Towards Generalizable Agents in Text-Based Educational Environments: A Study of Integrating RL with LLMs

Bahar Radmehr
EPFL
bahar.radmehr@epfl.ch

Adish Singla
MPI-SWS
adishs@mpi-sws.org

Tanja Käser
EPFL
tanja.kaeser@epfl.ch

ABSTRACT

There has been a growing interest in developing learner models to enhance learning and teaching experiences in educational environments. However, existing works have primarily focused on structured environments relying on meticulously crafted representations of tasks, thereby limiting the agent's ability to generalize skills across tasks. In this paper, we aim to enhance the generalization capabilities of agents in open-ended text-based learning environments by integrating Reinforcement Learning (RL) with Large Language Models (LLMs). We investigate three types of agents: (i) RL-based agents that utilize natural language for state and action representations to find the best interaction strategy, (ii) LLM-based agents that leverage the model's general knowledge and reasoning through prompting, and (iii) hybrid LLM-assisted RL agents that combine these two strategies to improve agents' performance and generalization. To support the development and evaluation of these agents, we introduce `PharmaSimText`, a novel benchmark derived from the PharmaSim virtual pharmacy environment designed for practicing diagnostic conversations. Our results show that RL-based agents excel in task completion but lack in asking quality diagnostic questions. In contrast, LLM-based agents perform better in asking diagnostic questions but fall short of completing the task. Finally, hybrid LLM-assisted RL agents enable us to overcome these limitations, highlighting the potential of combining RL and LLMs to develop high-performing agents for open-ended learning environments.

Keywords

Reinforcement Learning, Large Language Models, Text-Based Educational Environments, Learner Models

1. INTRODUCTION

Learner models are foundational to the advancement of educational technologies, serving as a versatile tool for a multitude of applications that enhance both teaching and learning experiences [1]. By simulating the interactions and data of students, these computational models provide a safe and controlled environment for teacher training, allowing educators to refine their methods without direct implications on actual students [2]. They also facilitate the development and evaluation of adaptive learning systems [3] or new algorithms [4]. Furthermore, they have been applied for testing theories of learning [5] and foster collaboration skills in students through interacting with virtual peers [6].

Reinforcement learning (RL) offers a promising avenue for developing these learner models/agents [7]. Existing works on RL for educational domains have primarily focused on developing techniques for curriculum optimization [8–11], providing tailored hints and feedback [12, 13], and generating educational content [14, 15]. Only a limited number of works have explored the use of RL-based learner agents that effectively operate in the learning environments [16, 17]. However, these RL-based learner agents have been studied for structured tasks with well-defined rules, such as mathematics and logic puzzles. In such environments, RL's capabilities are naturally exploited due to the straightforward definition of state and action representations using engineered features obtained from the existing structure [7, 16, 18]. However, the reliance on hand-crafted features and engineered state representations limits the ability of these RL agents to be used in unstructured domains and to generalize their learned skills and knowledge across different tasks.

Recent advances in generative AI, in particular Large Language Models (LLMs), provide new opportunities to drastically improve state-of-the-art educational technology [19]. LLMs are capable of generating coherent and contextually relevant content, engaging in meaningful dialogues, and executing specific linguistic tasks without explicit training [20, 21]. So far, in education, LLMs have mainly been *applied* for generating educational content [22–24], automating grading and feedback processes [25–30], and facilitating the development of collaborative systems [31–33]. Few works have also used LLMs for learner modeling in programming domains [34] or for simulating students' behaviors as a basis for an interactive tool for teacher training [35]. However, despite their proficiency in linguistic tasks, LLMs often fall short in decision-making in a constrained environment, a domain where RL agents excel due to their inherent capability to make feasible decisions within a given environment [36].

Given the strengths and limitations of RL and LLM-based agents, recent works have investigated the integration of

LLMs with RL to design agents that overcome the individual limitations of these agents. For instance, this integration has been used to substantially improve reward design and exploration efficiency in various domains [37–40]. However, most of these approaches have focused on the use of LLMs for training, bearing the risk of taking on LLMs' limitations in decision-making in constrained environments.

In this paper, we investigate the integration of RL and LLMs to create agents with enhanced generalizability in text-based educational environments, focusing on employing the LLM in the inference phase. To support our investigations, we present a novel text-based simulation benchmark, `PharmaSimText`, adapted from the PharmaSim virtual pharmacy environment designed for practicing diagnostic conversations. We present three types of agents: (i) RL-based agents employing natural language based representations, (ii) LLM-based agents invoked through prompting, and (iii) hybrid models where LLMs assist RL agents in the inference phase.

We extensively evaluate all agents based on their ability to engage in effective diagnostic conversations and achieve accurate diagnoses on the `PharmaSimText` benchmark, focusing on their performance across a range of rephrased scenarios across diverse patient profiles. With our experiments, we aim to address three research questions: Which agent type demonstrates overall superior performance in conducting effective diagnostic conversations and achieving accurate diagnoses for all available patients **(RQ1)**? How does reflective prompting influence the diagnostic performance and conversation quality of LLM-involved agents **(RQ2)**? How do diagnostic performance and conversation quality vary among different agent types across diverse patients **(RQ3)**? Our results demonstrate that a specific type of LLM-assisted RL agent outperforms all other agents in a combined score by effectively balancing accurate diagnosis along with high-quality diagnostic conversations. The source code and benchmark are released on GitHub.[1]

2. RELATED WORK

Given our focus on integrating RL agents and LLMs to create generalizable learner models, we review prior work in developing learner models, explore the growing field of intelligent agents in text-based interactive games and finally discuss recent advancements in integrating RL and LLMs.

Learner agents in educational environments. There is a large body of research [1] on simulating learners in online environments. Existing research provides rich, but not generalizable learner representations, for example by generating cognitive models from problem-solving demonstrations (e.g., SimStudent [41]) or simulates learners from student models in a data-driven way [42–44], leading to less rich, but more generalizable representations. RL is a promising tool to address these limitations. However, in the education domain, this framework has been primarily applied for pedagogical policy induction [8–11], providing tailored hints [12, 13], generating educational content [14, 15], and assessing interventions in educational platforms [45, 46]. Despite its potential, the exploration of RL-based learner agents for effective operation in learning environments remains limited [16, 17]. Prior work has for example used Proximal Policy Optimization (PPO) for designing learner models in intelligent tutoring systems [16] or employed neural and symbolic program synthesize to create student attempts in a block-based programming environment [47]. In this paper, we develop a series of learner agents for an open-ended educational environment.

Agents for text-based interactive games. The growing interest in developing intelligent agents for text-based interactive games, especially those that mimic real-world scenarios [36, 48, 49], has led to diverse methodologies encompassing RL [50], behavior cloning (BC) [36], and prompting LLMs [51, 52]. A well-known example is the game ScienceWorld [36], where players engage in scientific experiments through environment exploration and interaction. Within the RL framework, state-of-the art employs deep reinforced relevance networks (DRRNs) [50], treating text-based interactions as partially-observable Markov decision processes (POMDPs), and learning distinct text representations for observations and actions to estimate Q-values via a scorer network. Within the LLM domain, LLM-based strategies use prompts at each interaction step for strategic planning and action selection. While some studies [51] engage in a single interaction round with the environment, others [52, 53] use a multi-round approach, facilitating iterative refinement through repeated attempts. In this paper, we develop a series of agents for a text-based educational environment simulating real-world scenarios happening in a pharmacy.

RL and LLM integration. Recently, LLMs have been used to assist RL agents in various tasks, demonstrating notable advancements in reward design and exploration efficiency. For example, [39] utilized text corpora to pre-train agents, thereby shaping their exploration by suggesting goals based on the agents' current state descriptions. Furthermore, [40] proposed a novel approach to simplify reward design by employing LLMs to generate reward signals from textual prompts that describe desired behaviors. In a similar vein, [37] showcased the innovative application of few-shot LLM prompting to hypothesize world models for RL agents, which improves training sample efficiency and allows agents to correct LLM errors through interaction with the environment. While these studies highlight the synergistic potential of integrating LLMs with RL techniques to achieve more objective-aligned agent behaviors, directed exploration, and efficient training processes, the use of LLMs in the training phase bears the risk of carrying over their limitations in decision-making in constrained environments. A notable gap, therefore, remains in using LLMs to assist RL agents during the inference phase. Specifically, the current body of work has not addressed the use of LLMs to aid RL agents in adapting and transferring their learned skills to novel environments or tasks post-training. In our work, we aim to bridge this gap by focusing on utilizing LLMs as assistants for RL agents during generalization to new settings.

