

PharmaSimText: A Text-Based Educational Playground filled with RL-LLM Agents That Work Together Even in Disagreement

Bahar Radmehr
EPFL
Lausanne, Switzerland
bahar.radmehr@epfl.ch

Tanja Käser
EPFL
Lausanne, Switzerland
tanja.kaeser@epfl.ch

Adish Singla
MPI-SWS
Saarbrücken, Germany
adishs@mpi-sws.org

There has been a growing interest in developing simulated learners 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 learner’s ability to generalize skills across tasks. In this paper, we aim to enhance simulated learners’ generalization capabilities in less-structured 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, (ii) *LLM-based* agents that leverage the model’s general knowledge and reasoning through prompting, and (iii) hybrid *RL-LLM* agents that combine these two strategies to improve agents’ performance and generalizability. To support the development of these agents, we introduce `PharmaSimText`, a novel benchmark developed with expert-evaluated GPT-4 generations derived from a virtual pharmacy environment designed for practicing diagnostic conversations. After experimenting with *RL-based* and *LLM-based* agents using GPT-4 and open-source LLMs along with a wide range of strategies for combining them, we find that *RL-based* agents are good at completing tasks, but not at asking quality diagnostic questions. Conversely, *LLM-based* agents are better at asking diagnostic questions, but not at completing tasks. Finally, specific variations of hybrid *RL-LLM* agents enable us to overcome these limitations. Our findings highlight the potential of combining methods based on RL and LLMs in creating generalizable agents that have solutions close to human ones with the LLM component, while remaining faithful to controlled environments with the RL component. The source code and benchmark are available on GitHub.¹

Keywords: reinforcement learning, large language models, text-based educational environments, simulated learners

¹<https://github.com/epfl-ml4ed/PharmaSimText>

1. INTRODUCTION

Simulated learners are crucial to the advancement of educational technologies, serving as a versatile tool for enhancing learning and teaching. Simulating student interactions and data provides a safe and controlled environment for teacher training, allowing educators to refine their methods without directly affecting real students (Robinson et al., 2018). Simulated learners also facilitate the development and evaluation of adaptive learning systems or new algorithms (Nazaretsky et al., 2019). Furthermore, they have been applied for testing theories of learning (MacLellan et al., 2016) and foster collaboration skills in students through interacting with virtual peers (Pareto, 2014).

Reinforcement learning (RL) offers a promising avenue for developing simulated learners. Existing works on RL for education have however primarily focused on developing techniques for curriculum optimization (Whitehill and Movellan, 2018; Ju et al., 2020; Zhou et al., 2019; Rafferty et al., 2016), providing tailored hints and feedback (Efremov et al., 2020; Barnes and Stamper, 2008), and generating educational content (Ahmed et al., 2020; Padurean et al., 2024). Only a limited number of studies have examined the use of RL-based learner agents that effectively operate in online learning environments (MacLellan and Gupta, 2021; Bunel et al., 2018). 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 with engineered features obtained from the existing structure (Singla et al., 2021; McIlroy-Young et al., 2020; MacLellan and Gupta, 2021). Because of their reliance on hand-crafted features and engineered state representations, these RL agents can however not be used in unstructured domains or to generalize to unseen tasks.

Recent advances in generative AI, in particular Large Language Models (LLMs), provide new opportunities to drastically improve state-of-the-art educational technology (Denny et al., 2024). LLMs can generate coherent, contextually relevant content, engage in meaningful dialogues, and perform linguistic tasks without explicit training (Brown et al., 2020; Bubeck et al., 2023). So far, in education, LLMs (mostly GPT) have mainly been applied for generating educational content (Kumar et al., 2023; Sarsa et al., 2022; Phung et al., 2023a), automating grading and feedback processes (McNichols et al., 2023; Pankiewicz and Baker, 2023; Bewersdorff et al., 2023; Hirunyasiri et al., 2023; Phung et al., 2024; Pardos and Bhandari, 2023), and facilitating the development of collaborative systems (Tack and Piech, 2022; Lee et al., 2023; Schmucker et al., 2023). A few works have also used LLMs for modeling learners in programming domains (Nguyen et al., 2024) or for simulating students' behaviors as a basis for an interactive tool for teacher training (Markel et al., 2023). Despite their proficiency in linguistic tasks, LLMs often fail to make feasible decisions within constrained environments, a domain where RL agents excel (Wang et al., 2022).

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 (Nottingham et al., 2023; Li et al., 2022; Du et al., 2023; Kwon et al., 2023). However, most of these approaches have focused on leveraging LLMs for assistance during training, which introduces the risk of hybrid agents inheriting LLMs' decision-making limitations in constrained environments.

To develop performant RL agents, an enriched benchmark environment is essential. Prior

work has developed benchmarks for various applications such as Gymnasium (Towers et al., 2024) to test newly designed RL algorithms, ROBEL (Ahn et al., 2019) for training robotic agents, SafeBench (Xu et al., 2022) for autonomous vehicles, and Flow (Wu et al., 2017) for traffic control agents. However, only a few studies have developed benchmarks (e.g., Tutor-Gym (MacLellan and Gupta, 2021)) for educational applications. These benchmarks only cover well-structured tasks in math, and no benchmarks have been designed to develop simulated students for less-structured text-based learning 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 less-structured text-based simulation benchmark, `PharmaSimText`, adapted from the `PharmaSim` virtual pharmacy environment designed for practicing diagnostic conversations. We developed the benchmark by generating patient cases and diagnoses using GPT-4 (OpenAI, 2023), which were subsequently evaluated by a pharmacy expert. We present three types of agents: (i) *RL-based* agents employing natural language based representations, (ii) *LLM-based* agents invoked through prompting, and (iii) a wide range of hybrid *RL-LLM* agents where LLMs and RL agents collaborate in the inference phase using various strategies.

We extensively evaluate all agents based on their ability to engage in effective diagnostic conversations and achieve accurate diagnoses on the `PharmaSimText` benchmark. We employ a wide range of LLMs (GPT-4, four open-source LLMs, and two medically fine-tuned LLMs) in our experiments. With our experiments, we aim to address three research questions: Which agent type demonstrates the best performance in conducting effective diagnostic conversations and achieving accurate diagnoses (**RQ1**)? How does the specific LLM (GPT-4 or open-source LLMs) used in the LLM-involved agents affect agent performance (**RQ2**)? How does changing the power balance between the RL and LLM agents by introducing a disagreement mechanism in the strategy used for combining them influence the performance of the *RL-LLM* agents (**RQ3**)?

Our results demonstrate that combining RL and LLMs boosts performance, effectively balancing accurate diagnosis along with high-quality diagnostic conversations. Furthermore, using open-source models such as Llama3-70B (Dubey et al., 2024) or medically fine-tuned ones such as OpenBioLLM-70B (Ankit Pal, 2024) tends to decrease agent performance (see section 5.3 for discussion). Finally, the strategy used for combining RL and LLM agents has a large impact on performance. In this work, we contribute the following: (1) integration of RL and LLM agents for development of generalizable simulated learners in text-based educational environments, (2) developing `PharmaSimText` as a less-structured text-based benchmark for developing and evaluating simulated learner agents by extending the `PharmaSim` virtual pharmacy environment using LLM-generated patient scenarios evaluated by a pharmacy expert, and (3) comparative analysis of *RL-based*, *LLM-based*, and *RL-LLM* Agents and extensive investigation of effect of different LLMs, integration strategies, and prompting methods.

2. RELATED WORK

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

2.1. SIMULATED LEARNER AGENTS IN EDUCATIONAL ENVIRONMENTS

There is a large body of research on simulating learners in digital environments. A recent systematic review (Käser and Alexandron, 2023) has identified three primary methods for simulating learners. Cognitive-based simulations provide rich, but not generalizable learner representations, for example by generating cognitive models from problem-solving demonstrations (e.g., SimStudent (Li et al., 2011)). Probabilistic methods simulate learners based on students' behavior observed in previously collected data (Corbett and Anderson, 2005; Faucon et al., 2016). Similarly, data-driven simulations replay real student data to evaluate how changes in the environment affect learner model parameters (Botelho et al., 2016). While these two methods produce less detailed representations, they offer better generalizability.

RL is a promising tool to address these limitations. However, in the education domain, this framework has been primarily applied for pedagogical policy induction (Whitehill and Movellan, 2018; Ju et al., 2020; Zhou et al., 2019; Rafferty et al., 2016), providing tailored hints (Barnes and Stamper, 2008; Efremov et al., 2020), generating educational content (Ahmed et al., 2020; Padurean et al., 2024), assessing interventions in educational platforms (Rafferty et al., 2019; Mui et al., 2021), and designing tests for evaluating students' knowledge (He-Yueya and Singla, 2021). Despite its potential, the exploration of RL-based learner agents for effective operation in learning environments remains limited (MacLellan and Gupta, 2021; Bunel et al., 2018). Prior work has for example used proximal policy optimization for designing simulated learners trying to excel in intelligent tutoring systems (MacLellan and Gupta, 2021) or employed neural and symbolic program synthesis to create student attempts in a block-based programming environment (Singla and Theodoropoulos, 2022). In this paper, we develop a series of learner agents trying to showcase effective strategies in a less-structured text-based environment.

2.2. AGENTS FOR TEXT-BASED INTERACTIVE GAMES

The growing interest in developing agents for text-based interactive games, especially those mimicking real-world scenarios (Wang et al., 2022; Wang et al., 2024; Pan et al., 2023), has led to diverse methodologies such as RL (He et al., 2016; Ammanabrolu and Hausknecht, 2020) and prompting LLMs (Yao et al., 2023; Majumder et al., 2023; Ichter et al., 2022; Shinn et al., 2023). A well-known example is the game ScienceWorld (Wang et al., 2022), where players engage in scientific experiments through environment exploration.

RL Methods. Within the RL framework, state-of-the-art research employs Deep Reinforced Relevance Networks (DRRN) (He et al., 2016), treating text-based interactions as partially-observable Markov decision processes and learning distinct text representations for states and actions to estimate Q-values via a scorer network. In cases with structured relationships between environment elements, methods such as KG-A2C (Ammanabrolu and Hausknecht, 2020) create graph-structured state representations along with natural language embedding state representations. While effective in certain contexts, these methods can struggle with more complex interactive environments due to limited reasoning and planning capabilities.

LLM-based Methods. LLM-based methods use prompts at each interaction step for strategic planning and action selection. While some studies (e.g., Yao et al. 2023) engage in a single interaction round with the environment, others (e.g., Shinn et al. 2023; Majumder et al. 2023) use a multi-round approach, facilitating iterative refinement through repeated attempts. Techniques such as SAYCAN (Ichter et al., 2022), REACT (Yao et al., 2023), and REFLEXION (Shinn et al., 2023) represent the state-of-the-art approaches in this domain. SAYCAN merges LLM predictions with a value function to ensure action feasibility, REACT introduces a virtual 'think'

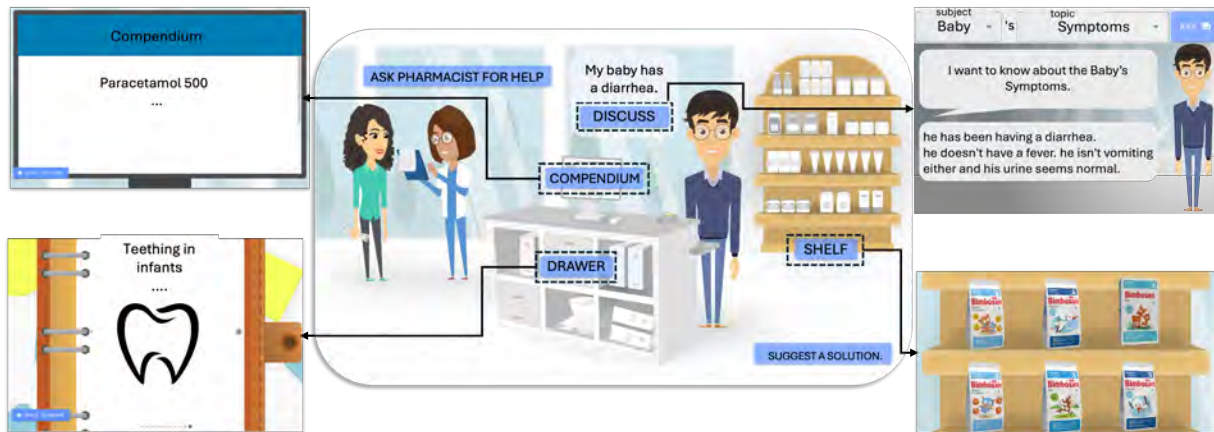


Figure 1: ‘**Father Inquiry**’ scenario in **PharmaSim** - A simulated pharmacy setting designed for practicing diagnostic conversational skills, where participants engage with a father seeking guidance for his infant child’s diarrhea.

action, enabling LLMs to generate subgoals while planning, and REFLEXION builds on this with a multi-round strategy for refined planning, reflecting on previous trials.

In this paper, we develop a series of agents for a text-based educational environment simulating real-world scenarios happening in a pharmacy.

2.3. 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, [Du et al. \(2023\)](#) utilized text corpora to pre-train agents, thereby shaping their exploration by suggesting goals based on the agents’ current state descriptions. Furthermore, [Kwon et al. \(2023\)](#) proposed an approach to simplify reward design by employing LLMs to generate reward signals from textual prompts that describe desired behaviors. In a similar vein, [Nottingham et al. \(2023\)](#) applied 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, especially in using LLMs to support RL agents in generalizing to new tasks and environments post training. In this work, we aim to bridge this gap by focusing on combining LLMs and RL agents’ decisions during generalization to new settings.

