each ASD student. Second, we formulate the problem of sequentially recommending lessons and questions to a student with ASD as a partially observable Markov decision process (POMDP) (Kaelbling, Littman, and Cassandra 1998). The transition probabilities are defined implicitly through the student's learning model, and thus indirectly depend on the

individual student's characteristics. Third, we use deep RL to learn effective and customized policies for the students. We use an LSTM (Hochreiter and Schmidhuber 1997) for the RL policy in order to capture the the temporal dependencies of the questions presented to the student. We also use a hierarchical representation of the action space, where we leverage the success of unsupervised clustering techniques to significantly reduce the action space of our problem. We provide initial results of our system on a digit identification task and discuss potential directions for future work.

Related Work

There exists a long history of research in developing ITS for students using sequential planning models, with a specific focus on POMDP planning to account for uncertainties in the teaching process. Previous work uses POMDPs to tackle uncertainties in the mental processes (Folsom-Kovarik, Sukthankar, and Schatz 2013) and emotional states (Theocharous et al. 2009) of students. Both approaches use a distinct POMDP per student and define an exact model of the transition probabilities for each student a priori. In practice, students have different underlying transition models; knowing these models in advance is unrealistic.

There has also been work in specifically using RL in an ITS. To our knowledge, RL was first incorporated into an ITS for students with ASD by Sarma and Ravindran (2007), who proposed their framework to teach students with ASD to differentiate between four different patterns. In their approach, they use the negative of the mean square error of the output of the student neural network model as the reward for the RL algorithm. As such, they do not consider whether the questions are too difficult or too easy for the student. Additionally, they do not incorporate a memory model or the ZPD in their representation of autistic students.

Using RL in ITS was further extended by Chi, Vanlehn, and Litman; Malpani, Ravindran, and Murthy (2010; 2011). In the former work, they learn Markov transitions from a set of data collected using random tutoring strategies and use it to infer a successful tutoring strategy; however, this approach is infeasible using real data, as most existing data involving student-ITS interactions does not involve a random strategy, which can lead to data sparsity in the collected data. In the latter work, they use RL to implicitly train the ITS with an adaptive student model that estimates the learning parameters of the student; however, they do not directly consider the memory nor the ZPD of the student, nor do they directly consider autistic students.

Recently, RL was applied to POMDP models for intelligent tutoring systems (Wang 2018; 2014); however, students with ASD were not directly considered in those works. In these papers, the reward for the RL algorithm is based solely on student preferences: that is, the RL algorithm receives a negative reward if the student rejects the question and a positive reward if the student accepts the question. These methods were evaluated only on the student rejection rate and not learning outcomes (such as performance on a final exam in the subject being taught).

Problem Setting

We describe the abstract environment within which our ITS operates. We assume that there a set of N distinct concepts $c_1, c_2, ...c_N$ that we want the student with ASD to learn. Each concept could correspond to a different part of speech, different mathematical operations, and so on. As previously discussed, a critical challenge faced by students with ASD is generalizing past knowledge to slightly modified questions. To ensure that these students can overcome this difficulty, we assume that for each concept c_i, there exists a set of questions and examples j_i that correspond to that particular concept. For example, suppose c_i corresponds to the concept representing nouns. Then, there exists a set of questions and/or examples j_i pertaining to nouns which the ITS can ask the student to complete. In the real world, this question bank of examples can be created with the help of teachers and special-needs educators.

Given a question bank as input, our ITS, RELETAS, runs for a pre-determined number of rounds $\mathcal{Q}$, where $\mathcal{Q}$ refers to the number of questions attempted by the student. In each round $q = 1, 2, ...\mathcal{Q}$, RELETAS selects a question that corresponds to a particular concept and asks the student to attempt to answer that question. In response, the student provides an answer to the query and whether they believe that they are a bit confused or bored. The performance (i.e., correct or incorrect answer) and the feedback of the student (i.e., whether they are feeling a bit bored or confused) on this question is used as feedback to update the ITS question selection policy for future rounds. The goal of RELETAS is to find an optimal question selection policy, which maximizes learning outcomes for students with ASD.