3. PHARMASIMTEXT BENCHMARK

We created `PharmaSimText`, a text-based interactive environment, as an infrastructure for developing language agents capable of handling text-based learning tasks and generalizing in them. `PharmaSimText` is an interactive text-based environment designed based on PharmaSim, a scenario-based learning platform. It simulates real-world interactions be-

[1] https://github.com/epfl-ml4ed/PharmaSimText-LLM-RL

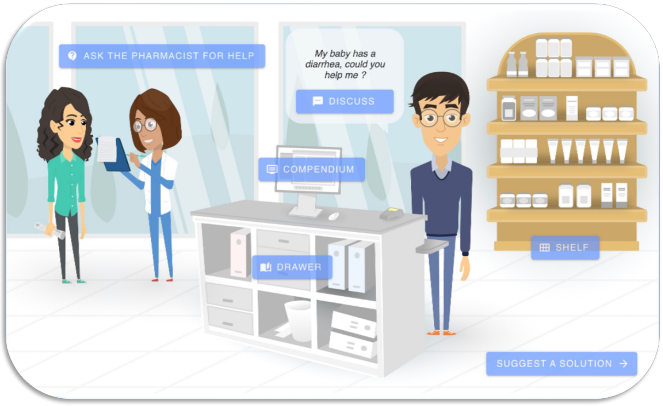

Figure 1: **'Father Inquiry' Scenario in PharmaSim** - A simulated pharmacy setting designed for practicing diagnostic conversational skills, where participants engage with a father seeking guidance for his infant child's diarrhea.

tween a pharmacist and a patient in a pharmacy setting. This benchmark includes more than 500 scenario variations that can be used for developing and evaluating learner agents.

3.1 PharmaSim

PharmaSim is a scenario-based learning environment designed to support the development of diagnostic skills. In each scenario, a patient comes to the pharmacy and asks for help with a specific problem. The player needs to identify different possible causes of this problem and mark how probable they are while interacting with the environment. Specifically, there are six different types of interactions: asking questions to the patient, seeking help from the pharmacist, searching about different kinds of medicine, looking for the specifications of products available on the shelf, reading about issues related to the problem, and offering a solution, which ends the game and moves the player to the post-test phase. In the post-test phase, players need to list three possible causes, rate their probability, and give an explanation for each of them. The determination of these likelihoods that leads to finding the most probable cause significantly depends on a set of patient inquiries containing essential information, which we henceforth refer to as *key questions*.

Currently, two different scenarios designed with insights from human experts are available in the game. For example, in one scenario (see Fig. 1), a father visits the pharmacy looking for help with his infant child's diarrhea. The scenario presents four probable causes for the child's condition. The player is required to ask four *key questions* to the father to gather the essential information needed to find the most probable cause behind the child's diarrhea. The relation between these *key questions* and the most probable cause of the child's diarrhea is illustrated in Fig. 2. For instance, inquiring about the child's age enables the player to deduce that teething is an improbable cause due to the child's young age.

3.2 PharmaSimText

To develop our benchmark, several modifications to PharmaSim were implemented.

Migration to a text-based environment. As the first step, we did two adaptions to PharmaSim to migrate it to a text-based environment. First, we simplified interactions to two types of actions: asking questions to the patient about various characters phrased similar to PharmaSim as "`I want to know about the character's topic.`" and advancing to the post-test by proposing a solution as "`I want to suggest a solution.`". Second, we modified the post-test questions to offer a feasible assessment for the agents. To this end, we revised the three causes question to focus solely on the most probable cause of the patient's issue.

Extension of available scenarios. In the next step, we focused on enriching `PharmaSimText` and enhancing its complexity. For this purpose, we expanded the two scenarios available in the original environment across three dimensions: (1) introducing new patients, (2) varying the scenarios to alternate the most probable cause of each patient's problem, and (3) diversifying patient responses by rephrasing them. Given the scale of extension, relying solely on human expertise was impractical. Instead, we leveraged the generative capabilities of LLMs combined with human insights to develop the scenarios in `PharmaSimText`. Prior to prompting LLMs for creating scenarios, we structured our expanded scenarios to align with the pharmacy assistant training curriculum of Switzerland. We gathered a set of health problems from the curriculum, assigning each to a fictional patient with a specified age and gender. We further identified a range of illnesses from the curriculum's textbooks, known to manifest symptoms relevant to the chosen problems.

Prompting LLMs for scenario creation. The scenario creation process involved three steps: (1) we prompted the LLM to generate a list of *key questions* aimed at diagnosing the most probable cause of the patient's problem, (2) the LLM was tasked to simulate patient responses, assuming each illness on the list was the most probable cause behind their problem, and (3) the LLM was prompted to generate answers to common patient inquiries done by pharmacists. We used GPT-4 as the LLM for scenario creation; the exact prompts employed can be found on our GitHub repository (link provided in Footnote 1). To ensure realism and applicability, a human expert has reviewed all of the scenarios and provided feedback including minor changes which were reflected in the final version of the scenarios. Additionally, the LLM was employed to diversify existing patient responses through paraphrasing, enhancing the scenarios' complexity. To further augment this complexity, fictional characters were introduced as distractors, enabling players to engage in more nuanced interactions.

Statistics on the PharmaSimText benchmark. The obtained benchmark contains eight distinct scenarios, each revolving around a unique patient profile. Details about the patients can be found in Table 1. On average, each scenario presents seven potential causes for the patient's problem, resulting in a total of 47 scenario variations. Patient responses in each variation are articulated in ten diverse phrasings to enhance the depth and variability. Furthermore, each scenario necessitates the identification of an average of 7.8 *key questions* by the player. As a result, `PharmaSimText` can provide an enriched environment for further studies on agents for text-based interactive tasks and agents' generalizability.

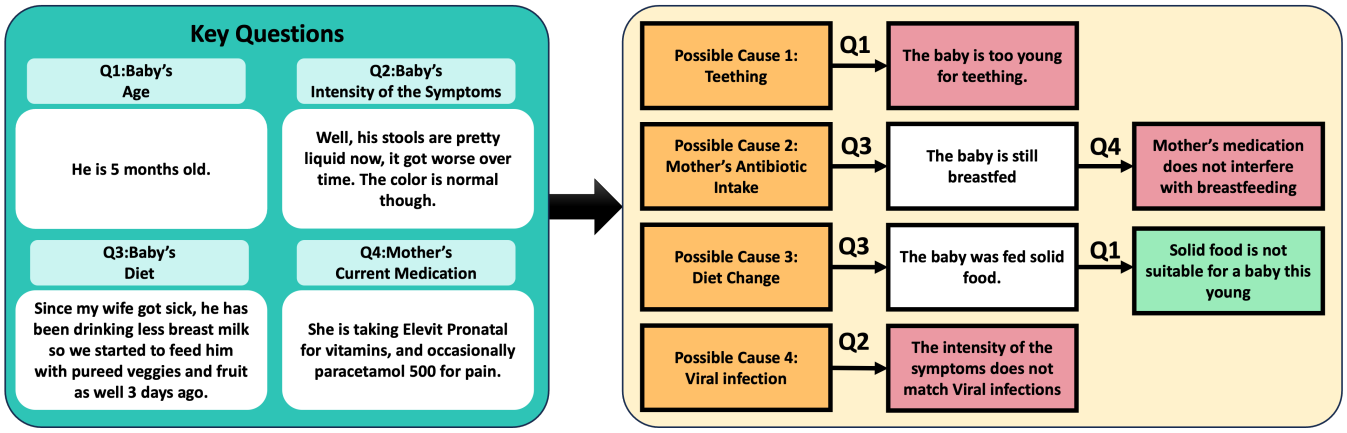

Figure 2: **Diagnostic Strategy in the 'Father Inquiry' Scenario of PharmaSim**, depicting the process of identifying the most likely cause of an infant's diarrhea. Players must pose four key questions to the father to collect crucial information, enabling the determination of the most probable cause of the child's diarrhea among four potential causes.