3. PHARMASIMTEXT BENCHMARK

We created **PharmaSimText**, a text-based educational environment, as a test-bench for developing agents capable of handling text-based learning tasks and generalizing in them.

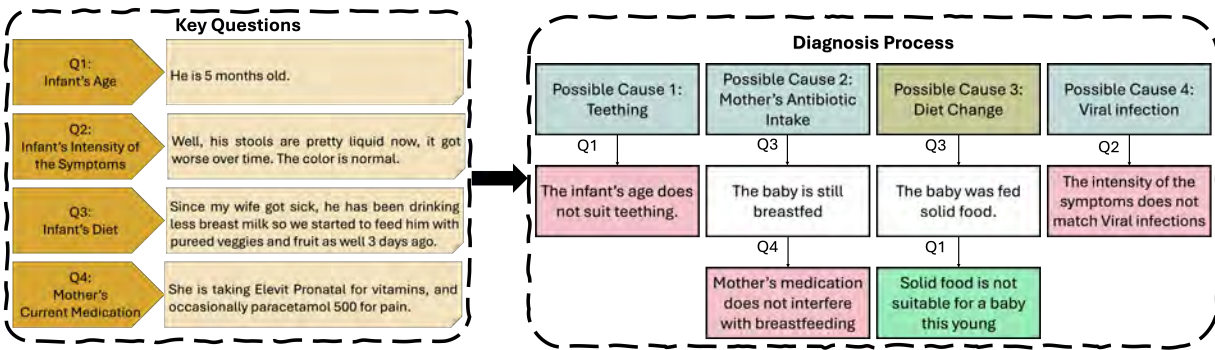


Figure 2: **Diagnostic Strategy in the ‘Father Inquiry’ Scenario of PharmaSim**, illustrating the process of identifying the most likely cause of the infant’s diarrhea. Four key questions of the father is necessary to collect crucial information for determining the most probable cause of the child’s diarrhea among four potential causes.

PharmaSimText is an interactive text-based environment designed based on PharmaSim, a scenario-based learning platform. It simulates real-world interactions between a pharmacist and a patient in a pharmacy setting. PharmaSimText requires students to engage in guided inquiry by selecting effective questions from an extensive pre-selected list, offering greater flexibility compared to traditional environments with limited instructor-defined options. PharmaSimText extends PharmaSim by adding scenarios generated using GPT-4 and evaluated by a pharmacy expert. The benchmark includes near 40 possible diagnoses in total for 7 different medical conditions with 10 different phrasing versions available for each component resulting in a rich problem space for developing and evaluating learner agents. This benchmark covers the contents of Switzerland’s pharmacy assistant training program that can be used for students’ learning.

3.1. PHARMASIM

PharmaSim is a scenario-based learning environment designed to support the development of diagnostic conversational skills in pharmacy assistant students. Currently, two different scenarios designed with insights from pharmacy experts are available in the game. Fig. 1 shows the ‘Father Inquiry’ scenario, where a father visits the pharmacy looking for help with his child’s diarrhea.

All scenarios follow the same structure: 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 of the patient, seeking help from the pharmacist, searching about different kinds of medicine, looking for the specifications of products available on the shelf, reading/consulting documentation of medical knowledge and facts, 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 the most probable causes along with their likelihoods significantly depends on a set of questions gathering the essential information, which we henceforth refer to as *key questions*. For example, the ‘Father Inquiry’ scenario presents four possible causes for

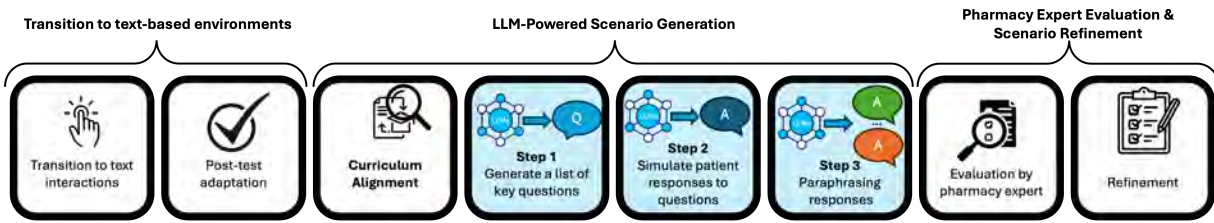


Figure 3: **Pipeline of PharmaSimText benchmark development.** A three-stage development process was followed: (1) Transition to text-based environments to facilitate agent training via interaction and post-test adaptation. (2) LLM-powered scenario generation for extending existing PharmaSim scenarios and creating new ones based on pharmacy assistant training curricula. The steps involving an LLM are marked in blue. (3) Pharmacy Expert Evaluation & Scenario Refinement to ensure the quality of the benchmark.

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 while considering several causes. 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, we followed the three main stages illustrated in Fig. 3. We first transitioned PharmaSim to a text-based environment, enabling agent interaction and assessment. We then employed GPT-4 to extend the existing scenarios and to create new ones. Finally, all scenarios were evaluated by a pharmacy expert and refined based on their feedback.

3.2.1. Transition to text-based environments

To transform PharmaSim into a text-based environment, two modifications were made. First, we streamlined interactions into two types: ‘‘I want to know about the {character}’s {topic}’’ for asking the patient questions about various characters, and ‘‘I want to suggest a solution’’ for proceeding to the post-test. Other action types available in PharmaSim, such as seeking help from the pharmacist, searching for different medicines, examining product specifications on the shelf, and consulting medical documentation, were excluded to keep the focus on the diagnostic conversation. Second, we modified the post-test questions to provide agents with a feasible assessment. To accomplish this, we revised the three-causes question to focus exclusively on the most likely cause. Furthermore, *key questions* were used to evaluate the agents’ trajectory after playing the game.

3.2.2. LLM-Powered Scenario Generation

We expanded the two scenarios available in the original environment across three dimensions: (1) introducing new patients, (2) varying the scenarios to alter the most probable cause of each patient’s problem, and (3) rephrasing patient responses to provide a broader range of answers. Given the scale of extension, relying solely on human expertise was impractical. Instead, we

Table 1: **Resulting quality of GPT-4 generated scenarios.** Correctness, Realism, Completeness, Conceal, Clarity, and Consistency per scenario as well as mean \pm standard deviation over all scenarios. GPT-4 achieves high values in all measures.

Patient	Correctness	Realism	Completeness	Conceal	Clarity	Consistency
Breastfeeding-related	0.85	0.85	0.88	1.00	1.00	1.00
Urological	0.95	0.87	0.86	1.00	1.00	1.00
Gynecological	0.77	0.72	0.73	1.00	1.00	1.00
Eye-related	0.94	0.94	0.79	1.00	1.00	1.00
Joint Pain	0.93	0.88	0.9	1.00	1.00	0.50
Sore Throat	0.94	0.92	0.78	1.00	1.00	1.00
Overall	0.90 \pm 0.07	0.86 \pm 0.08	0.82 \pm 0.07	1.00 \pm 0.00	1.00 \pm 0.00	0.92 \pm 0.20

leveraged the generative capabilities of LLMs combined with human insights to develop the scenarios in PharmaSimText.

Curriculum Alignment. We consulted the pharmacy assistant training curriculum of Switzerland to align the expanded and newly generated scenarios with students’ training program as well as with medical facts. Specifically, 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 LLM-powered scenario creation process as reflected in Fig. 3 involved three steps: (2.1) we prompted the LLM to generate a list of *key questions* aimed at diagnosing the most probable cause of the patient’s problem, (2.2) the LLM was tasked with simulating patient responses to key questions, assuming each illness on the list as the most probable cause, as well as generating answers to other common patient inquiries typically addressed by pharmacists, and (2.3) the LLM was employed to diversify existing patient responses through paraphrasing for enhancing the scenarios’ complexity. We used GPT-4 (OpenAI, 2023) as the LLM for scenario creation; the exact prompts employed can be found in Appendix 8.1. To further augment the benchmark’s complexity, several fictional characters were introduced as distractors, enabling players to engage in more nuanced interactions.

3.2.3. Pharmacy Expert Evaluation & Scenario Refinement

To ensure the quality of the generated scenarios, we had them evaluated by a pharmacy expert. Specifically, we assessed the scenarios along six dimensions: *Correctness*, *Realism*, *Completeness*, *Conceal*, *Clarity*, and *Consistency*, using a rubric developed based on measures for the evaluation of LLM generations suggested in prior work (Nguyen et al., 2024; Phung et al., 2023b). *Correctness* measures whether the provided patient answers align with medical facts. *Realism* assesses whether the patient responses mimic real-life interactions. *Completeness* evaluates whether the predefined questions cover all necessary information for an accurate diagnosis. *Conceal* checks whether the LLM-generated answer does not directly give away the correct diagnosis. Finally, *Clarity* assesses the readability of the patient answers and *Consistency* examines the patient answers for potential contradictions. Three of the measures (*Correctness*, *Realism*,

and *Completeness*) were directly assessed by the pharmacy expert. The rest of the measures (*Conceal*, *Clarity*, and *Consistency*) were calculated based on annotations of the pharmacy expert’s answers by the researchers. To ensure the quality of the annotation, two scenarios were independently coded by two researchers with expertise in the field of Educational Technology including a researcher who did not co-author the study, resulting in an inter-rater agreement of $\kappa_{Conceal} = 1.0$ with 10 samples (number of possible causes), $\kappa_{Clarity} = 0.81$ with 170 samples (number of question answers), and $\kappa_{Consistency} = 1.0$ with 10 samples (number of possible causes). The better agreement in *Conceal* and *Consistency* is because they are calculated using logical arguments. Further implementation details about the rubric is detailed in Table 4 in Appendix 8.2.

We calculated all measures after one round of LLM generation and pharmacy expert feedback. The resulting scores are illustrated in Table 1. While the LLM generations were mostly medically correct (mean *Correctness* score of 0.9), we observed fluctuations between the different scenarios. Specifically, the LLM performed less well in breastfeeding-related and gynecological topics, suggesting lower prior knowledge of GPT-4 in these topics. The same trend is present for *Realism*. Notably, in the initial steps of prompt design, we observed a tendency of the LLM to generate responses based on extreme cases of a certain diagnosis, lowering the *Realism*. This problem was mitigated after adjustments to our prompt. Regarding *Completeness*, the initial suggestions from the LLM missed 1-3 key questions necessary to cover all the information needed for diagnosis across different patient cases as specified by the pharmacy expert, leading to a rather low score (*Completeness* score of 0.82). Hence, while the LLM’s assistance in designing educational content is beneficial to some extent, there is a need for more human expert involvement at this step. Furthermore, the LLM did not directly reveal any diagnoses in the patient answers (perfect *Conceal* score of 1), as specified in the prompt. Finally, when investigating *Clarity* and *Consistency*, we found that the generations were clear and consistent almost all the time, with the LLM failing to be consistent only in one case (‘Joint Pain’ scenario). These results emphasize the capability of LLMs to assist in the generation of educational content.

Following this evaluation, all of the scenarios were refined based on the feedback given by the pharmacy expert. We performed several rounds of refinement, resulting in perfect scores in all the 6 dimensions for the final scenarios provided in the benchmark². The resulting scenarios cover seven medical problems (e.g., sore throat), including the possible causes and number of key questions for each, and are detailed in Table 5 in Appendix 8.3. On average, each scenario presents seven potential causes for the patient’s problem, resulting in a total of 37 scenario variations. Each scenario further necessitates the identification of an average of 7.8 *key questions* by the player. Patient responses in each scenario are articulated in ten diverse phrasings to enhance the depth and variability. We, therefore, obtain an extensive problem space with many variations for an agent to train on, providing a rich environment for further studies on agents for text-based interactive tasks and agents’ generalizability.

4. AGENTS FOR PHARMASIMTEXT

We developed three types of agents for `PharmaSimText` representing various degrees of RL and LLM integration as illustrated in Fig. 4: pure *RL-based* agents, pure *LLM-based* agents, and *RL-LLM* agents.

²<https://github.com/epfl-ml4ed/PharmaSimText>

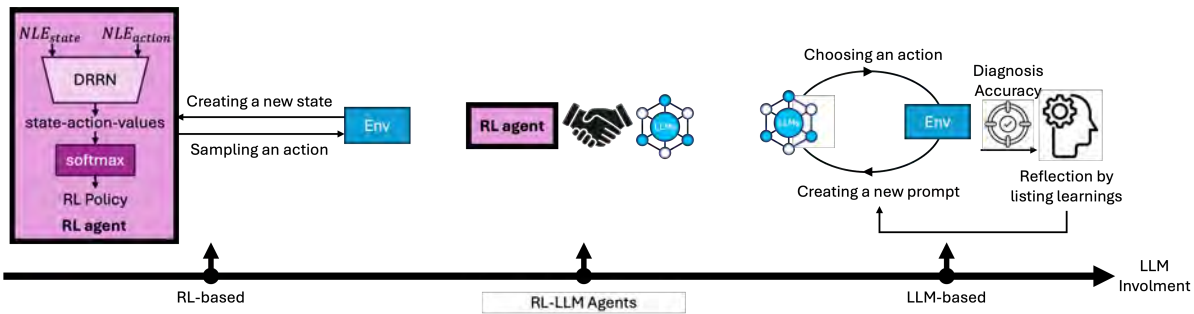


Figure 4: **Overview of agents.** In the *RL-based* agent (left), a DRRN evaluates the state-action embeddings at each step and returns the state-action value for all the available actions. A softmax probability distribution is then used to determine the next action. The *LLM-based* agent (right) prompts an LLM at each step to choose the next best action. This agent is given multiple trials at solving a task, and reflects on what can be improved between trials. For the *RL-LLM* agents, the LLM and RL switch roles, between one of them suggesting a set of actions and the other one choosing from this set.