Handwritten Digit Identification

For exposition, we instantiate this environment with a real-world learning task of: "teaching students with ASD how to identify written digits." While we focus on this specific task, our environment is general enough to accommodate many real-world learning scenarios for autistic students, such as emotion recognition (Golan et al. 2009). In the digit identification task, students must label images of handwritten digits between 0-9. We focus on this task for the following reasons. First, students with ASD tend to have difficulties with learning and remembering basic mathematical objects and concepts (Winoto et al. 2018), so this task is relevant. Second, due to the variance of shapes for individual digits in handwritten digits, accurate recognition of handwritten digits represents a setting in which autistic students must generalize their knowledge about digit shapes to different examples. Because autistic students tend to struggle with generalization, this setting is challenging. Third, the general setting of multi-class classification is a common one encountered by students during traditional schooling.

The MNIST dataset (LeCun et al. 1998) consists of $70,000$ images of handwritten digits. In our setting, each separate digit $i \in \{0, ..., 9\}$ represents a distinct concept (for a total of $N = 10$ concepts), and j_i refers to the set of datapoints in the MNIST dataset which are labeled as digit i. We split the dataset into a training question bank consisting of

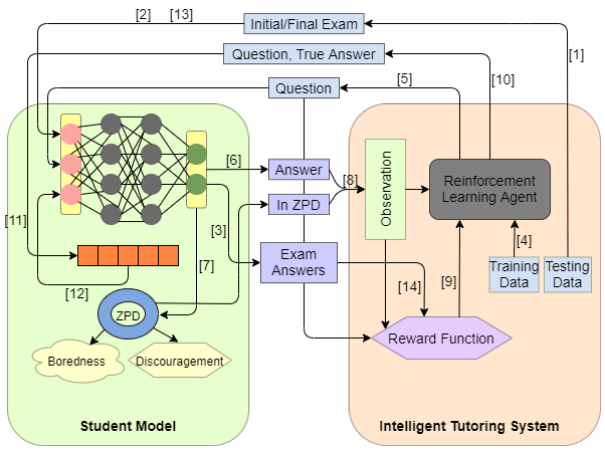

Figure 1: A high-level diagram showing the interaction between the student model and the ITS during the RL training process. At the start of each episode, the student is given the hold-out test questions [1] to take the initial exam [2] using the initialization of student NN [3]. Then, at each timestep, the RL agent chooses a question from the training data [4] to ask the student [5]. The student NN answers the selected question [6]. The output from the NN is used to determine whether the question was in the student's ZPD [7]. That information and the answer is observed by the RL agent [8]. The RL agent uses that information to choose a new question in the next timestep. The reward function uses that observation and the question to provide a reward to the RL agent [9]. Before the student is presented with the new question, the student is given the answer to the previous question [10]. The question-answer pair is stored in the student's working memory [11]. All elements in the short term memory are used to train the student NN [12]. At the end of the episode, the student takes the final exam with the same test questions as in the initial test [13]; the improvement in the test scores is used as part of the reward for the RL agent [14].

$60,000$ questions and answers and a testing question bank consisting of $10,000$ questions and answers.

RELETAS

We now describe RELETAS, a novel ITS that uses RL to teach students with ASD. The goal of RELETAS is to find an optimal question selection policy, which maximizes learning outcomes for students with ASD. RELETAS consists of the following components: a student model, a POMDP, and an RL algorithm. We use the student model to approximately model autistic student learning behavior when interacting with an ITS. By interacting with the students through the POMDP model, the RL algorithm learns a policy for presenting questions to the students. We show the high-level process of the student-ITS interaction in Figure 1.

Student Model

We create two populations of synthetic students: one for training and one for testing. Each student has a parameterized function for answering questions posed by the RL policy, a memory model, and a ZPD model. We describe our model of students with ASD in detail.

Student Neural Network We model the student's current capability with a parameterized function $\mathcal{F}$, which takes as input a matrix representation of an image and outputs a probability distribution over possible labels. We can select different functions depending on the task. For example, if we want the student to make predictions based on time-series data, we could use an LSTM (Hochreiter and Schmidhuber 1997) as our parameterized function. We model the learning process of the student as updating the model parameters as the student is provided with more examples and labels from an expert. The update rule used here can be task dependent and provided by domain experts.

As noted by Cohen (1994), artificial neural networks trained with backpropagation can approximately model the generalization abilities of autistic individuals. At a high level, they report that ANN models with either very small neuronal connection densities or very high neuronal connection densities resemble the capabilities of individuals with ASD. In our work, we use a convolutional neural network with a large number of neurons to model the decision-making process of the student. We choose to use a CNN following the previous literature on visual processing in ASD (Nagai, Moriwaki, and Asada 2015) and because CNNs have recently been shown to align with biological object recognition in people (Kuzovkin et al. 2018). When training the model, we use Adam (Kingma and Ba 2015) as our gradient descent optimization algorithm and categorical cross entropy loss as our loss function.