Problem	# of Possible Causes	Possible Causes	# of Key Questions
Infant Diarrhea	4	Change of diet, Teething, Current medication of the mother, Viral Infection	4
Breastfeeding-related	6	Engorgement, Plugged Ducts, Cracked Nipples, Mastitis, Thrush, Low Milk Supply	7
Urological	4	Prostate Hyperplasia, Cystitis, Urge Incontinence, Stress Incontinence	6
Skin-related	10	Sunburn, Insect Bites, Acne, Eczema, Athlete's Foot, Psoriasis, Rashes, Warts and Corns, Cold Sores, Neurodermatitis	10
Eye-related	5	Dry Eyes, Allergic Conjunctivitis, Pink Eye, Eye Strain, Stye	11
Gynecological	8	UTI, Cystitis, Kidney Stones, Overactive Bladder, Pregnancy, STI, Stress Incontinence, Fungal Infection	8
Joint Pain	5	Osteoarthritis, Muscle Sprains, Tendonitis, Bursitis, Gout	9
Sore Throat	5	Common Cold, Influenza, Sinusitis, Pharyngitis, Bronchitis	7

Table 1: **Overview of `PharmaSimText` Scenarios.** Every task within the benchmark is centered on a unique health problem, which could stem from various causes. Players must ask several *key questions* to arrive at a correct diagnosis.

4. AGENTS FOR PHARMASIMTEXT

We developed three types of agents for `PharmaSimText` that embody various degrees of RL and LLM synergy: *RL-based* agents, *LLM-based* agents, and *LLM-assisted RL* agents.

4.1 RL-based Agents

RL agents learn to interact within an environment by taking actions based on their current state and receiving feedback in the form of rewards or penalties for those actions [54]. They try to maximize their obtained cumulative reward over time to effectively learn the best policy for achieving their goal within the environment. One well-known method in RL involves estimating a metric called Q-value, which represents the expected future rewards for taking a certain action in a given state. Deep Q-Networks (DQNs)[55] approximate these Q-values using deep neural networks, enabling handling of complex, high-dimensional environments by learning to predict the Q-values directly from the environmental states. DQNs are trained through interactions with the environment, using their experience to iteratively refine and make their estimations of Q-values more accurate.

Following previous work on text-based games, we utilized state-of-the-art, a DRRN [50] as the *RL-based* agent for interacting with `PharmaSimText`. The DRRN is designed to learn distinct representations for the text-based states and actions by employing two separate networks: the state encoder and the action encoder. A scorer network then evaluates these representations to estimate their Q-values. At a given step t in the environment, the current state s_t and the action taken a_t are fed into the DRRN. Initially, s_t and a_t are encoded as sequences of word embeddings, which are subsequently processed by a Recurrent Neural Network (RNN) within both the state and action encoders to obtain respective embeddings for s_t and a_t. Following the RNN layer, a Multi-Layer Perceptron (MLP) in each encoder refines these embeddings into more concise representations. These representations are then concatenated and fed into the scorer network's MLP, which yields an estimation of the Q-value $Q(s_t, a_t)$.

In our case, the valid actions at time step t are interactions available in the environment presented to the agent as a list of sentences. After taking each action, the agent will receive

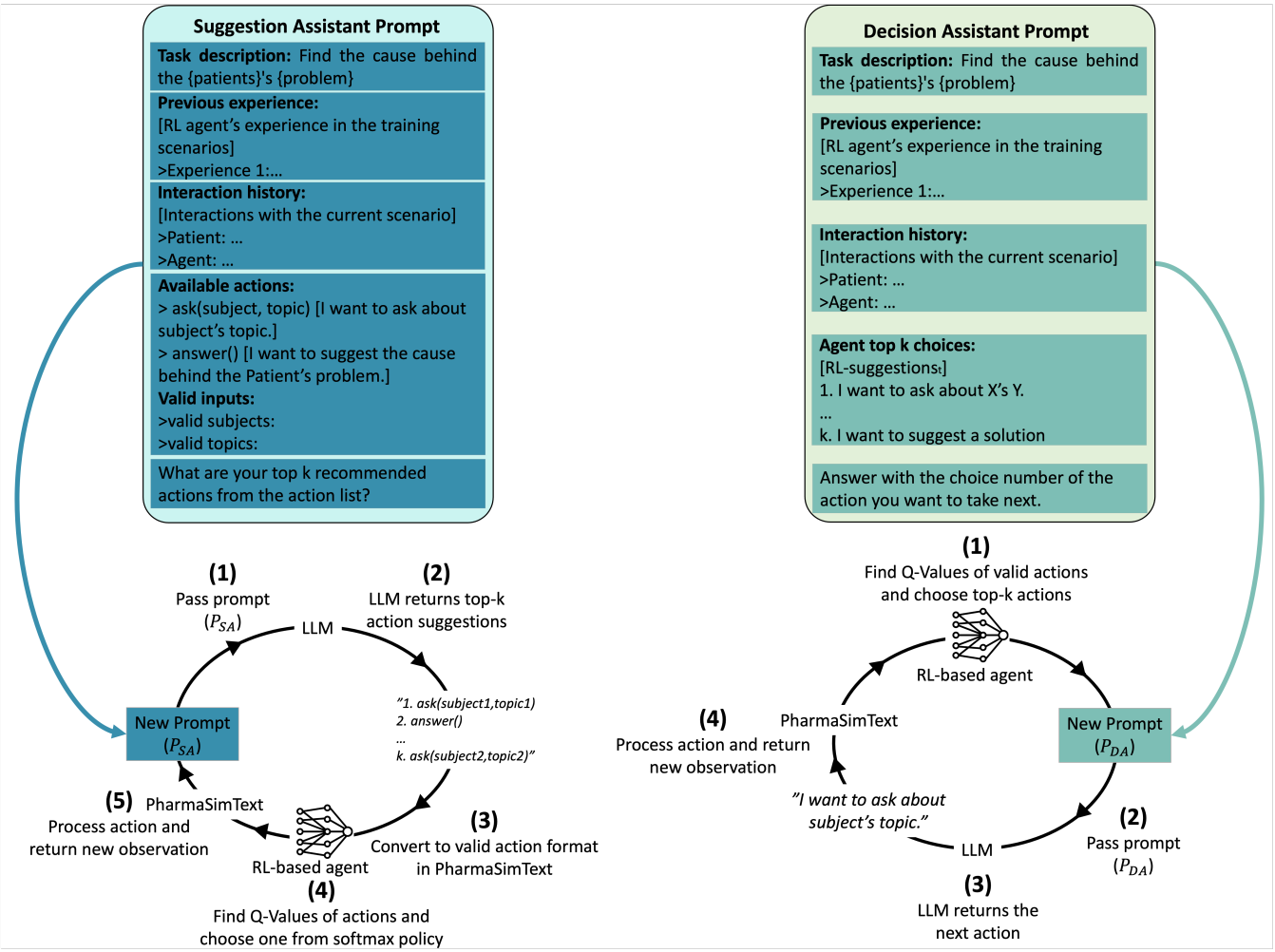

Figure 3: **LLM-assisted RL agents.** An LLM is prompted to assist the RL agent at the inference time to aid in generalization. In the Suggestion-Assisted RL (SA-RL) agent (left), the LLM suggests k actions at each step for the RL agent to choose from. In the Decision-Assisted RL (DA-RL) agent (right), the LLM selects an action from the top-k choices provided by the RL agent.

a new observation o_t that is formatted as: `Interaction type; Selected interaction; The patient's response`. For instance, in the scenario related to infant diarrhea if the agent decides to ask about the infant's age, the new observation will be formatted as: `Discuss; I want to know about the infant's age; He is 5 months old`. Therefore, the agent should consider the full history of its observations to comprehend its current state s_t in the environment.