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 (Sutton and Barto, 2018). 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 specific action in a given state. Deep Q-Networks (DQNs)(Mnih et al., 2013) approximate these Q-values using deep neural networks, enabling the handling of complex, high-dimensional environments by learning to predict the Q-values directly from the agent’s state. DQNs are trained through interactions with the environment, using their experience to make their estimations of Q-values more accurate iteratively.

Following previous work on text-based games (Wang et al., 2022), we utilized the state-of-the-art method, a DRRN (He et al., 2016) as the *RL-based* agent for interacting with PharmaSimText. The DRRN is designed to learn distinct representations for text-based states and actions by employing two separate networks: a state encoder and an 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 a reaction r_t from the environment 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 reaction 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 received reactions 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 (Bojanowski et al., 2017) to generate text representations for both reactions 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 (Ammanabrolu and Hausknecht, 2020). Second, unlike the text-based environments for which DRRNs were originally proposed, the reaction 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. Instead, the full reaction history is needed as a part of the context given to the agent. Therefore, we introduced a unit called the **state updater** before the state encoder that takes the previous embedded state $\text{emb-}s_{t-1}$ and the new embedded reaction $e(r_t)$ and returns the updated state after the current reaction $\text{emb-}s_t$. 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 that 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 reaction embeddings in the history returns $\text{emb-}s_t = \text{emb-}s_{t-1} + e(r_t)$ as the new embedded state $\text{emb-}s_t$.

4.2. LLM-BASED AGENTS

The agents based on LLMs prompt an LLM at each step of interacting with the environment to determine the best next action to take. During interaction with the LLM, the agent provides a prompt that contains the task description, past interactions with the patient (which includes the questions asked by the agent and responses from the patient), and valid actions available in the current state to determine the most appropriate next action. Moreover, the task description is structured as `Find the cause behind the patient's problem, and the interaction history is presented as a dialogue between the patient and the agent.` Action texts are labeled as agent's questions and environment feedback texts are labeled as patient responses. Each action type among the valid actions is formatted as a function along with its permissible input values for the LLM to interpret. This functional formatting is complemented by a description explaining the action's purpose. For instance, the interaction "I want to ask about the character's (e.g., baby's) topic (e.g., symptoms)" is presented as `ask(character, topic): Asking a question about the topic related to the character, followed by a list of valid subjects and topics.` This formatting strategy decreases 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 $n = 3$ attempts to control the computational cost of prompting. Should all suggested actions remain invalid, we selected the valid action that had the smallest distance in the natural language embedding space to the n -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.

Recent advancements in developing LLM-based agents have shown a noticeable improvement in their ability to complete tasks employing a strategy called reflective prompting which gives the LLM multiple trials to accomplish a task and requires the LLM to reflect between each trial (Majumder et al., 2023). To improve the LLM-based agent further, we therefore implemented reflective prompting. In the following, we refer to the LLM-based agent described above, which is given only one trial, as *non-reflective*, and to the LLM-based agent given multiple trials, as *reflective*.

The *reflective agent* employs a prompting strategy akin to that of the *non-reflective agent* to determine the optimal subsequent action. The *non-reflective agent* prompt is augmented with a segment including learnings from prior interactions with the same patient having the same diagnosis. After completing each trial, the LLM is asked to reflect on its previous strategy based on the observed outcomes. Following this, the agent updates its textual memory of previous learnings, and the updated memory is used for prompting in the next trial. This approach is inspired by research on self-reflective LLMs, notably the continually learning language agent CLIN (Majumder et al., 2023). Similar to CLIN, we constructed the learning memory using causal formats such as “X is necessary for Y” to guide future interactions. As a result, the reflective agent can dynamically adapt and refine its approach, enhancing its decision-making process over time.

4.3. RL-LLM AGENTS THAT AGREE&ACCEPT

RL-based agents’ perspective is limited by what they have experienced during training, which could make them less effective in tasks with unfamiliar elements not encountered during training (Yao et al., 2021). Furthermore, they find the shortest path to achieving the reward, often making their solutions very different from human ones (McIlroy-Young et al., 2020). LLMs, on the other hand, can generalize to new cases in context and provide more human-like solutions; however, they cannot perform in constrained environments or change some of their prior biases based on their inputs (Lin et al., 2023). To address each agent’s weaknesses while keeping their strengths, we combined them in a series of agents known as *RL-LLM* agents in three versions that either Agree&Accept, Disagree&Veto, or Disagree&Interact. Here, we introduce the most fundamental version of *RL-LLM* Agents that Agree&Accept.

As illustrated in Fig. 5, in these agents, the LLM and *RL-based* models switch roles between being a Suggestion-Maker making high-level plans and a Decision-Maker determining the most optimal action given the constraints of the Suggestion-Maker’s plan, creating two types of agents: *RL-Suggests-LLM-Accepts&Decides* ($RL \rightarrow LLM_{ACCEPT}$) and *LLM-Suggests-RL-Accepts&Decides* ($LLM \rightarrow RL_{ACCEPT}$).

4.3.1. *RL-Suggests-LLM-Accepts&Decides* ($RL \rightarrow LLM_{ACCEPT}$):

At a time step t , we collect the top- k actions $RL-Suggested_t$ based on the *RL-based* agent’s policy. The LLM is then prompted to choose the best action from $RL-Suggested_t$. The prompt contains the task description, history of interactions, prior experience with the patient, and actions in $RL-Suggested_t$ (see Fig.5 (left)). Since there are different numbers of actions available at the interaction and post-test steps, we set $k_{interaction} = 5$ and $k_{post-test} = 2$ to ensure the

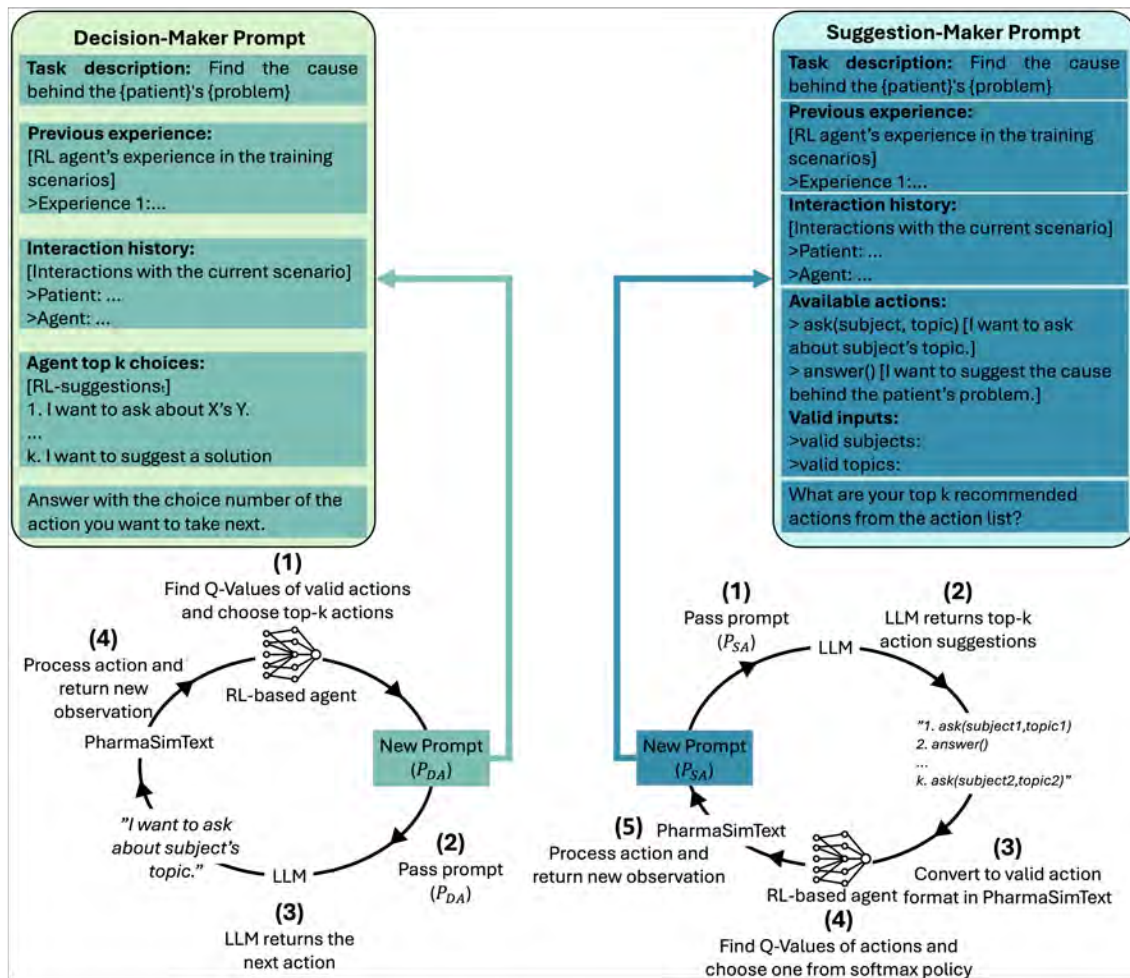


Figure 5: **RL-LLM agents that Agree&Accept.** The LLM and RL agents alternate roles: one suggests actions, the other selects from them. In the *RL-Suggests-LLM-Accepts&Decides* ($RL \rightarrow LLM_{ACCEPT}$) agent (left), the LLM selects an action from suggestions provided by the RL agent. In the *LLM-Suggests-RL-Accepts&Decides* ($LLM \rightarrow RL_{ACCEPT}$) agent (right), the LLM suggests several actions at each step for the RL agent to choose from.

suggestions cover optimal actions from Suggestion-Maker’s perspective while allowing Decision-Maker enough freedom of choice.

4.3.2. *LLM-Suggests-RL-Accepts&Decides* ($LLM \rightarrow RL_{ACCEPT}$):

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 *RL-based* agent calculates Q-values for each action in $LLM-Suggested_t$, and the next action is sampled from the soft-max distribution of the Q-values. The prompt for the LLM (see Fig.5 (right)) contains the task description, the history of interactions, prior experience with the patient, and valid actions for the current state. We again set $k_{interaction} = 5$ for the interaction steps and $k_{post-test} = 2$ for the post-test steps.

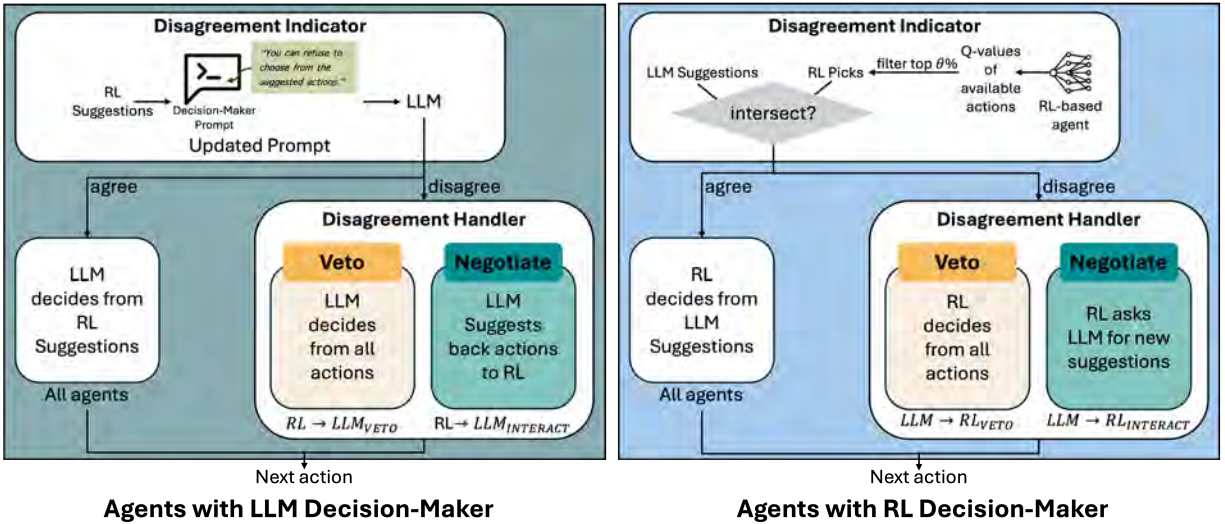


Figure 6: **Disagreement Mechanism in agents with different Decision-Makers and Disagreement Handling:** In agents with LLM Decision-Maker, after receiving the RL suggestions, the LLM is prompted to choose from them or disagree with them. Depending on Disagreement Handling with veto or interaction, the LLM vetos with choosing the next action from all the valid actions or interacts with suggesting actions back to the RL (left). In agents with RL Decision-Maker, the RL chooses an action from the LLM’s suggestions, if at least one of them is among the top $\theta\%$ of its own action ranking and disagrees otherwise. In Disagreement Handling, the RL either vetos with selecting the next action from all the valid actions or interacts with re-prompting the LLM for new suggestions until they agree(right).

4.4. RL-LLM AGENTS THAT DISAGREE&VETO

In some occasions, the suggestions made by the Suggestion-Maker component might not correspond with the optimal actions determined by the other component leading to a Disagreement between them. The performance of RL-LLM Agents that Agree&Accept, as discussed in Section 4.3, may be influenced by potential shortcomings in the suggestions, given that they always adhere to the constraints set by the Suggestion-Maker.