Student Memory Model Students do not have perfect memories. As a result, they tend to forget questions or examples that they may have successfully attempted in the past. In our approach, the student maintains a fixed-length memory queue of the past n questions and answers. When the student is asked to answer a new question and is given the proper answer to it, they retrieve from memory all n of the previous questions and answers stored in the memory queue. These examples are used to train the student ANN for a single epoch. This model is similar to a student's working memory. After being presented with a new question, a student may then recall all previously-given questions and answers and use those question-answer pairs to strengthen their knowledge in those areas.

Zone of Proximal Development Model The theory of the zone of proximal development (ZPD) refers to the difference between what a student can accomplish on their own and what a student can accomplish with the help of a knowledgeable other (Murray and Arroyo 2002). In other words, for a posed question to be in a student's ZPD, it must not be too difficult or too easy for the student to answer. As a result, the student does not become too bored or confused, respectively, and the question is at an appropriate level that leads to their proximal development.

We use the CNN model to incorporate the student's zone of proximal development in our student models. More for-

mally, the output from the CNN $\mathcal{F}(m_q)$ can be equivalently viewed as the probability of the student providing the correct answer to question m_q. Each student has a student-specific parameter β, which we select for each student by sampling uniformly at random from the range $[0.3, 0.5]$. The ZPD of the student is defined as follows: the student remains in their ZPD after attempting question m_q if and only if $\mathcal{F}(m_q) \in [0.5 - \beta, 0.5 + \beta]$. Note that, in this definition, $\mathcal{F}(m_q)$ represents the ease of successfully completing question m_q. If $\mathcal{F}(m_q) \in [0, 0.5 - \beta]$, then that implies that the student found the question m_q too easy, whereas if $\mathcal{F}(m_q) \in [0.5 + \beta, 1]$, then that implies that the student found question m_q too difficult.

Each time the student is given a question that is outside of their ZPD, either the student's level of boredom or level of discouragement is increased by 1. Once the student becomes too bored or too discouraged, the student stops answering questions entirely and the interaction with the ITS ends.

POMDP Model

We now describe the POMDP model used in RELETAS. A partially observable Markov decision process (POMDP) (Kaelbling, Littman, and Cassandra 1998), $M = \langle \mathcal{S}, \mathcal{A}, \Omega, \mathcal{P}, \mathcal{O}, R, \gamma \rangle$, is comprised of: the state space $\mathcal{S}$; the action space $\mathcal{A}$; the observation space Ω; the transition probability distribution $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$; the conditional observation probability $\mathcal{O} : s' \times a \rightarrow o$; the reward function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$; and a scalar discount factor $\gamma \in (0, 1]$, which governs the importance of future rewards. In the POMDP setting, the algorithm interacting with the environment does not directly observe the state and, instead, gets potentially noisy observations of the state.

Actions Instead of representing each question in the question bank as a possible action, we organize the action space in the following way. Before the ITS interacts with the student, we separate the training data based on its label. Within each concept, we perform k-means clustering (Lloyd 1982) to construct k clusters of the data, where k is a tunable parameter that we can set. We choose to cluster the actions to capture similarity between questions. The ITS selects a concept and cluster from which it would like to present a question to the student. The exact question is then selected uniformly at randomly from that cluster. By clustering the actions, we reduce the action space significantly — from $60,000$ (if we were to represent each individual question from the set of training questions as an action) to $N \times k$, where N is the number of concepts and k is the number of clusters within each concept.

Observations The underlying state of the POMDP depends on the student network, which the RL algorithm does not have access to; instead, the RL algorithm receives incomplete observations of the state. At each timestep, after performing an action, the RL algorithm receives an observation $o \in \Omega$ of the following form: $o = \langle \tau, c_i, j_i, l_i, f, z \rangle$.