We introduced two modifications to adapt the original DRRN to our environment. First, we employed pre-trained sentence embeddings from fastText [56] to generate text representations for both observations and actions. This choice was motivated by previous work showing that training the RNNs in the encoders of a DRRN with a loss function solely aligned with the RL objectives leads to unstable training and suboptimal embeddings [57]. Second, unlike the environments that DRRNs were proposed to tackle the tasks in, the observation at a given time step t in `PharmaSimText` does not suffice for the agent to obtain a notion of the current state in the environment and the whole full observation history is needed as a part of context given to the agent. Therefore, we introduced a unit called the **state updater** before the state encoder that takes the previous embedded state $e(s_{t-1})$ and the new embedded observation $e(o_t)$ and returns the updated state after the current observation s_t. We experimented with five different methods in the state updater: mean pooling, max pooling, summation, an LSTM layer, and an LSTM layer with self attention. After a series of experiments, we observed the method based on summation led to the most stable training; therefore this method was adopted in our state updater. Formally, this method based on the summation of all the observation embeddings in the history, returns $e(s_t) = e(s_{t-1}) + e(o_t)$ as the new embedded state $e(s_t)$.

4.2 LLM-based Agents

The agents based on LLMs prompt an LLM at each step of interacting with the environment to find the best next action to finish the task. These agents can either have only one trial or multiple trials to complete the task along with reflection on their strategy between each trial. We respectively denote these two agent types by *none-reflective* and *reflective*.

The *none-reflective agent* interacts with the LLM by issuing a single prompt that contains the task description, the history of interactions (consisting of the agent's questions and the patient's responses), prior experience with the patient, and valid actions available at the current step to choose the most appropriate subsequent action. The task description is structured as `Find the cause behind the patient's problem`, while the interaction history is presented as a dialogue between the patient and the agent, with action texts labeled as agent's questions and environment's feedback text as patient responses. To format the valid actions, each action type is formatted as a function along with its permissible input values, which the LLM can interpret. This is complemented by a descriptive text explaining the action's purpose. For instance, the interaction "I want to ask about the subject's topic" is formatted as `ask(subject, topic): Asking a question about the subject related to the topic`, followed by a list of valid subjects and topics. This meticulous formatting strategy plays an essential role in minimizing the likelihood of the LLM suggesting invalid actions.

Despite efforts to format valid actions to guide the LLM, there are instances where the LLM still proposes an action that is invalid within the `PharmaSimText` environment. In such cases, we implemented a strategy where the LLM was prompted to suggest an alternative action, repeating this process for a maximum of $k = 3$ attempts. Should all suggested actions remain invalid, we selected the valid action that has the smallest distance in the natural language embedding space to the k-th suggested action. This approach ensures that the LLM's output is effectively grounded in the set of actions that are feasible within the environment.

The *reflective agent* employs a prompting strategy akin to that of the *none-reflective agent* to determine the optimal subsequent action. The *none-reflective agent* prompt is augmented with a segment including learnings from prior engagements with the same patient having the same cause. This reflective process involves prompting the LLM to evaluate its previous strategies based on the observed outcomes after completing each trial. The agent then updates its textual memory of previous learnings, and the updated memory is used for prompting in the next trial. This approach was inspired by research on self-reflective LLMs, notably the continually learning language agent CLIN[52]. Similar to CLIN, we constructed the learning memory using causal formats such as "X is necessary for Y" to guide future interactions. This mechanism enables the reflective agent to dynamically adapt and refine its approach, enhancing its decision-making process over time.

4.3 LLM-assisted RL Agents

The perspective of *RL-based* agents remains limited to their experience during training, potentially hindering the performance in tasks with unfamiliar elements not encountered during their training. To address this, we leveraged LLMs' commonsense reasoning capabilities to augment RL agents' decision-making processes. As shown in Fig. 3, we explored two methods for integrating LLM assistance: *Suggestion-Assisted RL (SA-RL)* and *Decision-Assisted RL (DA-RL)*.

In the *SA-RL* approach, at a given time step t, the LLM is prompted to suggest a list of k best actions to be taken at that state called LLM-Suggested$_t$. The actions' Q-values in LLM-Suggested$_t$ are then calculated by the RL agent, and the next action is sampled from the probability distribution obtained by taking softmax over the estimated Q-values. The prompting format here is similar to the *LLM-based* agents discussed in Section 4.2 containing the task description, the history of interactions, prior experience with the patient, and valid actions at that step. We set $k = 5$ in the interaction steps and $k = 2$ in the posttest steps.

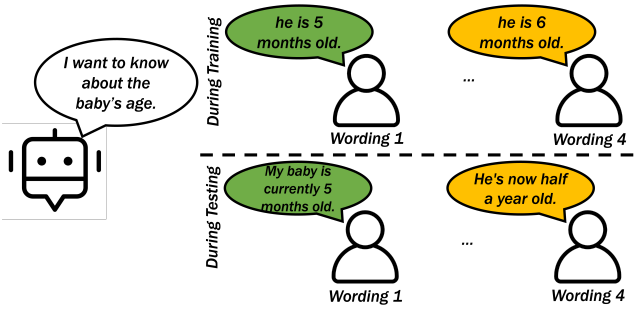

Figure 4: **Generalization task**, requiring the agents to generalize over different wordings of a scenario.

In the *DA-RL* approach, at a given time step t, we collect a list of k most probable actions under the RL agent's policy RL-Suggested$_t$. Then, an LLM is prompted to choose the best action among the actions in RL-Suggested$_t$. The prompting used for this task contains the task description, the history of interactions, prior experience with the patient, and the actions in RL-Suggested$_t$. Therefore, the LLM acts as a decision assistant for the RL agent. Notably, in our implementation, we set $k = 5$ in the interaction steps and $k = 2$ in the post-test steps.

Based on whether the LLM is given an opportunity to reflect on its past decisions or not, we obtain two versions of *DA-RL* and *SA-RL* approaches, which we distinguish via reflective/none-reflective prefixes. Thus, we study four *LLM-assisted RL* agents: *none-reflective-DA-RL*, *reflective-DA-RL*, *none-reflective-SA-RL*, and *reflective-SA-RL*.

5. EXPERIMENTAL EVALUATION

We evaluated our agents in `PharmaSimText` to assess which agent type demonstrates the most effective diagnostic conversations and accurate diagnoses among all patients **(RQ1)**, to investigate the impact of reflective prompting on the diagnostic performance and interaction quality of LLM-involved agents **(RQ2)**, and to explore how diagnostic performance and conversation quality vary among the different agent types when confronted with different patients **(RQ3)**.

5.1 Experimental Setup

Our evaluation was focused on the generalization capabilities of the agents, specifically their ability to navigate tasks featuring not previously encountered elements. We assessed the agents' generalizability across rephrased versions of already-encountered scenarios, aiming to measure their reliance on the precise wording of these scenarios. Figure 4 provides insight into our evaluation methodology for generalization,

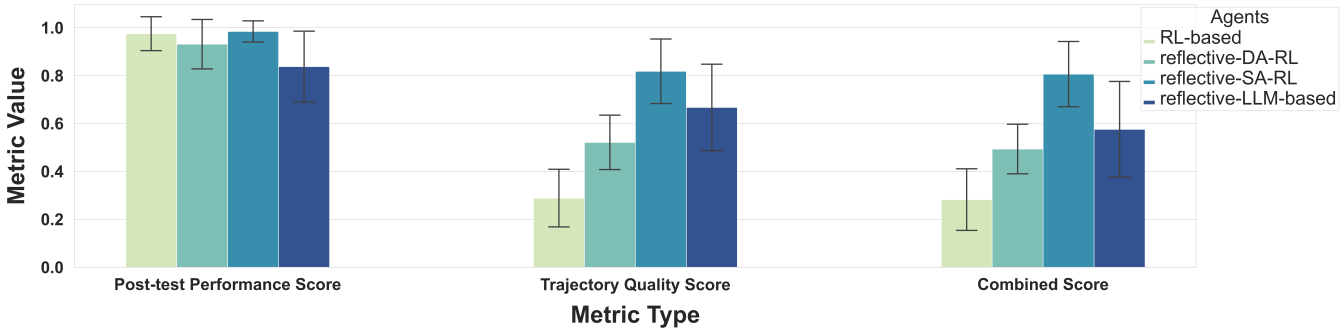

Figure 5: **Agent Performance on PharmaSimText.** *Post-test Performance Score* (left), *Trajectory Quality Score* (middle), and *Combined Score* (right) of the *RL-based* agent, the reflective-*DA-RL* agent, the reflective-*SA-RL* agent, and the reflective-*LLM-based* agent. In the *SA-RL* agent, the LLM suggests k actions at each step for the RL agent to choose from. In the *DA-RL* agent, the LLM selects an action from the top-k choices provided by the RL agent. Scores are averaged across all patients in `PharmaSimText`.

illustrating the diversity created by rephrased answer options in a specific scenario.