We hence implemented a Disagreement Mechanism with Veto inside the agents called RL-LLM Agents that Disagree&Veto, enabling the Decision-Maker component to refuse choosing an action from the list proposed by the Suggestion-Maker and to take full control of decisions. The Disagreement Mechanism running at every step of interaction has two stages: Disagreement Indication and Disagreement Handling. During Disagreement Indication, the Suggestion-Maker’s partner indicates whether it disagrees or not. If it does not disagree, it will go ahead and choose among the suggestions of the Suggestion-Maker. In case of Disagreement, the agent moves to the next stage, Disagreement Handling, where the Decision-Maker takes full control and chooses from all available actions. Again, in these agents, the LLM-based and RL-based models switch roles between being a Decision-Maker that Disagree&Veto and a Suggestion-Maker to create two types of agents, RL-Suggests-LLM-Decides/Vetos ($RL \rightarrow LLM_{VETO}$) and LLM-Suggests-RL-Decides/Vetos ($LLM \rightarrow RL_{VETO}$).

4.4.1. *RL-Suggests-LLM-Decides/Vetos* ($RL \rightarrow LLM_{\text{VETO}}$).

As shown in Fig. 6 (left), for Disagreement Indication in these agents, we modified the prompt to inform the LLM about the option of Disagreement with RL suggestions. In case the LLM disagrees, it is handled by letting the LLM take full control in that step and choose among all available actions. This is done by prompting the LLM to choose from all valid actions, similar to the *LLM-based* agent.

4.4.2. *LLM-Suggests-RL-Decides/Vetos* ($LLM \rightarrow RL_{\text{VETO}}$).

As shown in Fig. 6 (right), a Disagreement is indicated in these agents if no action in LLM suggestions is among the top $\theta\%$ of actions in terms of their Q-values in RL model's perspective. θ is a hyper-parameter that needs to be optimized. Similar to the $RL \rightarrow LLM_{\text{VETO}}$ agent, a Disagreement is handled by letting the *RL-based* agent take full control in that step and choose among all available actions based on its Q-values.

4.5. RL-LLM AGENTS THAT DISAGREE&INTERACT

As previously discussed, each of the *RL-based* and *LLM-based* agents comes with its own shortcomings. As a result, even though Disagreement with vetoing can mitigate the *Suggestion-Maker's* shortcomings, it might exacerbate shortcomings of the *Decision-Maker*. Therefore, we introduce a new Disagreement Mechanism with interaction in the last version of agents, *RL-LLM agents that Disagree&Interact*, in which if a Disagreement happens the models can communicate new insights designed based on their strengths through interaction. This mechanism happens in the two stages of Disagreement Indication and Disagreement Handling.

Similar to *RL-LLM agents that Disagree&Veto* in Section 4.4, during Disagreement Indication, the *Suggestion-Maker's* partner indicates whether it disagrees or not. If it does not disagree, it chooses among the suggestions of the *Suggestion-Maker*. In case of Disagreement, the agent moves to the next stage, Disagreement Handling, where this time both models communicate new insights designed based on their strengths through interaction. The Disagreement Handling methods based on interaction are designed to ensure more human-like solutions by taking the *LLM-based* agent's suggestions and excelling in constrained environments by the *RL-based* agent taking the final decisions. Again, in these agents, the *LLM-based* and *RL-based* agents switch between being a *Decision-Maker that Disagree&Interact* and a *Suggestion-Maker* creating two types of agents, *RL-Suggests-LLM-Decides/Interacts* ($RL \rightarrow LLM_{\text{INTERACT}}$) and *LLM-Suggests-RL-Decides/Interacts* ($LLM \rightarrow RL_{\text{INTERACT}}$).

4.5.1. *RL-Suggests-LLM-Decides/Interacts* ($RL \rightarrow LLM_{\text{INTERACT}}$)

As shown in Fig. 6 (left), Disagreement Indication in these agents happens by modifying LLM's prompt similar to $RL \rightarrow LLM_{\text{VETO}}$ agents in Section 4.4. In case the LLM disagrees, it is handled by the LLM suggesting back several actions to the *RL-based* agent to choose from, leading to a reversal of the LLM and RL agent roles at this interaction step. For this, the LLM is prompted to suggest actions to the *RL-based* agent to choose from, similar to when it took the *Suggestion-Maker* role in the $LLM \rightarrow RL_{\text{ACCEPT}}$ agent.

4.5.2. *LLM-Suggests-RL-Decides/Interacts* ($LLM \rightarrow RL_{INTERACT}$)

As shown in Fig. 6 (right), a Disagreement is indicated in these agents by comparing LLM Suggestions to RL model's top picks similar to $LLM \rightarrow RL_{VETO}$ agents in Section 4.4. A Disagreement is handled by the *RL-based* agent re-prompting the LLM for new suggestions. For this, the LLM is prompted to suggest actions to the *RL-based* agent to choose from, similar to when it took the *Suggestion-Maker* role in the $LLM \rightarrow RL_{ACCEPT}$ agent. This process is done by re-prompting the LLM to suggest different actions from its previous suggestion for a maximum of five times or until an agreement is reached, i.e., at least one suggested action is in the top $\theta\%$ in terms of Q-values. If the maximum number of re-prompting does not lead to an agreement, the *RL-based* agent fully takes over for that step.

5. EXPERIMENTAL EVALUATION

We evaluated our agents on `PharmaSimText` to assess what type of agent performs best at conducting effective diagnostic conversations and correctly diagnosing patients (**RQ1**)? how does the specific LLM (GPT-4 or open-source LLMs) used in the LLM-involved agents affect agent performance (**RQ2**)? how does changing the power balance between the RL and LLM agents by introducing a disagreement mechanism in the strategy used for combining them influence the performance of the *RL-LLM* agents (**RQ3**)?. The different agent types based on GPT-4 without a disagreement mechanism and their resulting performance on `PharmaSimText` have been published in (Radmehr et al., 2024). Here, we restate and extend these results as a basis for our extension to open-source LLMs and new *RL-LLM* agents.

5.1. EXPERIMENTAL SETUP

Our evaluation focused on the agents' generalization capabilities, particularly their ability to navigate tasks with elements they had not encountered before.

5.1.1. Generalization Assessment

We assessed the agents based on their ability to generalize across rephrased versions of already encountered scenarios, to determine how much they relied on the precise wording of those scenarios. Figure 7 illustrates our evaluation methodology for generalization through an example of rephrased answer options creating diversity.

In `PharmaSimText`, an agent's success in a subtask is based on two factors: identifying the main cause of the patient's problem and asking *key questions* during the diagnostic conversation. Here, a subtask denotes the combination of a cause and a wording. Accordingly, we defined three metrics:

- *Diagnostic Accuracy Score*: Binary indicator of correct diagnosis of the patient's condition. This metric assesses how well an agent can identify the most probable cause of a patient's problem.
- *Conversational Completeness Score*: The percentage of key questions asked by the agent. This metric assesses how well an agent conversation with the patient covers all the key information needed for diagnosis.

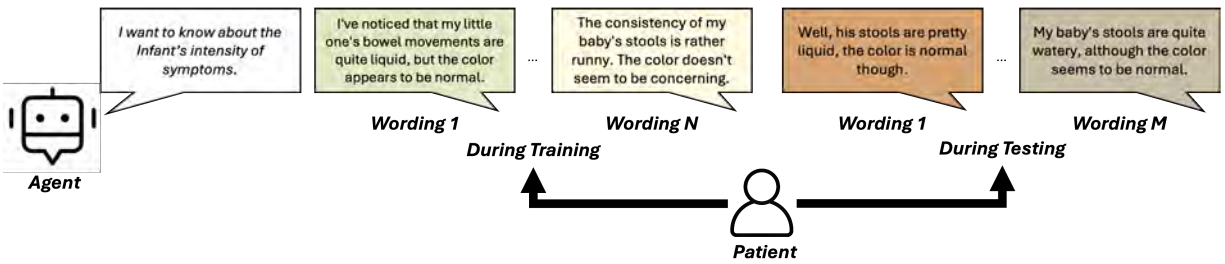


Figure 7: **Generalization task**, requiring the agent to generalize to a never-seen-before wording of a scenario.

- *Combined Score*: The multiplication of the *Diagnostic Accuracy Score* and *Conversational Completeness Score* metric. This metric combines both of the above elements.

5.1.2. Agent Training and Evaluation

We developed and trained all agents separately for each patient. As part of the process, different wordings of subtasks leading to the same diagnosis were randomly split into training, validation, and test sets. Therefore, all diagnoses applicable to a patient were included in the training, validation, and test sets in distinct wordings. As a result, the agents saw all diagnoses during training and validation, but not all wordings. We used 80% of the available wordings for each diagnosis for training, 10% for validation, and 10% for testing.

During training, every time the *RL-based* agent interacted with the environment, it was given a random task from the designated training set. At each time step t , the *RL-based* agent took an action sampled from a softmax policy derived from all available actions' Q-values. In the training process, a temperature decaying linearly from 1 to 0.001 controlled the randomness of the softmax policy. In each interaction, the agent was rewarded using a reward function that awarded it a positive reward of +1 when it succeeded in the posttest and penalized it with a negative reward of -1 when it failed. The agent also received a small negative penalty of -0.01 for each interaction it made.

The *LLM-based* agent types were trained and evaluated with the exact same train-validation-test split. These agents gain experience through interactions with the training subtasks, which they can then leverage on the test subtasks. For the *RL-LLM* agents with LLM Decision-Maker with disagreement mechanisms, the threshold θ was optimized based on the best *Combined Score* on the validation set. For the agent types involving LLMs, we experimented with seven different LLMs: GPT-4 (OpenAI, 2023); open-source models Llama3-70B (Dubey et al., 2024), Llama3-8B (Dubey et al., 2024), Gemma2-9B-it (Team, 2024), and Mixtral8x7B (Jiang et al., 2024); and medically fine-tuned open source models OpenBioLLM-70B (Ankit Pal, 2024) and OpenBioLLM-8B (Ankit Pal, 2024). In the following, we will replace the term LLM with the name of the specific LLM used in the *RL-LLM* and *LLM-based* agents.

5.2. RQ1: EFFICACY OF DIFFERENT AGENT TYPES BASED ON GPT-4

In the first analysis, we assessed agents' *Diagnostic Accuracy Score* and *Conversational Completeness Score*. Figure 8 illustrates the *Diagnostic Accuracy Score*, *Conversational Completeness Score*, and *Combined Score* of the *RL-based*, $RL \rightarrow GPT-4_{ACCEPT}$, $GPT-4 \rightarrow RL_{ACCEPT}$, and

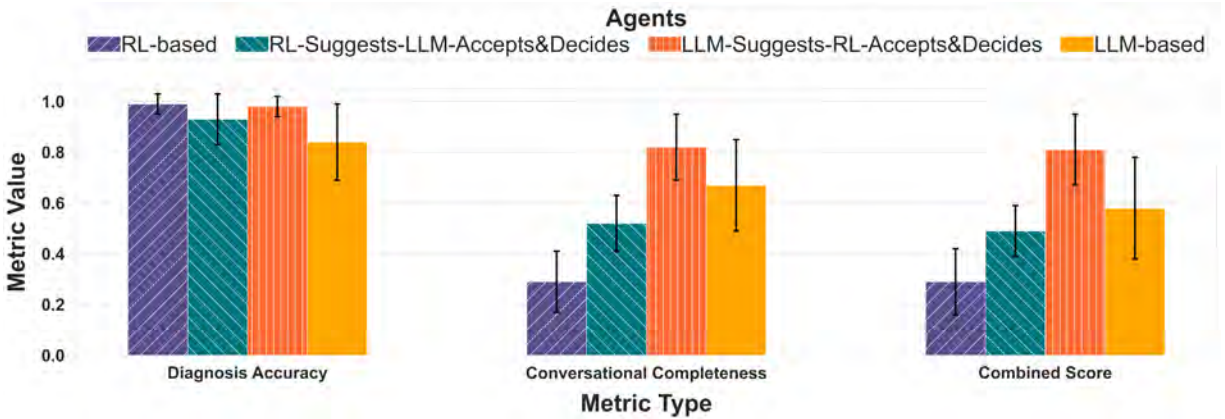


Figure 8: **RQ1: Agent Performance on PharmaSimText.** *Diagnostic Accuracy Score* (left), *Conversational Completeness Score* (middle), and *Combined Score* (right) of the *RL-based*, $RL \rightarrow GPT-4_{ACCEPT}$, $GPT-4 \rightarrow RL_{ACCEPT}$, and *GPT-4-based* agents. In the $RL \rightarrow GPT-4_{ACCEPT}$ agent, $GPT-4$ selects an action from the top- k choices provided by the *RL-based* agent. In the $GPT-4 \rightarrow RL_{ACCEPT}$ agent, $GPT-4$ suggests k actions at each step for the *RL-based* agent to choose from. Scores are averaged across all patients in PharmaSimText.

GPT-4-based agents aggregated over all patients.

We observed that the *RL-based* agent achieved a high *Diagnostic Accuracy Score* (0.99), indicating its ability to learn the correct diagnoses for different scenarios. However, this agent’s approach often lacked the depth and nuance of a meaningful diagnostic conversation, reflected in its low *Conversational Completeness Score* (0.29). This observation is probably due to its lack of background knowledge and common sense reasoning. In contrast, the *GPT-4-based* agent exhibited a superior ability for engaging in meaningful diagnostic dialogues, reflected in a higher *Conversational Completeness Score* (0.67). However, the *GPT-4-based* agent exhibited a lower *Diagnostic Accuracy Score* (0.84) than the *RL-based* agent, indicating that its ability to consistently reach the correct diagnosis was inferior compared to the *RL-based* agent.