The first component τ denotes the total number of rounds remaining in the student-ITS interaction. When $\tau = 0$, the episode terminates. The second component c_i denotes the concept corresponding to the question that the student was asked in the previous round. The third component j_i denotes the cluster within the concept from which the question asked in the previous round was drawn. The fourth component $l_i = \{1, ..., N\}$ denotes the label (answer) that the student gave for the question in the previous round. Although the student neural network outputs a probability distribution over labels, our RL algorithm only observes the answer that the student thinks is most likely to be true. The fifth component $f = \{0, 1\}$ is a binary flag that denotes whether the student correctly answered the question from the last round.

The sixth component $z = \{0, 1\}$ is a binary flag that denotes whether the student noted that the question increased their confusion or boredom. If $z = 1$, that indicates that the question was likely outside of that student's ZPD. In a real setting, we could get this observation by either asking the student to answer how they are feeling or how difficult they perceive the question to be when they answer each question, or by enabling the ITS to have access to indicators of attention, such as eye gaze (D'Mello et al. 2012), and/or underlying emotional state (Mauss and Robinson 2012).

Rewards Our reward function $R(s, a)$ used in training the RL algorithm is defined as follows. At the start of the episode, the student takes the final exam, which is the question bank of hold-out test examples. Their score is recorded as σ_o. At each timestep during training, when the student is presented with questions by the ITS, if the student correctly answers the question and the question was within the student's ZPD, then the RL algorithm receives a reward of 2. If the student incorrectly answers the question, but the question was within the student's ZPD, the RL algorithm receives a reward of 1; the same goes for if the student correctly answers the question, but the question was not within the student's ZPD. If the question was neither answered correctly, nor within the student's ZPD, then the RL algorithm receives a reward of 0. At the end of the episode, the student takes the final exam again. Their score σ_f is recorded. The reward in the last step thus has an additional term which is $(\sigma_f - \sigma_o) \times 100$. We scale the additional term because the primary objective of the RL algorithm is to ensure that the student performs well on the final exam.

Reinforcement Learning

Our goal is to develop an ITS which customizes its question selection policies to each individual student's capabilities. As such, we choose to use RL to learn a customized policy for presenting questions to students. In RL, the goal of the algorithm is to determine the most rewarding behavior over time, as represented by a policy which maps observations to actions. Policy gradient algorithms are a family of RL algorithms, where the policy is directly modeled and optimized. Proximal policy optimization (PPO) algorithms are a type of policy gradient algorithms that have shown great performance on a variety of tasks (Schulman et al. 2017). We use the version of PPO created by Hill et al. (2018) as our RL algorithm. To capture the temporal dependencies of the questions presented to the student, we use an LSTM (Hochreiter and Schmidhuber 1997) as our policy network.

Experiments

In all experiments, we compare the performance of PPO to a random baseline, where the students are randomly presented with questions to answer. For PPO, we use a learning rate of 1×10^{-6} for the first and third experiments and a learning rate of 1×10^{-6} for the second experiment. For all experiments, we use $k = 3$ for the action clustering For all experiments, we train PPO for $30,000$ rounds (timesteps), and set the maximum length of each episode to be 100 rounds. As a result, the number of episodes for each experiment differs.

We assume that the students have some prior knowledge related to the learning task. This assumption corresponds to pretraining the student population with a small number of examples from the training set. To perform this pretraining procedure, we uniformly at random select 400 examples from the training dataset and train the CNN for three epochs before the student interacts with the ITS. . We choose $n = 5$ for the length of the student memory queue.

In the testing phase, we cease the training of PPO. We generate 200 new students to interact with PPO and the random baseline and report the performance of the students on the initial pretest (when they rely on their pretrained knowledge) and the performance of the students on the final test (after they have interacted with either the baseline or PPO).

We conduct the following experiments. In the first experiment (setting one), we vary the student-specific parameter β in our training and testing populations of students. In other words, each student has a different ZPD. In the second experiment (setting two), we vary β, as in setting one, and we vary the weights in the weight vector for the 10 different concepts by independently sampling it. By varying the weights, we change how much examples from each of the concepts contributes to the loss. This sampling is performed before the pretraining phase, resulting in different prior knowledge for each student. In the third experiment (setting three), we vary also β. Instead of independently sampling the weight vector, as in setting two, we use a fixed weight vector for each of the 10 concepts. This sampling is performed before the pretraining phase, meaning all students should have a similar distribution of prior knowledge, but it is imbalanced among the 10 digits.

Results

We report our preliminary results on the aforementioned experimental settings. In general, we find that the policies learned using RL are able to achieve greater reward with training. We also find that, in settings one and three, the policy learned using RL performs better on average than the random baseline. However, we need to perform more experiments to determine if the results are statistically significant.