We defined agent success in a subtask based on two aspects: identifying the most probable cause of the patient's problem and asking the *key questions* in the conversation. Here a subtask denotes the combination of a cause and a wording. We therefore introduced three metrics:

- *Post-test Performance Score*: binary indicator of correct diagnosis of the patient's problem. It measures the agent's ability to identify the most probable cause of the patient's problem.
- *Trajectory Quality Score*: fraction of key questions involved in the agent's conversation. It measures the quality of the agent's conversation.
- *Combined Score*: product of the *Post-test Performance Score* and *Trajectory Quality Score*. It measures both the above elements together.

5.2 Agent Training and Evaluation

We developed and trained all of the agents separately for each patient. In this process, different wordings of subtasks leading to the same cause were split randomly to a training, validation, and test set. Therefore, the training, validation, and test sets included subtasks of all of the causes available for a patient in distinct wordings. Specifically, the agents saw all the causes during training and validation, but not all wordings. In our experiments, 80% of the available wordings for each cause were used for training and the remaining wordings were split in half for the validation and test set.

The *RL-based* agents were trained using subtasks from the designated training set being given a random subtask at each episode of interaction with the environment. At a given time step t, the agent took an action sampled from a softmax policy obtained from the Q-values of all of the actions available. The randomness of the softmax policy was controlled using a temperature decaying from 1 to 0.001 linearly during the training. After each interaction, the agent was rewarded using a reward function that awarded the agent a positive reward of +1 when it successfully completed the posttest and penalizes with -1 otherwise. Moreover, each interaction of the agent was penalized by a small negative reward of -0.01.

Following each iteration of training, these agents underwent an evaluation phase using subtasks from the validation set. The iteration that yielded the highest average *Post-test Performance Score* on the subtasks in validation set was used for testing and also served as the foundation for the RL component within the *LLM-assisted RL* agents.

The agents that had an LLM involved in their structures used the GPT-4 model. The *LLM-based* agents initially gain experience through interactions within the training subtasks. This acquired experience is subsequently leveraged during their engagement with the test subtasks.

5.3 RQ1: Efficacy of Different Agent Types

In our first analysis, we aimed to assess the agents' efficacy in diagnostic dialogues and accuracy in diagnoses aggregated over all patients. Figure 5 illustrates the *Post-test Performance Score*, *Trajectory Quality Score*, and *Combined Score* of the different agents.

We observed that the *RL-based* agent achieved a high *Post-test Performance Score*, indicating its ability to arrive at the correct diagnosis through a process of trial and error. However, this agent's approach often lacked the depth and nuance of a meaningful diagnostic conversation, reflected in its low *Trajectory Quality Score*. This observation is probably due to its lack of background knowledge and common sense reasoning. Conversely, the *LLM-based* agent exhibited a superior capacity for engaging in meaningful diagnostic dialogues, reflected in a higher *Trajectory Quality Score*. However, the *LLM-based* agent exhibited a lower *Post-test Performance Score* than the *RL-based* agent, indicating that its ability to consistently reach the correct diagnosis is inferior compared to the *RL-based* agent.

In examining the *LLM-assisted RL* agents, both *DA-RL* and *SA-RL* agents surpassed the *LLM-based* agent in *Post-test Performance Score*, indicating that integrating LLM with RL generally improves diagnostic precision of purely *LLM-*

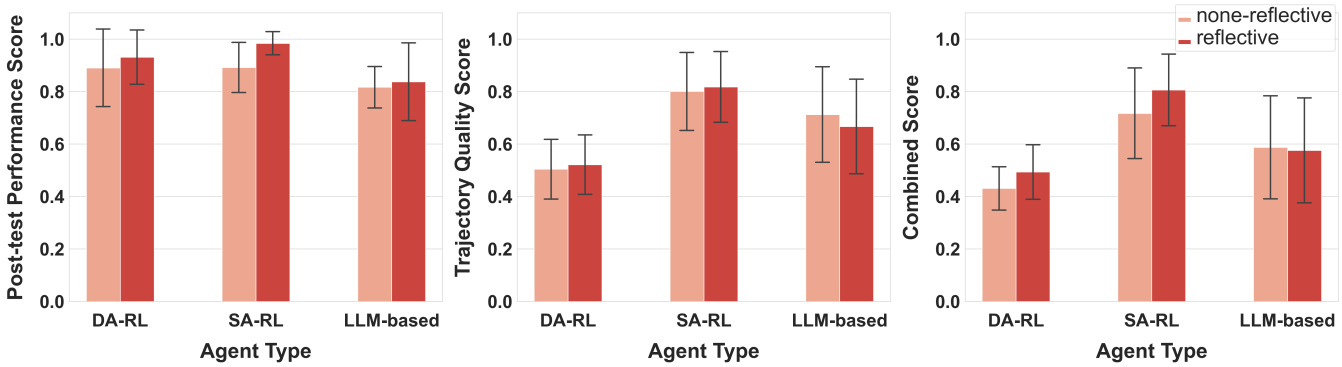

Figure 6: **Performance of reflective and none-reflective agents on PharmaSimText.** *Post-test Performance Score* (left), *Trajectory Quality Score* (middle), and *Combined Score* (right) for none-reflective and reflective *DA-RL*, *SA-RL*, and *LLM-based* agents.

based agents. Notably, the *SA-RL* agent exhibited superior *Post-test Performance Score* closely mirroring that of the *RL-based* agent. The *DA-RL*'s relative under-performance may have stemmed from its longer trajectories compared to the *RL-based* agent, leading to unfamiliar states where the DRRN struggled to provide accurate diagnoses, thereby affecting the *DA-RL*'s RL-driven suggestions. Furthermore, in terms of engaging in quality diagnostic dialogues, the *SA-RL* agent was also superior to the *DA-RL* agent. This superiority is likely due to the RL framework's preference for shorter, more direct solutions, which reduced the action quality suggested by the DRRN in prolonged interactions. This effect was more pronounced in the *DA-RL* agent, potentially constraining the quality of diagnostic conversations.

In the comparison of the agents in the *Combined Score*, the *SA-RL* agent emerged as the standout performer. Unlike its counterparts, the *SA-RL* agent adeptly navigated the dual challenges posed by the benchmark, demonstrating both a high conversation quality and diagnostic accuracy. This achievement highlights the *SA-RL* agent's unique capacity to capture the strengths of both *RL-based* and *LLM-based* agents through the addition of suggestion-based assistance from LLMs to the RL agents' decision-making process.

To further investigate the results, we performed additional statistical tests. A Kruskal-Wallis test shows significant differences between the agents for the *Trajectory Quality Score* and *Combined Score* ($p_{trajectory} < 0.0001$ and $p_{combined} < 0.001$) and a trend to significance for the *Post-test Performance Score* ($p_{performance} = 0.052$). Post-hoc comparisons using Mann-Whitney U tests with a Benjamini-Hochberg correction for the *Combined Score* indicate significant differences between 5 out of 6 pairs of agents supporting prior findings. For instance, the comparison between *RL-based* agent and *SA-RL* agent resulted in a p-value smaller than 0.01, and for the comparison between *SA-RL* agent and *LLM-based* agent the p-value was smaller than 0.05.