Both *RL-LLM* agents surpassed the *GPT-4-based* agent in *Diagnostic Accuracy Score*, indicating that integrating LLMs with RL generally improves the diagnostic precision of pure *LLM-based* agents. Notably, the $GPT-4 \rightarrow RL_{ACCEPT}$ agent exhibited superior *Diagnostic Accuracy Score* (0.98) closely mirroring that of the *RL-based* agent. The $RL \rightarrow GPT-4_{ACCEPT}$ ’s relative under-performance (*Diagnostic Accuracy Score* of 0.93) might be due to its longer trajectories compared to the *RL-based* agent, leading to unfamiliar states, where the DRRN struggled to provide accurate diagnoses. Furthermore, in terms of engaging in quality diagnostic dialogues, the $GPT-4 \rightarrow RL_{ACCEPT}$ agent (0.82) outperformed the $RL \rightarrow GPT-4_{ACCEPT}$ agent (0.52). This observation is likely due to the RL framework’s preference for shorter, more direct solutions, reducing the action quality suggested by the *RL-based* agent in prolonged interactions.

The $GPT-4 \rightarrow RL_{ACCEPT}$ agent further outperformed all other agents in terms of *Combined Score*. It demonstrated both a high *Conversational Completeness Score* and *Diagnostic Accuracy Score*. This achievement highlights the $GPT-4 \rightarrow RL_{ACCEPT}$ 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 indicated significant differences between the agents for the *Conversational Completeness Score* and *Combined Score* ($p_{\text{Conversational Completeness}} < .0001$ and $p_{\text{Combined}} < .001$) and a trend to significance for the *Diagnostic Accuracy Score* ($p_{\text{Diagnostic Accuracy}} = .052$). Pairwise post-hoc comparisons using Mann-Whitney U tests with a Benjamini-Hochberg correction for the *Combined Score* indicate significant differences between five out of six pairs of agents, with the only non-significant comparison being between the *GPT-4-based* and $\text{RL} \rightarrow \text{GPT-4}_{\text{ACCEPT}}$ agent. For the *Conversational Completeness Score*, 4 out of 6 pairwise comparisons were significant, with non-significant comparisons being between the *GPT-4-based* and $\text{RL} \rightarrow \text{GPT-4}_{\text{ACCEPT}}$ agent as well as *GPT-4-based* and $\text{GPT-4} \rightarrow \text{RL}_{\text{ACCEPT}}$ agent.

RL-based agents achieve a high diagnostic accuracy, but demonstrate limited *Conversational Completeness Score*. *LLM-based* agents excel in conducting high-quality conversations, but show less accuracy in diagnosing. *RL-LLM* agents combine the best of both worlds as demonstrated by the $\text{LLM} \rightarrow \text{RL}_{\text{ACCEPT}}$ agent achieving the highest *Combined Score*.

5.3. RQ2: PERFORMANCE OF BASE OR MEDICALLY FINE-TUNED OPEN-SOURCE LLMs

In our second analysis, we investigated the impact of the specific LLM used on the performance of the *RL-LLM* and *LLM-based* agents. We only report the performance of GPT-4, and the best open-source general (Llama3-70B) and medically fine-tuned LLM (OpenBioLLM-70B). The results of all other models can be found in Tables 7, 8, and 9 in the Appendix 8.6.1. Figure 9 illustrates the *Diagnostic Accuracy Score* and *Conversational Completeness Score* for the *RL-based* agents as well as the *RL-LLM*, and *LLM-based* agents using different LLMs.

Not unexpectedly, for the *LLM-based* agents, *GPT-4-based* agent demonstrated a higher *Diagnostic Accuracy Score* than the open-source models (*GPT-4-based*: 0.84, *Llama3-70B-based*: 0.7, *OpenBioLLM-70B-based*: 0.65). These differences are indeed significant ($p = 0.02^*$)³. In terms of *Conversational Completeness Score*, the *GPT-4-based* and *Llama3-70B-based* agents show similar performance (*GPT-4-based*: 0.67, *Llama3-70B-based*: 0.65). Notably, the medically fine-tuned OpenBioLLM-70B, which we expected to capture more key questions due to its medical knowledge, performs significantly worse (0.4, $p = .02^*$). However, upon closer inspection, we noticed that the *OpenBioLLM-70B-based* agent tended to repeat the questions that were already asked despite the prompt stating that repeating questions would not lead to new information.

Among the *RL-Suggests-LLM-Accepts&Decides* ($\text{RL} \rightarrow \text{LLM}_{\text{ACCEPT}}$) agents, both of the $\text{RL} \rightarrow \text{Llama3}_{\text{ACCEPT}}$ and $\text{RL} \rightarrow \text{OpenBio}_{\text{ACCEPT}}$ agents exhibit an almost perfect *Diagnostic Accuracy Score* ($\text{RL} \rightarrow \text{GPT-4}_{\text{ACCEPT}}$: 1.0, $\text{RL} \rightarrow \text{Llama3}_{\text{ACCEPT}}$: 0.98), with the $\text{RL} \rightarrow \text{GPT-4}_{\text{ACCEPT}}$ showing a slightly lower score (0.93). These differences are significant ($p = .04^*$). For the *Conversational Completeness Score*, we found similar results as for the *LLM-based* agents, with the $\text{RL} \rightarrow \text{OpenBio}_{\text{ACCEPT}}$ agent performing significantly worse than the two other agents (GPT-4: 0.52, Llama3: 0.49, OpenBio: 0.3, $p = .01^*$). This observation might be due to the tendency of OpenBioLLM-70B to choose the first suggestion offered by the *RL-based* agent resulting in a similar diagnostic conversation to the *RL-based* agent's conversation.

For the *LLM-Suggests-RL-Accepts&Decides* ($\text{LLM} \rightarrow \text{RL}_{\text{ACCEPT}}$) agents, the $\text{GPT-4} \rightarrow \text{RL}_{\text{ACCEPT}}$ agent outperforms the other agents both in terms of *Diagnostic Accuracy Score* (GPT-

³In the following, all reported p -values are indicated by Kruskal-Wallis tests.

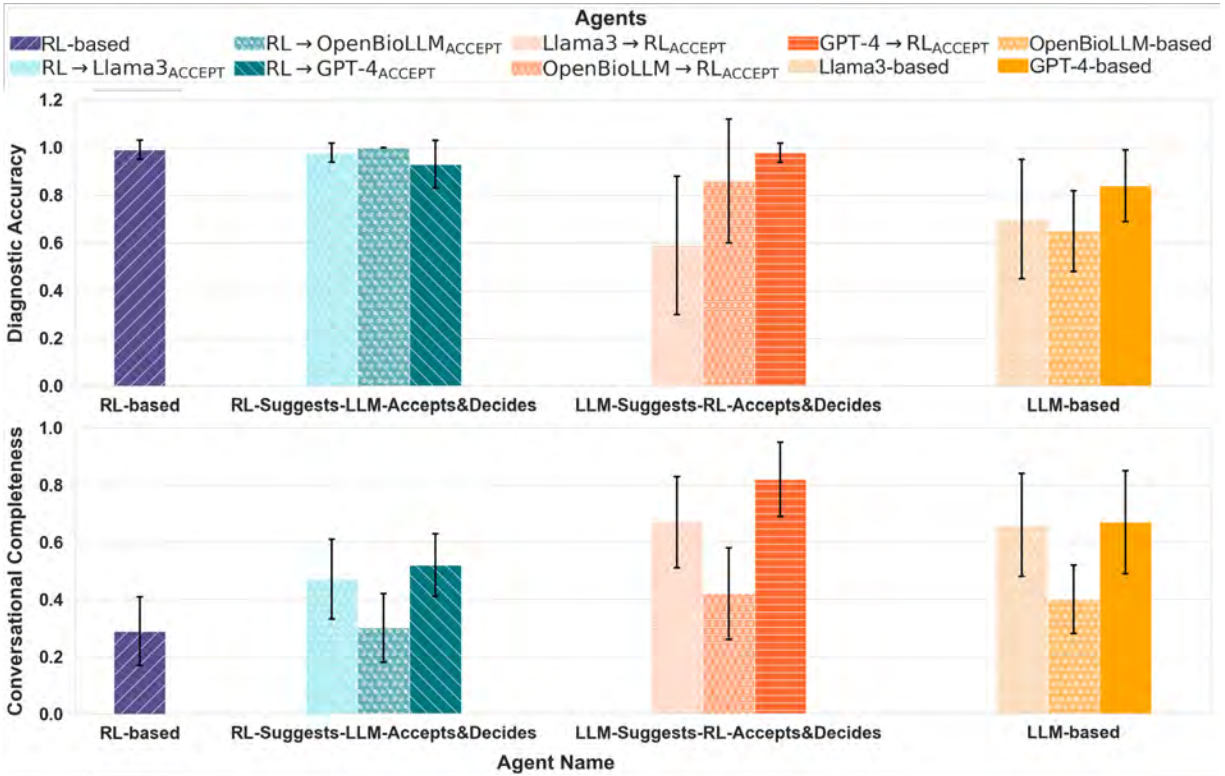


Figure 9: **RQ2: Comparison of agents' performance using different LLMs.** *Diagnostic Accuracy Score* (top) and *Conversational Completeness Score* (bottom) of the *RL-based*, as well as the $RL \rightarrow LLM_{ACCEPT}$, the $LLM \rightarrow RL_{ACCEPT}$, and the *LLM-based* agents using Llama3-70B, OpenBioLLM-70B, or GPT-4. Scores are averaged across all patients in PharmaSimText.

4: 0.98, Llama3: 0.81, OpenBio: 0.82) and *Conversational Completeness Score* (GPT-4: 0.82, Llama3: 0.7, OpenBio: 0.48). For *Diagnostic Accuracy Score* the differences are not significant ($p = .14$) despite the $GPT-4 \rightarrow RL_{ACCEPT}$ outperforming the $Llama3 \rightarrow RL_{ACCEPT}$ and $OpenBio \rightarrow RL_{ACCEPT}$. In contrast, we find significant differences between the agents for *Conversational Completeness Score*, with the $OpenBio \rightarrow RL_{ACCEPT}$ performing significantly worse in this regard ($p = .007^*$). For the $Llama3 \rightarrow RL_{ACCEPT}$ agent, a detailed inspection revealed that, although the *Llama3-70B-based* agent did include moving to the post-test among its suggestions, when the *RL-based* agent decided to choose that action, the *Llama3-70B-based* agent persisted in suggesting asking the patient more questions, leading to random suggestions for the *RL-based* agent to choose from.

Although the open-source models Llama3-70B and OpenBioLLM-70B can compete in some aspects with GPT-4 when used as the LLM component of the $RL \rightarrow LLM_{ACCEPT}$ and *LLM-based* agents, they cannot reach the performance of the $GPT-4 \rightarrow RL_{ACCEPT}$ agent. Further studies are needed to ease the use of open-source models for developing high performing agents.

5.4. RQ3: EFFECT OF DISAGREEMENT MECHANISM

In a final analysis, we examined effect of different disagreement mechanisms on agents' performance. We again only report the performance of GPT-4, and the best open-source gen-

Table 2: **RQ3: Performance of agents with LLM Decision-Maker with different disagreement mechanisms.** *Diagnostic Accuracy Score* and *Conversational Completeness Score* of $RL \rightarrow LLM_{ACCEPT}$, $RL \rightarrow LLM_{VETO}$, and $RL \rightarrow LLM_{INTERACT}$ agents using GPT-4, Llama3-70B, or OpenBioLLM-70B. In disagreement with veto the LLM decides itself on the next action from all the valid actions and in disagreement with interaction the LLM suggests actions back to the RL to choose from. Results are mean \pm standard deviation across all patients in the PharmaSimText.

Agent	<i>Diagnostic Accuracy Score</i>			<i>Conversational Completeness Score</i>		
	Llama3	OpenBio	GPT-4	Llama3	OpenBio	GPT-4
$RL \rightarrow LLM_{ACCEPT}$	0.98\pm0.04	1.00\pm0.00	0.93 \pm 0.10	0.47 \pm 0.14	0.30 \pm 0.12	0.52 \pm 0.11
$RL \rightarrow LLM_{VETO}$	0.91 \pm 0.13	0.97 \pm 0.09	0.96 \pm 0.09	0.46 \pm 0.14	0.33 \pm 0.17	0.59 \pm 0.19
$RL \rightarrow LLM_{INTERACT}$	0.92 \pm 0.18	0.85 \pm 0.20	0.92 \pm 0.11	0.52 \pm 0.23	0.61\pm0.20	0.68 \pm 0.17

Table 3: **RQ3: Performance of agents with RL Decision-Maker with different disagreement mechanisms** *Diagnostic Accuracy Score* and *Conversational Completeness Score* of $LLM \rightarrow RL_{ACCEPT}$, $LLM \rightarrow RL_{VETO}$, and $LLM \rightarrow RL_{INTERACT}$ agents using GPT-4, Llama3-70B, or OpenBioLLM-70B. In disagreement with veto the RL decides itself on the next action from all the valid actions and in disagreement with interaction the *RL-based* agent re-prompts the LLM for new suggestions until they agree. Results are mean \pm standard deviation across all patients in the PharmaSimText.