Setting One

In Figure 2, we show the learning curve of PPO in setting one. We see that, over time, PPO receives more reward, then eventually levels out. At the end of training, PPO receives a reward of around $1,450$, which means that the final student with which PPO interacted correctly answered $1,450$ out of $10,000$ queries on the final exam.

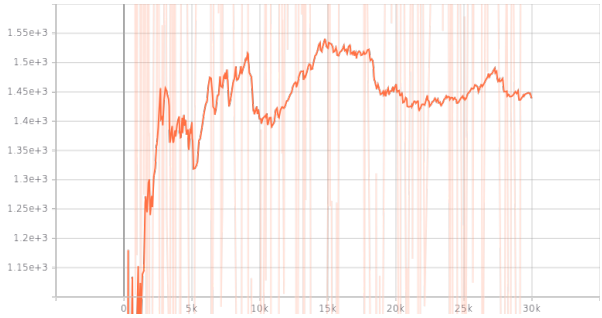

Figure 2: Learning curve of PPO for setting one. The x axis shows the number of timesteps that the ITS interacts with the student. The y axis shows the total reward for each episode.

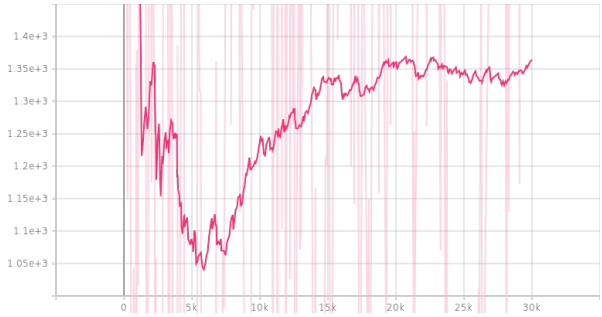

Figure 3: Learning curve of PPO for setting two. The x axis shows the number of timesteps that the ITS interacts with the student. The y axis shows the total reward for each episode.

In Table 1, we show the results of PPO compared to a random baseline on a hold-out test population of 200 students. Compared to the baseline, on average, PPO achieves higher episode reward, leads to a larger improvement on the final test, and interacts with the students for more timesteps per episode. We were surprised that using PPO led to higher episode reward on average and a larger improvement on the final test: we expected the performance of the RL algorithm to be on par with random, as noted by Sarma and Ravindran (2007) in a similar problem setting. One explanation is that the RL algorithm kept the students in their ZPD for longer, so the RL algorithm could present the student with more questions than the random baseline.

	Average Episode Reward	Average Improvement	Average Episode Length
PPO	1512.26	14.32	68.85
Random	1357.25	12.87	64.95

Table 1: Test results for setting one.

Setting Two

In Figure 3, we show the learning curve of PPO in setting two. We can see that, over time, PPO receives more reward,

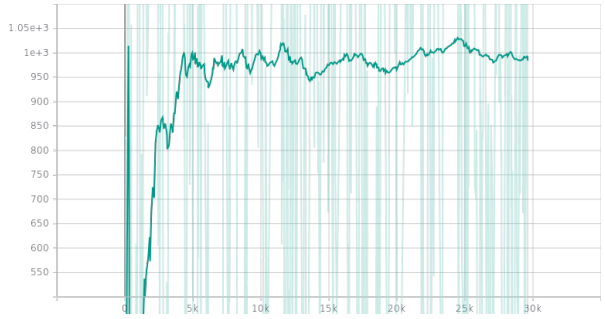

Figure 4: Learning curve of PPO for setting three. The x axis shows the number of timesteps that the ITS interacts with the student. The y axis shows the total reward for each episode.

then eventually levels out. However, compared to the learning curve for setting one, PPO receives less reward in general. Also, notably, the reward drops sharply during the training procedure, which may be explained by the algorithm encountering a student whose parameters (such as ZPD) made it such that the current policy did not work well with that student. At the end of training, PPO receives a reward of around $1,350$, which means that the final student that PPO interacted with was able to answer $1,350$ out of $10,000$ queries correctly on the final exam.

In Table 2, we show the results of PPO compared to a random baseline on a hold-out test population of 200 students. Compared to a random baseline, on average, PPO achieves lower episode reward, leads to less improvement on the final test, and interacts with the students for less timesteps per episode. The results for setting two could be explained by the presence of more variance in the student parameters. Due to the higher overall variance, it could be more challenging for the RL algorithm to learn in this environment. As a result, PPO performance is worse than the random baseline.