In summary, the experimental outcomes highlight distinct strengths and weaknesses among the agents. The RL-based agent demonstrated proficiency in achieving a high Post-test Performance Score score, but was hindered in engaging in effective diagnostic dialogues due to limited background knowledge. Conversely, the LLM-based agent excelled in conducting high-quality conversations by leveraging its extensive knowledge base, though with less accuracy in diagnoses. The hybrid LLM-assisted RL agents, DA-RL and SA-RL, outperformed the LLM-based agent in diagnostic precision and surpassed the RL-based agent in dialogue quality. The SA-RL agent achieved both a high conversation quality and diagnostic accuracy, illustrating its effective integration of LLM and RL capabilities.

5.4 RQ2: Effect of Reflective Prompting

In our second analysis, we aimed to explore the impact of reflective prompting on the efficacy of LLM-involved agents. As described in Section 4, none-reflective agents were limited to a single attempt, whereas reflective agents were given three attempts per subtask with opportunities for reflection. Figure 6 illustrates the *Post-test Performance Score*, *Trajectory Quality Score*, and *Combined Score* for none-reflective and reflective *LLM-assisted RL* and *LLM-based* agents.

We observed a nuanced impact of reflective prompting on agent performance. Specifically, reflective prompting did not significantly impact the *Combined Score* of the purely *LLM-based* agent. For this agent, reflection led to shorter diagnostic conversations by eliminating what the agent considered redundant questions. However, this streamlining resulted in poorer conversation quality without significantly improving diagnosis accuracy, negating the potential diagnosis accuracy gains from reflection.

In contrast, the reflective process considerably enhanced the performance of the hybrid *LLM-assisted RL* agents. This improvement can be attributed to the reflective phase allowing the agents to reassess and refine their decision-making processes, leading to more accurate diagnoses. The performance boost was particularly notable in *SA-RL* agents, most likely due to their reliance on the LLM for suggesting potential actions during the interaction phase. This reliance provided a broader scope for reflection to influence decision-making, unlike *DA-RL* agents where decisions were more heavily influenced by the *RL-based* agent. This finding underscores the value of incorporating reflective mechanisms in enhancing the capabilities of hybrid agents.

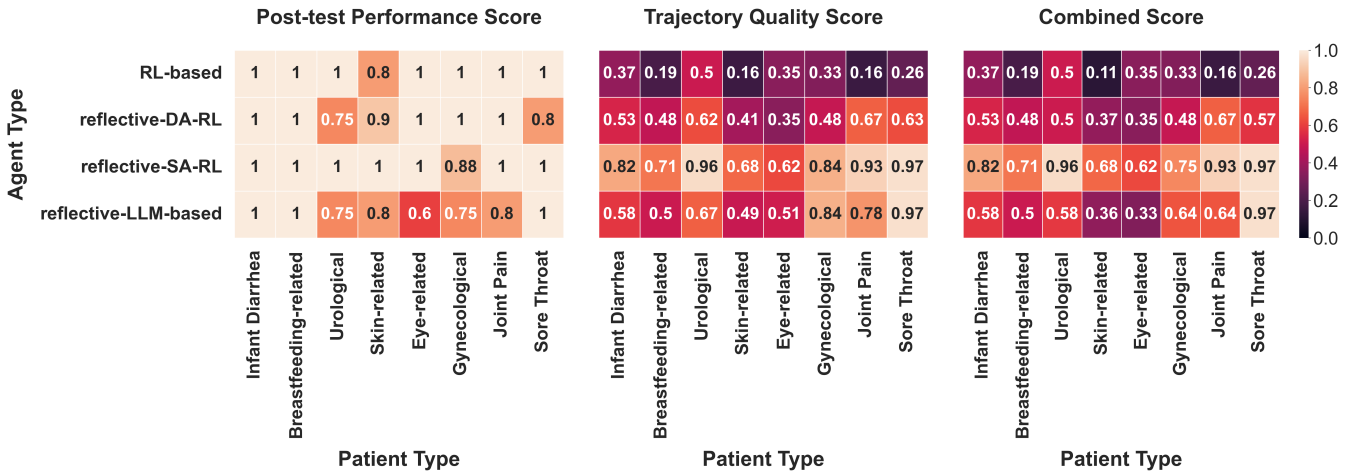

Figure 7: **Performance of different agents in interaction with different patients.** *Post-test Performance Score* (left), *Trajectory Quality Score* (middle), and *Combined Score* (right) for *RL-based* and reflective *SA-RL*, *DA-RL*, and *LLM-based* agents.

In summary, our experiment revealed that reflective prompting has a different effect on LLM-based and LLM-assisted RL agents. For the LLM-based agents, reflective prompting led to shorter and lower quality diagnostic conversations, with no significant improvement in diagnostic accuracy. On the other hand, the LLM-assisted RL agents benefited from reflection, showing improvements in diagnostic accuracy. This enhancement was more pronounced for SA-RL agents, which rely more on LLM suggestions.

5.5 RQ3: Agent Efficacy for Different Patients

In our final analysis, we investigated the performance of our agents across the different patients. Figure 7 illustrates the *Post-test Performance Score*, *Trajectory Quality Score*, and *Combined Score* for each patient averaged over all of the subtasks available for that patient in `PharmaSimText` for the *RL-based* agent as well as the reflective *SA-RL*, *DA-RL*, and *LLM-based* agents.

We again observed that the *RL-based* agent showed superior *Post-test Performance Score* across all patients, while the *LLM-based* agent was not able to identify all causes correctly for five out of the nine patients. The *LLM-assisted RL* agents managed to overcome this limitation, with the *SA-RL* agent showing superior performance than the *DA-RL* agent. The opposite result was found for the *Trajectory Quality Score*. While the *LLM-based* agents conducted high-quality diagnostic dialogues, the *RL-based* agent exhibited a suboptimal *Trajectory Quality Score* for all of the patients, often incorporating merely one or two key questions within its diagnostic conversations, highlighting the extent of its deviation from an effective diagnostic interaction. Again, the *LLM-assisted RL* agents overcame this limitation, with the *SA-RL* agent generally showing the highest *Trajectory Quality Score* scores.

Our examination of the *Combined Score* revealed that, except for the *SA-RL* agent, most agents encounter difficulties in scenarios related to Skin and Eye conditions. A closer inspection of their *Post-test Performance Score* and *Trajectory Quality Score* metrics suggested that these agents face challenges in different facets of the scenarios related to these specific patients. A particularly noteworthy observation is the superior performance of the *SA-RL* agent, which overcomes the limitations of purely *RL-based* and *LLM-based* agents across all patient categories.

Given the inferior performance of the *RL-based* agent in the *Trajectory Quality Score*, we examined the dialogues generated by the *RL-based* agent and the *SA-RL* agent within an identical scenario that resulted in a correct diagnosis, as illustrated in Fig. 8. This comparison reveals a pronounced contrast in the conversational dynamics of these two agents. The dialogue led by the *SA-RL* agent exhibits a flow that is markedly more reminiscent of human-like interaction, in contrast to the *RL-based* agent's brief conversation. Notably, the *RL-based* agent's approach is characterized by posing a single key question before directly drawing a conclusion. In comparison, the *SA-RL* agent engages in a more thorough inquiry, covering a broader spectrum of key questions in a logically sequential manner.

In summary, the hybrid LLM-assisted RL agents manage to ovecome the limitations of solely RL-based and LLM-based agents, with the SA-RL agent demonstrating superior performance across all patients. The RL-based agent exhibits a behavior characterized by short conversation, limiting interactions to very few key questions, while the SA-RL agent follows a more human-like behavior.

6. DISCUSSION AND CONCLUSION

In this paper, we explored integration of RL and LLMs to enhance learner models in educational technologies. While RL-based agents show promise in structured learning tasks, they struggle with open-ended environments and skill generalization. Conversely, LLMs excel in generating student-like responses, but fail in constrained action spaces. By combining RL and LLMs, we aimed to develop more generalizable agents for text-based educational settings. We assessed our agents, including *RL-based*, *LLM-based*, and hybrid models, on their ability to conduct diagnostic conversations and make accurate diagnoses in our novel benchmark `PharmaSimText`.