Agent	<i>Diagnostic Accuracy Score</i>			<i>Conversational Completeness Score</i>		
	Llama3	OpenBio	GPT-4	Llama3	OpenBio	GPT-4
$LLM \rightarrow RL_{ACCEPT}$	0.59 \pm 0.29	0.86 \pm 0.26	0.98 \pm 0.04	0.67 \pm 0.16	0.42 \pm 0.16	0.82 \pm 0.13
$LLM \rightarrow RL_{VETO}$	0.94\pm0.12	0.95 \pm 0.08	0.96 \pm 0.07	0.67 \pm 0.18	0.39 \pm 0.17	0.71 \pm 0.13
$LLM \rightarrow RL_{INTERACT}$	0.93\pm0.10	0.99\pm0.03	0.99 \pm 0.04	0.75 \pm 0.14	0.59\pm0.14	0.76 \pm 0.09

eral (Llama3-70B) and medically fine-tuned LLM (OpenBioLLM-70B). The results of all other models can be found in Tables 10 and 11 in the Appendix 8.6.2.

Agents with LLM Decision-Maker. Table 2 shows the *Diagnostic Accuracy Score* and *Conversational Completeness Score* for the *RL-Suggests-LLM-Accepts&Decides* ($RL \rightarrow LLM_{ACCEPT}$), *RL-Suggests-LLM-Decides/Vetos* ($RL \rightarrow LLM_{VETO}$), and *RL-Suggests-LLM-Decides/Interacts* ($RL \rightarrow LLM_{INTERACT}$) agents. As expected, the $RL \rightarrow LLM_{INTERACT}$ agent with the interaction-based disagreement mechanism has the highest *Conversational Completeness Score* for all different LLM types while preserving the *Diagnostic Accuracy Score*. As detailed in Section 5.3, the $RL \rightarrow LLM_{ACCEPT}$ agent demonstrated a high *Diagnostic Accuracy Score*, but a low *Conversational Completeness Score* without a disagreement mechanism. Therefore, giving all the power to the *RL-based* agent in the $RL \rightarrow LLM_{VETO}$ agent does not improve *Conversational Completeness Score*. Combining the strengths of *RL-based* and *LLM-based* agents in the disagreement mechanism based on interaction improved *Conversational Completeness Score* without degrading *Diagnostic Accuracy Score*, leading to a better *Combined Score*. A Kruskal-Wallis test indicated significant differences between the agents using OpenBioLLM-70B with different disagreement mechanisms for *Diagnostic Accuracy Score* and *Conversational Completeness Score* (GPT-4: $p = .82$, Llama3-70B: $p = .52$, OpenBioLLM-70B: $p = .045^*$ for

Diagnostic Accuracy Score, and GPT-4: $p = .21$, Llama3-70B: $p = .93$, OpenBioLLM-70B: $p = .022^*$ for *Conversational Completeness Score*).

Agents with RL Decision-Maker. Table 3 includes the *Diagnostic Accuracy Score* and *Conversational Completeness Score* for the *LLM-Suggests-RL-Accepts&Decides* ($\text{LLM} \rightarrow \text{RL}_{\text{ACCEPT}}$), *LLM-Suggests-RL-Decides/Vetos* ($\text{LLM} \rightarrow \text{RL}_{\text{VETO}}$), and *LLM-Suggests-RL-Decides/Interacts* ($\text{LLM} \rightarrow \text{RL}_{\text{INTERACT}}$) agents. Although none of the disagreement mechanisms improved the performance of agents with GPT-4, the disagreement mechanism with interaction substantially improved the *Diagnostic Accuracy Score* and *Conversational Completeness Score* of Llama3 $\rightarrow \text{RL}_{\text{INTERACT}}$ and OpenBio $\rightarrow \text{RL}_{\text{INTERACT}}$ agents compared to Llama3 $\rightarrow \text{RL}_{\text{ACCEPT}}$ and OpenBio $\rightarrow \text{RL}_{\text{ACCEPT}}$ agents. As detailed in Section 5.3, the Llama3 $\rightarrow \text{RL}_{\text{ACCEPT}}$ and OpenBio $\rightarrow \text{RL}_{\text{ACCEPT}}$ agents struggled with both the *Diagnostic Accuracy Score* and the *Conversational Completeness Score*. Disagreement with interaction not only lets the *RL-based* agent guide the LLM in the post-test, leading to a higher *Diagnostic Accuracy Score*, but also improves the *Conversational Completeness Score* by allowing the LLM to reflect on its suggestions in the case of disagreement. Interestingly, the optimal threshold θ for triggering a disagreement is lower for the OpenBio $\rightarrow \text{RL}_{\text{ACCEPT}}$ agent compared to the other two agents, implying less trust from the *RL-based* agent to suggestions of this LLM ($\theta_{\text{OpenBio-best}} = 0.36$, $\theta_{\text{Llama3-best}} = 0.56$, and $\theta_{\text{GPT-4-best}} = 0.56$). A Kruskal-Wallis test indicated significant differences in *Diagnostic Accuracy Score* between agents with Llama3-70B and in *Conversational Completeness Score* between agents with OpenBioLLM-70B (For *Diagnostic Accuracy Score*, GPT-4: $p = .72$, Llama3-70B: $p = .01^*$, OpenBioLLM-70B: $p = .34$ and for *Conversational Completeness Score*, GPT-4: $p = .21$, Llama3-70B: $p = .44$, OpenBioLLM-70B: $p = .03^*$).

While none of the disagreement mechanisms improved the performance of GPT-4-based *RL-LLM* agents, the disagreement mechanism based on interaction helped the performance of the agents based on open-source LLMs the most. This shows the potential offered by a well-designed disagreement mechanism for further improvement of agents based on open-source LLMs.

6. DISCUSSION AND CONCLUSION

In this paper, we explored the integration of RL and LLMs to improve simulated learners in educational technologies. We developed generalizable agents for text-based educational settings, including *RL-based*, *LLM-based*, and a wide range of *RL-LLM* agents. We evaluated them using our novel benchmark PharmaSimText.

We were interested in answering the following three research questions: What type of agent performs best at conducting effective diagnostic conversations and correctly diagnosing patients (**RQ1**)? How does the specific LLM (GPT-4 or open-source LLMs) used in the LLM-involved agents affect agent performance (**RQ2**)? How does changing the power balance between the RL and LLM agents by introducing a disagreement mechanism in the strategy used for combining them influence the performance of the *RL-LLM* agents (**RQ3**)?

To address our first research question, we assessed four agents: an *RL-based* agent, an *LLM-based* agent based on GPT-4, and two *RL-LLM* agents integrating GPT-4 with RL. The *RL-based* agent excelled in finding the correct diagnosis, but struggled in comprehensive diagnostic dialogues due to its limited knowledge. The *LLM-based* agent was adept in high-quality diagnostic conversations, but tended to misdiagnose patients. The *RL-LLM* agents were able to address

these limitations, with the $GPT-4 \rightarrow RL_{ACCEPT}$ agent achieving the best combination of *Diagnostic Accuracy Score* and *Conversational Completeness Score*.

To answer our second research question, we compared performance of agent types using GPT-4, a general-purpose open-source LLM (Llama3-70B) and a medically fine-tuned LLM (OpenBioLLM-70B). While Llama3-70B and OpenBioLLM-70B showed potential in *RL-LLM* agents, they could not match $GPT-4 \rightarrow RL_{ACCEPT}$'s performance. Further research is needed to facilitate the use of open-source models for the development of high-performing agents.

To address the last research question, we compared *RL-LLM* agents using different disagreement mechanisms. We tested three versions of *RL-LLM* agents that either agree with accepting, disagree with vetoing, or disagree with interacting. Our results showed that the disagreement mechanism's effectiveness depends on the handling mechanism and the LLM. While disagreement did not improve the performance of GPT-4-based *RL-LLM* agents, disagreement with interaction improved the performance of $RL \rightarrow OpenBio_{ACCEPT}$, $OpenBio \rightarrow RL_{ACCEPT}$, and $Llama3 \rightarrow RL_{ACCEPT}$ agents. This shows the potential of the disagreement mechanism for further improvement of agents based on open-source LLMs.

One of the limitations of this work is the focus on generalization at a single level of rephrased versions of the scenarios. Further research should be done within different generalization levels of *PharmaSimText*: generalizing to a new diagnosis of a known patient (subtask generalization) and to a new patient (task generalization). In generalizing to new subtasks, the agent should ideally transfer already known information about the patient in the seen subtasks and identify the contrasting information inside the new subtask that leads to a new diagnosis. However, during our initial experiments in generalization to new subtasks, we observed that the *RL-based* agent shows a bias towards the already-seen diagnoses which is in agreement with prior research (e.g., [Nikishin et al. 2022](#)). This observation implies that at this level, promising methods may include considering the out-of-distribution detection of the *RL-based* agent in the integration strategy to achieve zero-shot generalization and developing a novel LLM-guided meta Reinforcement Learning method for few-shot generalization. As for generalization to new tasks, the inquiry skills learned for the seen patient should be used for the new patient. Possibly the most promising direction for this end is to let the LLM guide an *RL-based* agent that masters different skills rather than just one reward at this highest level of generalization.

The presented agents involving an LLM component are prone to carry LLM biases in their solutions including the inquiry strategy and diagnosis. Although combination with *RL-based* agents in *RL-LLM* agents mitigated this to some extent, it still cannot be overlooked. To ensure fairness in utilizing them as simulated learners, future research should assess how closely these agents' behaviors mirror that of various human students' strategies in order to determine their suitability as a basis for interventions. Additionally, as these agents are designed to be successful learners, further research is required to further align their behavior with struggling students for a more fair representation of a student population. For broader applications, an additional step for studying of biases in LLM decisions is essential. Even though the presented hybrid agents require much less computational costs compared to fine-tuning LLMs for the task at hand, in the long run considering finetuning the RL agent with the new insights or using small LMs is more sustainable.

Our proposed LLM integration approach represents a promising step towards generalizable agents not only in less-structured text-based educational environments but also in settings requiring a more human-like approach within certain constraints. A wide range of such settings exist in the real world, from the development of agents that assist humans in playing games to

autonomous driving or rescue teams. The pairing of an RL agent trained inside a constrained environment with an LLM offers a new perspective on grounding LLMs in constraints. Moreover, our novel benchmark further facilitates research regarding developing generalizable agents for text-based educational environments and beyond.

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).

8. APPENDICES

In this section, we present prompts for generation of `PharmaSimText` scenarios, architecture of the *RL-based* agent, and extended results for more open-source LLMs.

8.1. PROMPTS FOR SCENARIO GENERATION

An example of the prompts used in the three steps of scenario generation as detailed in Sec. 3 for the Urological problems scenario.

8.1.1. Step1: Key Questions Generation

As seen in the following prompt, in this step the LLM is prompted to generate a list of key questions.

Prompt example in Step 1 for the Urological problems scenario

System: You are a pharmacist consulting me in designing an educational game for students studying to become a pharmacist. In this game they practice finding probable causes of patients problem by conducting diagnostic conversations. I'm creating new scenarios for the game where an imaginary patient walks into the pharmacy with a problem. I'll also give you a list of possible illnesses that can be relevant to that problem and their key symptoms.

Prompt: I want to add a scenario with a 56-year-old male patient who has urological problems. List of illnesses and their symptoms:

1. Prostate Hyperplasia:

- Urinary frequency
- Urinary urgency
- Trouble starting a urine stream
- Weak or interrupted urine stream
- Dribbling at the end of urination
- Nocturia (frequent urination during sleep)
- Urinary retention
- Urinary incontinence
- Pain after ejaculation or during urination

- Unusual color or smell of urine

2. Cystitis (Bladder Infection):

- Frequent urination
- Strong urge to urinate (urgency)
- Burning or tingling sensation during or after urination (dysuria)
- Low-grade fever
- Cloudy urine with a strong odor
- Blood in urine (hematuria)
- Trouble urinating, especially if you have a problem with your prostate

3. Urge Incontinence:

- Urgent, uncontrollable need to urinate
- Possible urine leakage
- Frequent urination (more than eight times a day)
- Nocturia (peeing more than twice at night)
- Leaking urine before reaching the bathroom or just after urination
- Wetting the bed (enuresis) at night

4. Stress Incontinence:

- Urinary leakage during physical stress (e.g., coughing, sneezing)
- Increased intra-abdominal pressure
- Weak pelvic floor muscles

Similarity in symptoms:

- **Frequent urination** is a common symptom in Prostate Hyperplasia, Cystitis, and Urge Incontinence.
- **Urinary urgency** is present in Prostate Hyperplasia and Urge Incontinence.
- **Trouble urinating** is a symptom in Prostate Hyperplasia and may occur in cases of Cystitis.
- **Urinary incontinence** is seen in Prostate Hyperplasia and Urge Incontinence.

Task: What are the key questions that the students need to ask to find out the cause behind the patient's problem?

Keep this list as short as possible but at the same time, one question should not be too general. Choose a title that reflects the content of the question well.

8.1.2. Step2: Simulation of patient's response to questions

As seen in the following prompts, this step is done in two parts. The LLM answers Key Questions in the first prompt and some common inquiries are answered using the next prompt.

Prompt example in Step 2.1 for the Urological problems scenario

This prompt is appended to the previous prompt and its answer

Task: Assume a patient has each of the illnesses on the list, how would they answer these questions?

Phrase it with wordings that a patient uses. If two different problems can lead to the same answer please use the same answer. Give a separate answer for a patient experiencing one of the illnesses on the list. Don't use the illness name in the answer. You should include an answer for all of the illnesses don't miss any.

Your answer should follow this template:

The response will be two .csv files. The first file is a table with two columns, "response_id" and "response_text". The response id should be a number. Each row is a unique response. The second table has one column for each illness and one row for each question. The first column has the title of the questions. The content of each cell is the response ID for the response to the column illness if the patient was asked the row question. Each cell in the second table should have exactly one response ID, however, multiple cells in a row can (and should, when applicable) have the same response ID. There must be at least 2 repetitive response ids in a row. Provide the csvs in a csv text file format.