	Average Episode Reward	Average Improvement	Average Episode Length
PPO	1206.53	11.46	64.73
Random	1275.05	12.06	66.84

Table 2: Test results for setting two.

Setting Three

In Figure 4, we show the learning curve of PPO in setting three. We can see that, over time, PPO receives more reward, then eventually levels out. At the end of training, PPO receives a reward of around $1,000$, which means that the final stuent that PPO interacted with was able to answer $1,000$ out of $10,000$ queries correctly on the final exam.

We also report the results of deploying the trained PPO policy and a random baseline on a hold-out test population of 200 students. In Table 3, we show these results. Compared to a random baseline, on average, PPO achieves higher episode reward, leads to a larger improvement on the final test, and interacts with the students for more timesteps per episode. Compared to setting two, PPO may perform better than the random baseline because there is less variance in the student parameters than in setting two.

	Average Episode Reward	Average Improvement	Average Episode Length
PPO	1054.20	9.83	65.73
Random	940.12	8.80	63.61

Table 3: Test results for setting three.

Discussion and Future Work

The results of our experiments to some extent show promise for using RL to learn tutoring policies for teaching students with autism spectrum disorder. However, the variance of performance metrics is notably high among different episodes since the student-specific parameter and pre-training data are both independently sampled for the synthetic student in each episode. Therefore, as part of future work, we plan to run more experiments for a larger number of test episodes comparing the random policy with the PPO policy to discern whether the performance difference between the two approaches is indeed statistically significant.

We also plan to improve upon the POMDP model by exploring different ways of modeling the action space, such as clustering questions based on difficulty. Another direction is to test our framework on domains of structured knowledge, where students must master certain skills before learning new ones, such as the task of learning calculus or a foreign language. Furthermore, we plan to expand our student model by using a more complex representation of student memory inspired by work in cognitive science and by incorporating more student-specific parameters, such as frustration. The student-specific parameters could be sampled from distributions pre-defined with domain knowledge, but could also be fitted to approximate real students when trajectory data of learning procedure of a population of students is available. Finally, we would like to show that our work can be used by wide variety of students. To demonstrate this, we plan to test our framework using some of the student models described by Lanillos et al. (2019).

Acknowledgments

Co-author Fang is supported in part by NSF grant IIS-1850477.

References

American Psychological Association. 2018. What is autism spectrum disorder? https://www.psychiatry.org/patients-families/autism/what-is-autism-spectrum-disorder.

Baio, J.; Wiggins, L.; Christensen, D.; Maenner, M. J.; Daniels, J.; Warren, Z.; Kurzius-Spencer, M.; Zahorodny,

W.; Rosenberg, C. R.; White, T.; Durkin, M. S.; Imm, P.; Nikolaou, L.; Yeargin-Allsopp, M.; Lee, L.-C.; Harrington, R.; Lopez, M.; Fitzgerald, R. T.; Hewitt, A.; Pettygrove, S.; Constantino, J. N.; Vehorn, A.; Shenouda, J.; Hall-Lande, J.; Braun, K. V. N.; and Dowling, N. F. 2018. Prevalence of autism spectrum disorder among children aged 8 years — autism and developmental disabilities monitoring network, 11 sites, United States, 2014. *Morbidity and Mortality Weekly Report Surveillance Summaries* 67(6).

Barendse, E. M.; Hendriks, M. P.; Jansen, J. F. A.; Backes, W. H.; Hofman, P. A. M.; Thoonen, G.; Kessels, R. P. C.; and Aldenkamp, A. P. 2013. Working memory deficits in high-function adolescents with autism spectrum disorders: Neuropsychological and neuroimaging correlates. *Jour. of Neurodevelopmental Disorders* 5.

Chi, M.; Vanlehn, K.; and Litman, D. 2010. Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutor tactics. In *Int. Conf. on Intelligent Tutoring Systems*.

Cohen, I. L. 1994. An artificial neural network analogue of learning in autism. *Biological Psychiatry* 36(1).

de Marchena, A. B.; Eigsti, I.-M.; and Yerys, B. E. 2015. Brief report: Generalization weaknesses in verbally fluent children and adolescents with autism spectrum disorder. *Jour. of Autism and Developmental Disorders* 45(10).