RL-based Agent's Conversation

> **Patient:** I have joint pain. Can you help?
> **Agent:** I want to know about the Patient's location of pain.
> **Patient:** It's mainly my knees and hips that hurt.
> **Agent:** I want to suggest a solution.
> **Patient:** What is the most probable cause of the Patient's problem?
> **Agent:** Osteoarthritis

SA-RL Agent's Conversation

> **Patient:** I have joint pain. Can you help?
> **Agent:** I want to know about the Patient's recent physical activity.
> **Patient:** I can't think of any recent unusual activity.
> **Agent:** I want to know about the Patient's onset of symptoms.
> **Patient:** The pain developed gradually over several weeks.
> **Agent:** I want to know about the Patient's underlying medical conditions.
> **Patient:** I have high blood pressure.
> **Agent:** I want to know about the Patient's presence of swelling redness or warmth.
> **Patient:** No swelling, redness, or warmth that I've noticed.
> **Agent:** I want to know about the Patient's prior joint problems.
> **Patient:** I've had some knee troubles on and off for years.
> **Agent:** I want to know about the Patient's duration of stiffness.
> **Patient:** I feel stiff for about half an hour in the morning.
> **Agent:** I want to know about the Patient's location of pain.
> **Patient:** It's mainly my knees and hips that hurt.
> **Agent:** I want to know about the Patient's lifestyle and diet.
> **Patient:** My diet is pretty balanced, and I rarely drink alcohol.
> **Agent:** I want to know about the Patient's age.
> **Patient:** I am 78 years old.
> **Agent:** I want to know about the Patient's medication history.
> **Patient:** I haven't taken anything for it yet.
> **Agent:** I want to know about the Patient's exercise.
> **Patient:** I don't exercise much due to the pain.
> **Agent:** I want to suggest a solution.
> **Patient:** What is the most probable cause of the Patient's problem?
> **Agent:** Osteoarthritis

Figure 8: **Example diagnostic conversations** conducted by the *RL-based* (top) and *SA-RL* agents (bottom) with the patient with joint pains in a test subtask with Osteoarthritis as the most probable cause.

Specifically, we were interested in answering the following three research questions: Which agent type demonstrates overall superior performance in conducting effective diagnostic conversations and achieving accurate diagnoses for all available patients **(RQ1)**? How does reflective prompting influence the diagnostic performance and conversation quality of LLM-involved agents **(RQ2)**? How do diagnostic performance and conversation quality vary among different agent types across diverse patients **(RQ3)**?

To address our first research question, we assessed four agents: one RL-based, one LLM-based, and two integrating LLMs with RL, in rephrased versions of the scenarios related to different patients in `PharmaSimText` that the agents had not seen before. Effective diagnostic conversations require high-quality conversations and accurate diagnoses. The RL agent excelled in finding the correct diagnosis but struggled in comprehensive diagnostic dialogues due to its limited knowledge. The LLM agent was adept in high-quality diagnostic conversations but tended to misdiagnose patients. LLM-RL integrations were able to address these limitations by enhancing the diagnostic accuracy compared to the *LLM-based* agent and the conversation quality compared to the *RL-based* agent. Among all agents, the *SA-RL* agent achieved the best combination of diagnostic accuracy and conversation quality.

The second research question investigated the benefits of reflective prompting of the LLMs in the LLM-involved agents. To answer this question, we compared the reflective versions of three LLM-involved agents with their none-reflective counterparts. In prior works, reflection showed noticeable improvements in task completion of prompted LLMs [52, 53]. Therefore, we hypothesized a noticeable drop in the performance of the LLM-involved agents after confining them to only one trial. Our results showed a mixed effect for reflection in the solely LLM-based agent and the hybrid agents. For the LLM-based agent, the reflection improved the diagnostic accuracy of the agent, but it decreased the quality of the agent's conversation by shortening its trajectory. For the hybrid agents, the reflective process increased the diagnostic accuracy. We therefore conclude that the effect of reflective prompting depends on the agent type.

To address the third research question, we analyzed the agents over the three metrics for each of the patients separately. We observed that the agents did not struggle with similar patients. In our subsequent analysis, we looked at an example of the conversations done by the *RL-based* agent and the *SA-RL* agent, and we observed that while the *RL-based* agent conversation seemed rushed, the *SA-RL*'s conversation seemed human-like and followed a sequential logic.

One of the limitations of this work is the focus on generalization at a single level of rephrased versions of the scenarios. A few possible generalization levels available `PharmaSimText` are: generalizing to a new wording of a known scenario (wording generalization), to a new diagnosis of a known patient (subtask generalization), and to a new patient (task generalization). Our presented experiments are limited to the wording generalization. Further research should be done within different generalization levels to evaluate current agents and propose new agent frameworks that consider the models' confidence in integration and leverage LLM insights for rapid adaptation of *RL-based* agents to new tasks. Moreover, our proposed reflective process showed limitations in improving the LLM-based agents. This suggests a need for further research for improved reflection in the interactive format of the `PharmaSimText` benchmark. Moreover, future research should consider evaluating the similarity of behavior of these agents to human students to further facilitate their use cases such as evaluating learning environments and collaborative learning.

To conclude, the proposed LLM integration approach represents a promising step towards agents with generalization capabilities in open-ended text-based educational environments. Furthermore, our implemented benchmark facilitates further research in developing agents with generalization capabilities at a higher level.

7. ACKNOWLEDGEMENTS

We thank Dr. Jibril Frej and Dr. Ethan Prihar for their expertise and support. This project was substantially financed by the Swiss State Secretariat for Education, Research and Innovation (SERI).

References

[1] Tanja Käser and Giora Alexandron. Simulated Learners in Educational Technology: A Systematic Literature Review and a Turing-like Test. *International Journal of Artificial Intelligence in Education (IJAIED)*, pages 1–41, 2023.

[2] Kevin Robinson, Keyarash Jahanian, and Justin Reich. Using Online Practice Spaces to Investigate Challenges in Enacting Principles of Equitable Computer Science Teaching. In *Proceedings of the Technical Symposium on Computer Science Education (SIGCSE)*, pages 882–887, 2018.

[3] Daniel Dickison, Steven Ritter, Tristan Nixon, Thomas K. Harris, Brendon Towle, R. Charles Murray, and Robert G. M. Hausmann. Predicting the Effects of Skill Model Changes on Student Progress. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS), Part II*, pages 300–302, 2010.

[4] Tanya Nazaretsky, Sara Hershkovitz, and Giora Alexandron. Kappa Learning: A New Item-Similarity Method for Clustering Educational Items from Response Data. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 2019.

[5] Christopher J. MacLellan, Erik Harpstead, Rony Patel, and Kenneth R. Koedinger. The Apprentice Learner Architecture: Closing the Loop between Learning Theory and Educational Data. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, pages 151–158, 2016.

[6] Lena Pareto. A Teachable Agent Game Engaging Primary School Children to Learn Arithmetic Concepts and Reasoning. *International Journal of Artificial Intelligence in Education (IJAIED)*, 24(3):251–283, 2014.

[7] Adish Singla, Anna N. Rafferty, Goran Radanovic, and Neil T. Heffernan. Reinforcement Learning for Education: Opportunities and Challenges. *CoRR*, abs/2107.08828, 2021.

[8] Jacob Whitehill and Javier R. Movellan. Approximately Optimal Teaching of Approximately Optimal Learners. *IEEE Transactions of Learning Technololy*, 11(2):152–164, 2018.

[9] Song Ju, Min Chi, and Guojing Zhou. Pick the Moment: Identifying Critical Pedagogical Decisions Using Long-Short Term Rewards. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 2020.

[10] Guojing Zhou, Hamoon Azizsoltani, Markel Sanz Ausin, Tiffany Barnes, and Min Chi. Hierarchical Reinforcement Learning for Pedagogical Policy Induction. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*, pages 544–556, 2019.

[11] Anna N. Rafferty, Emma Brunskill, Thomas L. Griffiths, and Patrick Shafto. Faster Teaching via POMDP Planning. *Cognitive Science*, 40(6):1290–1332, 2016.

[12] Aleksandr Efremov, Ahana Ghosh, and Adish Singla. Zero-shot Learning of Hint Policy via Reinforcement Learning and Program Synthesis. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 2020.