Prompt example in Step 2.2 for the Urological problems scenario

This prompt is appended to the previous prompt and its answer

Task: Assume a patient has each of the illnesses on the list, how would they answer these questions?

Phrase it with wordings that a patient uses. If two different problems can lead to the same answer please use the same answer. Give a separate answer for a patient experiencing one of the illnesses on the list. Don't use the illness name in the answer. You should include an answer for all of the illnesses don't miss any.

Your answer should follow this template:

The response will be two .csv files. The first file is a table with two columns, "response_id" and "response_text". The response id should be a number. Each row is a unique response. The second table has one column for each illness and one row for each question. The first column has the title of the questions. The content of each cell is the response ID for the response to the column illness if the patient was asked the row question. Each cell in the second table should have exactly one response ID, however, multiple cells in a row can (and should, when applicable) have the same response ID. There must be at least 2 repetitive response ids in a row. Provide the csvs in a csv text file format.

Questions: Symptoms Localization: "Where are you feeling these symptoms specifically?"

Symptoms Intensity: "how severe is your symptoms?"

Duration of Symptoms: "How long have you been experiencing these symptoms?"

Medication History: "Have you taken any medications recently for these or other issues?"

Allergies: "Do you have any known allergies, especially to medications?"

Underlying Medical Conditions: "Do you have any ongoing health conditions?"

Current Medications: “What medications are you currently taking?”
Surgical History: “Have you had any surgeries in the past?”
Pregnancy and Breastfeeding: “Are you currently pregnant or breastfeeding?”
Age: “Can you tell me how old you are?”
Sleep: “How has your sleep been lately?”
Diet: “Could you describe your usual diet?”
Exercise: “What does your typical exercise routine look like?”
Teeth: “Have you had any recent issues with your teeth or dental health?”

8.1.3. Step3: Rephrasing responses

As seen in the following prompt, in this step all of the sentences generated in the last step is rephrased in 10 different ways.

Prompt example in Step 3 for the Urological problems scenario

For more details, please refer to the `rewrite_sentence` function in the `scenario_helper.py` file.

System: You are a customer consulting with a pharmacist about your urological problems.

User: Task: try to articulate the sentences I give you in 9 different ways. Make sure you are using this template: [“s1”, “s2”, “s3”, “s4”, “s5”, “s6”, “s7”, “s8”, “s9”]

Sentence: “I’ve been going to the bathroom a lot more than usual, both during the day and at night. Sometimes, I feel like I need to go urgently, but then I have trouble starting.”

Assistant: [“I’ve noticed an increased frequency of bathroom visits, both in the daytime and at night. Occasionally, I experience a strong urge to go, but initiating the process becomes challenging.”,

“Lately, I’ve been using the restroom more frequently, not only during the day but also at night. There are moments when I feel a sudden need to go, yet I encounter difficulties in commencing the process.”,

“I’ve found myself visiting the bathroom more often than usual, both in the daytime and throughout the night. On occasion, I experience a sense of urgency, but initiating the process becomes troublesome.”,

“My bathroom trips have become more frequent recently, occurring not only during the day but also during the night. Sometimes, I feel an urgent need to go, but initiating the process becomes problematic.”,

“I’ve been making more trips to the bathroom than usual, both during the day and at night. There are instances when I feel an immediate need to go, but initiating the process becomes troublesome.”,

“Recently, I’ve had to use the restroom more often, both during the day and at night. There are moments when I feel an urgent need to go, but commencing the process becomes challenging.”,

“I’ve observed a heightened frequency of bathroom visits, both during the day and at night. Occasionally, I experience a strong urge to go, but initiating the process becomes

Table 4: **Rubric for scenario evaluation.** The indicated metrics measure the correctness of the medical facts, the diagnosis, as well as the quality of the generated text.

Dimension	Range	Description
<i>Correctness</i>	[0, 1]	Ratio of medically correct answers to all answers of a scenario. The correctness of each answer is indicated by a yes/no answer from the pharmacy expert.
<i>Realism</i>	[0, 1]	Ratio of realistic patient answers to all answers of a scenario. The realism of each answer is indicated by a yes/no answer from the pharmacy expert.
<i>Completeness</i>	[0, 1]	Ratio of LLM-suggested questions to all required questions. The number of required questions is the sum of missing questions (indicated by the pharmacy expert) for an accurate diagnosis and LLM-suggested questions.
<i>Conceal</i>	[0, 1]	Ratio of causes for which the diagnosis is not directly written in any patient answers. Binary annotation per cause, where 1 indicates correctly concealed.
<i>Clarity</i>	[0, 1]	Ratio of comprehensible answers to all patient answers. The comprehensibility of each answer is coded with a binary annotation, where 1 indicates an answer is easy to read and not wordy.
<i>Consistency</i>	[0, 1]	Ratio of causes for which all answers are consistent to all causes. Binary annotation per cause, where 1 indicates no contradiction.

Table 5: **Statistics of PharmaSimText Scenarios.** The obtained benchmark contains seven distinct scenarios, each centered on a unique health problem, which could stem from various 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
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

difficult.”,

“Lately, I’ve been visiting the bathroom more frequently, not only during the day but also at night. There are times when I feel a sudden need to go, but initiating the process becomes problematic.”,

“My restroom trips have become more frequent recently, taking place not only during the day but also at night. Sometimes, I feel an immediate need to go, but starting the process becomes challenging.”]

User: “Sentence: new sentence”

8.2. EVALUATION RUBRIC FOR LLM-GENERATED SCENARIOS

Implementation details of the rubric for evaluating LLM-generated scenarios can be found in Table 4.

8.3. STATISTICS ON PHARMASIMTEXT SCENARIOS

Details about PharmaSimText scenarios can be found in Table 5.

8.4. THE *RL-based* AGENT ARCHITECTURE

Details of the architecture used in the DRRN of the *RL-based* agent can be found in Table 6.

Table 6: Parameters of DRRN in the *RL-based* agent

Name	Value
num layers action	1
num layers state	1
num layers scorer	1
hidden dim action	64
hidden dim state	512
hidden dim scorer	512
γ	0.9
batch size	256
gradient clip	100

8.5. EFFECT OF REFLECTIVE PROMPTING

In an extended analysis, we explored the impact of reflective prompting on the performance of the *RL-LLM* and *LLM-based* agents. Again, we only report the performance of GPT-4, and the best open-source general (Llama3-70B) and medically fine-tuned LLM (OpenBioLLM-70B). The results of all other models can be found in Tables 7, 8, and 9 in the Appendix 8.6.3.

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 between attempts. Figure 10 illustrates the *Diagnostic Accuracy Score* and *Conversational Completeness Score* for none-reflective and reflective *RL-LLM* and *LLM-based* agents when choosing either GPT-4, Llama3-70B, or OpenBioLLM-70B as their LLM component.

We observed a nuanced impact of reflective prompting on agents' performance for different LLMs and agent types. For the *LLM-based* agents, reflective prompting did not significantly impact the *Diagnostic Accuracy Score* ($p_{Llama3-70B} = .87$, $p_{GPT-4} = .96$) and *Conversational Completeness Score* ($p_{Llama3-70B} = .71$, $p_{GPT-4} = .67$) of the *GPT-4-based* and *Llama3-70B-based* agents. For these agents, 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. In contrast, the reflection significantly improved the *Diagnostic Accuracy Score* ($p_{OpenBioLLM-70B} = .04^*$) without significantly changing *Conversational Completeness Score* ($p_{OpenBioLLM-70B} = .46$) of OpenBioLLM-70B.

For the $RL \rightarrow LLM_{ACCEPT}$ agents, reflection increased the *Diagnostic Accuracy Score* of all agents. However, differences were not significant ($p_{Llama3} = .14$, $p_{GPT-4} = .56$, $p_{OpenBio} = .06$). Reflection also increased the *Conversational Completeness Score* for the $RL \rightarrow Llama3_{ACCEPT}$ ($p_{Llama3} = .67$) and $RL \rightarrow GPT-4_{ACCEPT}$ ($p_{GPT-4} = .63$) agents. However, the *Conversational Completeness Score* slightly dropped for the $RL \rightarrow OpenBio_{ACCEPT}$ agent ($p_{OpenBio} = 0.53$) due to shorter trajectories of the reflective version.

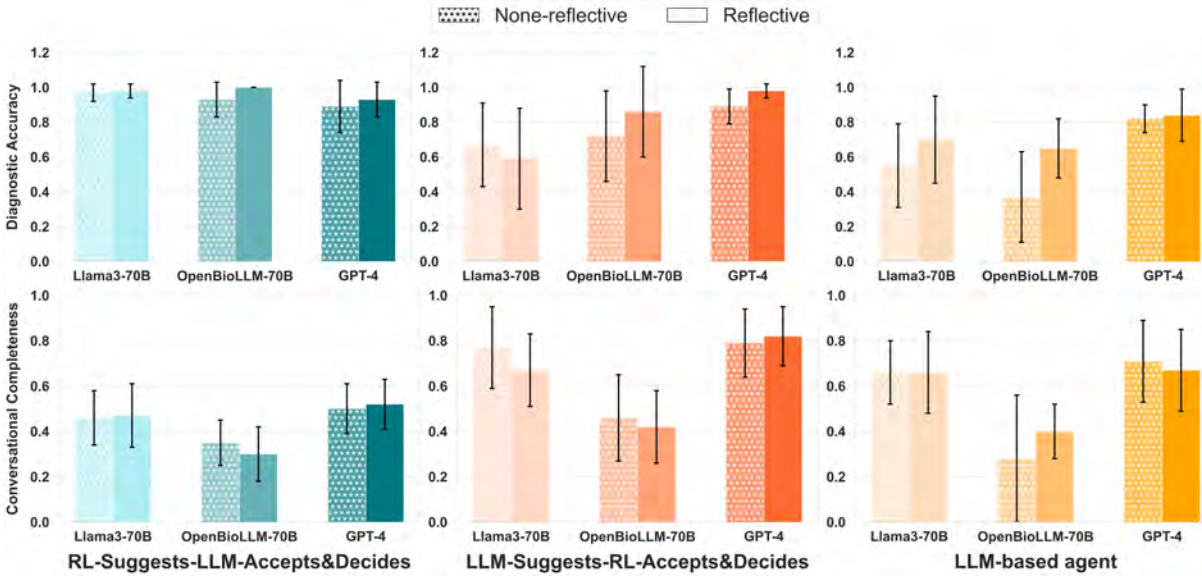


Figure 10: **Performance of none-reflective and reflective agents using different LLMs.** *Diagnostic Accuracy Score* and *Conversational Completeness Score* of the none-reflective (hashed bar) and reflective (filled bar) versions of the $RL \rightarrow LLM_{ACCEPT}$, $LLM \rightarrow RL_{ACCEPT}$, and LLM -based agents using Llama3-70B, OpenBioLLM-70B, or GPT-4. Scores are averaged across all patients in PharmaSimText.

For the $LLM \rightarrow RL_{ACCEPT}$ agents, reflection increased the *Diagnostic Accuracy Score* of all agents. However, differences were significant for $GPT-4 \rightarrow RL_{ACCEPT}$ ($p_{GPT-4} = .03$) and not significant for $OpenBio \rightarrow RL_{ACCEPT}$ and $Llama3 \rightarrow RL_{ACCEPT}$ ($p_{Llama3-70B} = .7$, $p_{OpenBioLLM-70B} = .7$). In terms of *Conversational Completeness Score*, reflection increased the performance of the $GPT-4 \rightarrow RL_{ACCEPT}$ agent ($p_{GPT-4} = .8$), but decreased the performance of the $Llama3 \rightarrow RL_{ACCEPT}$ ($p_{Llama3-70B} = .56$) and $OpenBio \rightarrow RL_{ACCEPT}$ agents ($p_{OpenBioLLM-70B} = .07^*$). Upon further inspection, we observed that reflective prompting further narrowed the Llama3-70B and OpenBioLLM-70B suggestions leading to the decreased *Conversational Completeness Score*.

The effectiveness of reflective prompting depends on the LLM and agent type. While reflection led to a significant increase in *Diagnostic Accuracy Score* for *OpenBioLLM-70B*-based and $GPT-4 \rightarrow RL_{ACCEPT}$ agents, it lowered *Conversational Completeness Score* for the $OpenBio \rightarrow RL_{ACCEPT}$ with a trend to a significant difference. Moreover, reflection improved *Diagnostic Accuracy Score* for all the OpenBioLLM-70B-based agents with an almost significant difference.

8.6. RESULTS FOR OTHER OPEN-SOURCE LLMs

In this section, we extend the results reported for RQ2 and RQ3 for Llama3-70B and OpenBioLLM-70B in Section 5 to open-source models LLaMa3-8B (Dubey et al., 2024), Gemma2-9B-it (Team, 2024), and Mixtral8x7B (Jiang et al., 2024), and medically fine-tuned open source model including OpenBioLLM-8B (Ankit Pal, 2024).

8.6.1. Extended results for RQ2: Performance of base or medically fine-tuned Open-Source LLMs

Tables 7, 8, and 9 include the results of *Diagnostic Accuracy Score*, *Conversational Completeness Score*, and *Combined Score* for seven variations of $RL \rightarrow LLM_{ACCEPT}$, $LLM \rightarrow RL_{ACCEPT}$, and *LLM-based* agents with different LLMs. Hereby, extending the results presented in Figure 9.