D'Mello, S.; Olney, A.; Williams, C.; and Hays, P. 2012. Gaze tutor: A gaze-reactive intelligent tutoring system. *Int. Jour. of Human-Computer Studies* 70(5).

Folsom-Kovarik, J. T.; Sukthankar, G.; and Schatz, S. 2013. Tractable POMDP representations for intelligent tutoring systems. *ACM Trans. on Intelligent Systems and Tech.* 4(2).

Gobbo, K., and Shmulsky, S. 2012. Classroom needs of community college students with asperger's disorder and autism spectrum disorders. *Community College Jour. of Research and Practice* 36(1).

Golan, O.; Ashwin, E.; Granader, Y.; McClintock, S.; Day, K.; Leggett, V.; and Baron-Cohen, S. 2009. Enhancing emotion recognition in children with autism spectrum conditions: An intervention using animated vehicles with real emotional faces. *Jour. of Autism and Developmental Disorders*.

Hill, A.; Raffin, A.; Ernestus, M.; Gleave, A.; Kanervisto, A.; Traore, R.; Dhariwal, P.; Hesse, C.; Klimov, O.; Nichol, A.; Plappert, M.; Radford, A.; Schulman, J.; Sidor, S.; and Wu, Y. 2018. Stable baselines. https://github.com/hill-a/stable-baselines.

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

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1).

Kingma, D. P., and Ba, J. L. 2015. Adam: A method for stochastic optimization. In *Int. Conf. on Learning Representations*.

Kuzovkin, I.; Vicente, R.; Petton, M.; Lachaux, J.-P.; Baciu, M.; Kahane, P.; Rheims, S.; Vidal, J. R.; and Aru, J. 2018. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Communications Biology* 1.

Lanillos, P.; Oliva, D.; Philippsen, A.; Yamashita, Y.; Nagai, Y.; and Cheng, G. 2019. A review on neural network models of schizophrenia and autism spectrum disorder. *arXiv:1906.10015*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*.

Lloyd, S. P. 1982. Least squares quantization in PCM. *IEEE Trans. on Information Theory* 28(2).

Malpani, A.; Ravindran, B.; and Murthy, H. 2011. Personalized intelligent tutoring system using reinforcement learning. In *Int. FLAIRS Conf.*

Mauss, I. B., and Robinson, M. D. 2012. Measures of emotion: A review. *Cognitional and Emotion* 23(2).

Mondragon, A.; Dufresne, A.; Nkambou, R.; and Poirier, P. 2017. An affective intelligent tutoring system in the special education of individuals with autism. In *Int. Conf. on Education and New Learning Tech.*

Murray, T., and Arroyo, I. 2002. Toward measuring and maintaining the zone of proximal development in adaptive instructional systems. In *Int. Conf. on Intelligent Tutoring Systems*.

Nagai, Y.; Moriwaki, T.; and Asada, M. 2015. Influence of excitation/inhibition imbalance on local processing bias in autism spectrum disorder. In *Annual Meeting of the Cognitive Science Society*.

Sahin, N. T.; Keshav, N. U.; Salisbury, J. P.; and A., V. 2018. Second version of google glass as a wearable socio-affective aid: Positive school desirability, high usability, and theoretical framework in a sample of children with autism. *Jour. of Medical Internet Research Human Factors* 5(1).

Sarma, B. H. S., and Ravindran, B. 2007. Intelligent tutoring systems using reinforcement learning to teach autistic students. *Home Informatics and Telematics: ICT for The Next Billion* 241.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv:1707.06347*.

Theocharous, G.; Beckwith, R.; Bukto, N.; and Philipose, M. 2009. Tractable POMDP planning algorithm for optimal teaching in SPAIS. In *Int. Joint Conf. on Artificial Intelligence Workshop on Plan, Activity, and Intent Recognition*.

Wang, F. 2014. POMDP framework for building an intelligent tutoring system. In *Int. Conf. on Computer Supported Education*.

Wang, F. 2018. Reinforcement learning in a POMDP based intelligent tutoring system for optimizing teaching strategies. *Int. Jour. of Information and Education Tech.* 8(8).

Winoto, P.; Chen, J.; Guo, H.; and Tang, T. Y. 2018. A mathematical and cognitive training application for children with autism: A system prototype. In *Int. Conf. on Human Computer Interaction*.