[13] Tiffany Barnes and John C. Stamper. Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, pages 373–382, 2008.

[14] Umair Z. Ahmed, Maria Christakis, Aleksandr Efremov, Nigel Fernandez, Ahana Ghosh, Abhik Roychoudhury, and Adish Singla. Synthesizing Tasks for Block-based Programming. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] Victor-Alexandru Padurean, Georgios Tzannetos, and Adish Singla. Neural Task Synthesis for Visual Programming. *Transactions of Machine Learning Research (TMLR)*, 2024.

[16] Christopher J. MacLellan and Adit Gupta. Learning Expert Models for Educationally Relevant Tasks using Reinforcement Learning. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 2021.

[17] Rudy Bunel, Matthew J. Hausknecht, Jacob Devlin, Rishabh Singh, and Pushmeet Kohli. Leveraging Grammar and Reinforcement Learning for Neural Program Synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[18] Reid McIlroy-Young, Siddhartha Sen, Jon M. Kleinberg, and Ashton Anderson. Aligning Superhuman AI with Human Behavior: Chess as a Model System. In *Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1677–1687, 2020.

[19] Paul Denny, Sumit Gulwani, Neil T. Heffernan, Tanja Käser, Steven Moore, Anna N. Rafferty, and Adish Singla. Generative AI for Education (GAIED): Advances, Opportunities, and Challenges. *CoRR*, abs/2402.01580, 2024.

[20] Tom B. Brown et al. Language Models are Few-Shot Learners. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[21] Sébastien Bubeck et al. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. *CoRR*, abs/2303.12712, 2023.

[22] Archana Praveen Kumar, Ashalatha Nayak, Manjula Shenoy K, Chaitanya, and Kaustav Ghosh. A Novel Framework for the Generation of Multiple Choice Question Stems Using Semantic and Machine-Learning Techniques. *International Journal of Artificial Intelligence in Education (IJAIED)*, pages 1–44, 2023.

[23] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *Proceedings of the Conference on International Computing Education Research (ICER)*, 2022.

[24] Tung Phung, Victor-Alexandru Padurean, José Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Singla, and Gustavo Soares. Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors. In *Proceedings of the Conference on International Computing Education Research - Volume 2 (ICER V.2)*, 2023.

[25] Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. Automated Distractor and Feedback Generation for Math Multiple-choice Questions via In-context Learning. *NeurIPS'23 Workshop on Generative AI for Education (GAIED)*, 2023.

[26] Maciej Pankiewicz and Ryan Shaun Baker. Large Language Models (GPT) for Automating Feedback on Programming Assignments. *CoRR*, abs/2307.00150, 2023.

[27] Arne Bewersdorff, Kathrin Seßler, Armin Baur, Enkelejda Kasneci, and Claudia Nerdel. Assessing Student Errors Experimentation Using Artificial Intelligence and Large Language Models: A Comparative Study with Human Raters. *CoRR*, abs/2308.06088, 2023.

[28] Dollaya Hirunyasiri, Danielle R. Thomas, Jionghao Lin, Kenneth R. Koedinger, and Vincent Aleven. Comparative Analysis of GPT-4 and Human Graders in Evaluating Praise Given to Students in Synthetic Dialogues. *CoRR*, abs/2307.02018, 2023.

[29] Tung Phung, Victor-Alexandru Pădurean, Anjali Singh, Christopher Brooks, José Cambronero, Sumit Gulwani, Adish Singla, and Gustavo Soares. Automating Human Tutor-Style Programming Feedback: Leveraging GPT-4 Tutor Model for Hint Generation and GPT-3.5 Student Model for Hint Validation. In *Proceedings of the International Learning Analytics and Knowledge Conference (LAK)*, 2024.

[30] Zachary A. Pardos and Shreya Bhandari. Learning Gain Differences between ChatGPT and Human Tutor Generated Algebra Hints. *CoRR*, abs/2302.06871, 2023.

[31] Anaïs Tack and Chris Piech. The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 2022.

[32] Unggi Lee, Sanghyeok Lee, Junbo Koh, Yeil Jeong, Haewon Jung, Gyuri Byun, Yunseo Lee, Jewoong Moon, Jieun Lim, and Hyeoncheol Kim. Generative Agent for Teacher Training: Designing Educational Problem-Solving Simulations with Large Language Model-based Agents for Pre-Service Teachers. *NeurIPS'23 Workshop on Generative AI for Education (GAIED)*, 2023.

[33] Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. Ruffle&Riley: Towards the Automated Induction of Conversational Tutoring Systems. *NeurIPS'23 Workshop on Generative AI for Education (GAIED)*, 2023.

[34] Manh Hung Nguyen, Sebastian Tschiatschek, and Adish Singla. Large Language Models for In-Context Student Modeling: Synthesizing Student's Behavior in Visual Programming. *CoRR*, abs/2310.10690, 2023.

[35] Julia M. Markel, Steven G. Opferman, James A. Landay, and Chris Piech. GPTeach: Interactive TA Training with GPT-based Students. In *Proceedings of the Conference on Learning @ Scale (L@S)*, pages 226–236, 2023.

[36] Ruoyao Wang, Peter A. Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is Your Agent Smarter than a 5th Grader? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11279–11298, 2022.

[37] Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. Do Embodied Agents Dream of Pixelated Sheep: Embodied Decision Making using Language Guided World Modelling. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 26311–26325, 2023.

[38] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, Jacob Andreas, Igor Mordatch, Antonio Torralba, and Yuke Zhu. Pre-Trained Language Models for Interactive Decision-Making. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[39] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding Pretraining in Reinforcement Learning with Large Language Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8657–8677, 2023.

[40] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward Design with Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[41] Nan Li, William W. Cohen, Kenneth R. Koedinger, and Noboru Matsuda. A Machine Learning Approach for Automatic Student Model Discovery. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, pages 31–40, 2011.

[42] Albert T. Corbett and John R. Anderson. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 2005.

[43] Louis Faucon, Lukasz Kidzinski, and Pierre Dillenbourg. Semi-Markov Model for Simulating MOOC Students. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, pages 358–363, 2016.

[44] Anthony F. Botelho, Seth Adjei, and Neil T. Heffernan. Modeling Interactions Across Skills: A Method to Construct and Compare Models Predicting the Existence of Skill Relationships. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, pages 292–297, 2016.

[45] Anna N. Rafferty, Joseph Jay Williams, and Huiji Ying. Statistical Consequences of Using Multi-Armed Bandits to Conduct Adaptive Educational Experiments. *Journal of Educational Data Mining (JEDM)*, 11:47–79, 2019.

[46] John Mui, Fuhua Lin, and M Ali Akber Dewan. Multi-Armed Bandit Algorithms for Adaptive Learning: A Survey. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*, pages 273–278, 2021.

[47] Adish Singla and Nikitas Theodoropoulos. From {Solution Synthesis} to {Student Attempt Synthesis} for Block-Based Visual Programming Tasks. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 2022.

[48] Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. SOTOPIA: Interactive Evaluation for Social Intelligence in Language Agents. *CoRR*, abs/2310.11667, 2023.

[49] Alexander Pan, Chan Jun Shern, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Jonathan Ng, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. Do the Rewards Justify the Means? Measuring Trade-Offs Between Rewards and Ethical Behavior in the MACHIAVELLI Benchmark. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 26837–26867, 2023.

[50] Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. Deep Reinforcement Learning with a Natural Language Action Space. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[51] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[52] Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter A. Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. CLIN: A Continually Learning Language Agent for Rapid Task Adaptation and Generalization. *CoRR*, abs/2310.10134, 2023.

[53] Noah Shinn, Beck Labash, and Ashwin Gopinath. Reflexion: An Autonomous Agent with Dynamic Memory and Self-Reflection. *CoRR*, abs/2303.11366, 2023.

[54] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[55] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing Atari with Deep Reinforcement Learning. *CoRR*, abs/1312.5602, 2013.

[56] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *CoRR*, abs/1607.04606, 2016.

[57] Prithviraj Ammanabrolu and Matthew J. Hausknecht. Graph Constrained Reinforcement Learning for Natural Language Action Spaces. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.