We also did initial experiments with Socially Intelligent Language Agent, SOTOPIA (Wang et al., 2024), acting as a LLM-based agent for three of the patients in PharmaSimText. However, for all of the patients SOTOPIA-based agent performed far worse than the *RL-based* agent in *Diagnostic Accuracy Score* (0.47) and *Conversational Completeness Score* (0.23) taking repetitive actions most of the time. Further adaptation of utilized prompts also did not significantly improve its performance. *RL-LLM* agents were designed to combine strengths of *RL/LLM* agents, therefore we believe combining *RL* and SOTOPIA agents will not boost performance as SOTOPIA-based agent is not better than the *RL-based* agent in any aspects.

8.6.2. Extended results for RQ3: Effect of Disagreement Mechanism

Tables 10 and 11 extend the results in Tables 2 and 3 for more LLMs. The optimal thresholds chosen for *RL-LLM* agents with *RL Decision-Maker* with Disagreement Mechanism are reported in Table 12

8.6.3. Extended results: Effect of Reflective Prompting

Tables 7, 8, and 9 include the results of *Diagnostic Accuracy Score*, *Conversational Completeness Score*, and *Combined Score* for none-reflective and reflective versions of seven variations of $RL \rightarrow LLM_{ACCEPT}$, $LLM \rightarrow RL_{ACCEPT}$, and *LLM-based* agents with different LLMs. Hereby, extending the results presented in Figure 10.

Table 7: Performance of *RL-Suggests-LLM-Accepts&Decides* agents with different LLMs. mean \pm standard deviation of *Diagnostic Accuracy Score*, *Conversational Completeness Score*, and *Combined Score* for none-reflective (left) and reflective (right) version of seven variations of $RL \rightarrow LLM_{ACCEPT}$ agents with different LLMs.

LLM	$RL \rightarrow LLM_{ACCEPT}$		
	<i>Diagnostic Accuracy Score</i>	<i>Conversational Completeness Score</i>	<i>Combined Score</i>
GPT-4	0.89 \pm 0.15,0.93 \pm 0.10	0.50 \pm 0.11,0.52 \pm 0.11	0.43 \pm 0.08,0.49 \pm 0.10
Llama3-70b-8192	0.97 \pm 0.05,0.98 \pm 0.04	0.46 \pm 0.12,0.47 \pm 0.14	0.45 \pm 0.13,0.46 \pm 0.15
Llama3-8b-8192	0.99 \pm 0.03,1.00 \pm 0.00	0.38 \pm 0.09,0.36 \pm 0.12	0.37 \pm 0.10,0.36 \pm 0.12
Gemma2-9b-it	0.97 \pm 0.05,0.99 \pm 0.04	0.44 \pm 0.12,0.34 \pm 0.13	0.44 \pm 0.12,0.34 \pm 0.13
Mixtral-8x7b-32768	0.89 \pm 0.18,1.00 \pm 0.00	0.41 \pm 0.14,0.34 \pm 0.14	0.37 \pm 0.17,0.34 \pm 0.14
Llama3-OpenBioLLM-70B	0.93 \pm 0.10,1.00 \pm 0.00	0.35 \pm 0.10,0.30 \pm 0.12	0.31 \pm 0.11,0.30 \pm 0.12
Llama3-OpenBioLLM-8B	0.99 \pm 0.04,0.99 \pm 0.04	0.30 \pm 0.12,0.30 \pm 0.12	0.30 \pm 0.13,0.30 \pm 0.13

Table 8: **Performance of LLM-Suggests-RL-Accepts&Decides agents with different LLMs.** mean±standard deviation of *Diagnostic Accuracy Score*, *Conversational Completeness Score*, and *Combined Score* for none-reflective (left) and reflective (right) versions of seven variations of LLM→RL_{ACCEPT} agents with different LLMs.

LLM	LLM→RL _{ACCEPT}		
	<i>Diagnostic Accuracy Score</i>	<i>Conversational Completeness Score</i>	<i>Combined Score</i>
GPT-4	0.89±0.1,0.98±0.04	0.79±0.15,0.82±0.13	0.71±0.17,0.81±0.14
Llama3-70b-8192	0.67±0.24,0.59±0.29	0.77±0.18,0.67±0.16	0.54±0.27,0.42±0.26
Llama3-8b-8192	0.42±0.21,0.76±0.18	0.36±0.11,0.32±0.13	0.16±0.07,0.25±0.14
Gemma2-9b-it	0.68±0.14,0.81±0.13	0.65±0.16,0.49±0.13	0.46±0.18,0.39±0.13
Mixtral-8x7b-32768	0.46±0.27,0.87±0.19	0.34±0.11,0.40±0.15	0.15±0.12,0.35±0.18
Llama3-OpenBioLLM-70B	0.72±0.26,0.86±0.26	0.46±0.19,0.42±0.16	0.37±0.20,0.38±0.20
Llama3-OpenBioLLM-8B	0.19±0.12,0.21±0.09	0.04±0.07,0.05±0.07	0.02±0.04,0.01±0.02

Table 9: **Performance of LLM-based agents with different LLMs.** mean±standard deviation of *Diagnostic Accuracy Score*, *Conversational Completeness Score*, and *Combined Score* for none-reflective (left) and reflective (right) versions of seven variations of LLM-based agents with different LLMs.

LLM	LLM-based		
	<i>Diagnostic Accuracy Score</i>	<i>Conversational Completeness Score</i>	<i>Combined Score</i>
GPT-4	0.82±0.08,0.84±0.15	0.71±0.18,0.67±0.18	0.59±0.20,0.58±0.20
Llama3-70b-8192	0.55±0.24,0.7±0.25	0.66±0.14,0.66±0.18	0.35±0.18,0.49±0.24
Llama3-8b-8192	0.28±0.15,0.40±0.20	0.19±0.10,0.24±0.11	0.05±0.04,0.12±0.08
Gemma2-9b-it	0.49±0.19,0.39±0.22	0.46±0.15,0.26±0.17	0.22±0.11,0.13±0.1
Mixtral-8x7b-32768	0.31±0.24,0.54±0.25	0.23±0.22,0.44±0.26	0.10±0.14,0.27±0.25
Llama3-OpenBioLLM-70B	0.37±0.26,0.65±0.17	0.28±0.28,0.40±0.12	0.15±0.19,0.29±0.11
Llama3-OpenBioLLM-8B	0.18±0.06,0.27±0.15	0.00±0.00,0.03±0.05	0.00±0.00,0.03±0.05

Table 10: **Combined Score of RL-LLM agents with LLM Decision-Maker with different Disagreement Mechanisms with different LLMs.** *Combined Score* variation of RL-LLM agents with LLM Decision-Maker with seven different LLMs.

LLM	RL-LLM Agents with LLM Decision-Maker		
	RL→LLM _{ACCEPT}	RL→LLM _{VECTO}	RL→LLM _{INTERACT}
GPT-4	0.49±0.1	0.56±0.17	0.61±0.15
Llama3-70b-8192	0.46±0.15	0.42±0.15	0.49±0.25
Llama3-8b-8192	0.36±0.12	0.3±0.08	0.31±0.12
Gemma2-9b-it	0.34±0.13	0.44±0.14	0.46±0.21
Mixtral-8x7b-32768	0.34±0.14	0.35±0.14	0.27±0.13
Llama3-OpenBioLLM-70B	0.30±0.12	0.32±0.15	0.56±0.25
Llama3-OpenBioLLM-8B	0.30±0.13	0.30±0.13	0.03±0.03

Table 11: **Combined Score of RL-LLM agents with RL Decision-Maker with different LLMs.** Combined Score of agents with RL Decision-Maker with seven different LLMs and different Disagreement Mechanisms.

LLM	RL-LLM agents with RL Decision-Maker		
	LLM→RL _{ACCEPT}	LLM→RL _{VETO}	LLM→RL _{INTERACT}
GPT-4	0.81±0.14	0.68±0.12	0.75±0.11
Llama3-70b-8192	0.42±0.26	0.64±0.21	0.72±0.17
Llama3-8b-8192	0.25±0.14	0.26±0.14	0.22±0.17
Gemma2-9b-it	0.39±0.13	0.44±0.11	0.45±0.16
Mixtral-8x7b-32768	0.35±0.18	0.29±0.12	0.33±0.14
Llama3-OpenBioLLM-70B	0.38±0.20	0.38±0.17	0.59±0.15
Llama3-OpenBioLLM-8B	0.01±0.02	0.34±0.11	0.31±0.10

Table 12: **Optimal thresholds for RL-LLM agents with RL Decision-Maker with disagreement mechanism.** Thresholds used for disagreement in agents with RL Decision-Maker for different LLMs. The results are averaged over all the available scenarios in PharmaSimText.

LLM	Agents with RL Decision-Maker (Thresholds)	
	LLM→RL _{VETO}	LLM→RL _{INTERACT}
GPT-4	0.50	0.55
Llama3-70b-8192	0.60	0.80
Llama3-8b-8192	0.25	0.42
Gemma2-9b-it	0.25	0.62
Mixtral-8x7b-32768	0.23	0.20
Llama3-OpenBioLLM-70B	0.30	0.50
Llama3-OpenBioLLM-8B	0.12	0.15

8.7. EXAMPLE OF AGENTS' CONVERSATION

Given the inferior performance of the *RL-based* agent in the *Conversational Completeness Score*, we examined the dialogues generated by the *RL-based* agent and the GPT-4 → RL_{ACCEPT} agent within an identical scenario that resulted in a correct diagnosis, as illustrated in Fig. 11. This comparison reveals a pronounced contrast in the conversational dynamics of these two agents. The dialogue led by the GPT-4 → RL_{ACCEPT} 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 GPT-4 → RL_{ACCEPT} agent engages in a more thorough inquiry, covering a broader spectrum of key questions in a logically sequential manner.

RL-based 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

GPT4 → RL_{ACCEPT} 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 11: **Example diagnostic conversations** conducted by the *RL-based* (left) and GPT-4→RL_{ACCEPT} agents (right) with the patient with joint pains in a test subtask with Osteoarthritis as the most probable cause.

REFERENCES

- AHMED, U. Z., CHRISTAKIS, M., EFREMOV, A., FERNANDEZ, N., GHOSH, A., ROYCHOUDHURY, A., AND SINGLA, A. 2020. Synthesizing tasks for block-based programming. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Curran Associates, Inc., Red Hook, NY, USA, 1–12.
- AHN, M., ZHU, H., HARTIKAINEN, K., PONTE, H., GUPTA, A., LEVINE, S., AND KUMAR, V. 2019. ROBEL: robotics benchmarks for learning with low-cost robots. In *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds. Proceedings of Machine Learning Research, vol. 100. PMLR, 1300–1313.
- AMMANABROLU, P. AND HAUSKNECHT, M. J. 2020. Graph constrained reinforcement learning for natural language action spaces. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- ANKIT PAL, M. S. 2024. Openbiollms: Advancing open-source large language models for healthcare and life sciences. <https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B>.
- BARNES, T. AND STAMPER, J. C. 2008. Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*. Springer, Berlin, Germany, 373–382.
- BEWERSDORFF, A., SESSLER, K., BAUR, A., KASNECI, E., AND NERDEL, C. 2023. Assessing Student Errors Experimentation Using Artificial Intelligence and Large Language Models: A Comparative Study with Human Raters. *Online publication*. [abs/2308.06088](https://arxiv.org/abs/2308.06088), 100177.
- BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics* 5, 135–146.
- BOTELHO, A. F., ADJEI, S., AND HEFFERNAN, N. T. 2016. Modeling interactions across skills: A method to construct and compare models predicting the existence of skill relationships. In *Proceed-*

- ings of the 9th International Conference on Educational Data Mining (EDM 2016). International Educational Data Mining Society, Raleigh, NC, USA, 292–297.
- BROWN, T. B., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., AGARWAL, S., HERBERT-VOSS, A., KRUEGER, G., HENIGHAN, T., CHILD, R., RAMESH, A., ZIEGLER, D. M., WU, J., WINTER, C., HESSE, C., CHEN, M., SIGLER, E., LITWIN, M., GRAY, S., CHESS, B., CLARK, J., BERNER, C., MCCANDLISH, S., RADFORD, A., SUTSKEVER, I., AND AMODEI, D. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds.
- BUBECK, S. ET AL. 2023. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. *Online publication. abs/2303.12712*. arXiv preprint, 154 pages.
- BUNEL, R., HAUSKNECHT, M. J., DEVLIN, J., SINGH, R., AND KOHLI, P. 2018. Leveraging grammar and reinforcement learning for neural program synthesis. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- CORBETT, A. T. AND ANDERSON, J. R. 2005. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction* 4, 253–278.
- DENNY, P., GULWANI, S., HEFFERNAN, N. T., KÄSER, T., MOORE, S., RAFFERTY, A. N., AND SINGLA, A. 2024. Generative AI for Education (GAIED): Advances, Opportunities, and Challenges. *CoRR abs/2402.01580*.
- DU, Y., WATKINS, O., WANG, Z., COLAS, C., DARRELL, T., ABBEEL, P., GUPTA, A., AND ANDREAS, J. 2023. Guiding pretraining in reinforcement learning with large language models. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, Honolulu, HI, USA, 8657–8677.
- DUBEY, A., JAUHRI, A., PANDEY, A., KADIAN, A., AL-DAHLE, A., LETMAN, A., MATHUR, A., SCHELLEN, A., YANG, A., FAN, A., GOYAL, A., HARTSHORN, A., YANG, A., MITRA, A., SRIVANKUMAR, A., KORENEV, A., HINSVARK, A., RAO, A., ZHANG, A., RODRIGUEZ, A., GREGERSON, A., SPATARU, A., ROZIÈRE, B., BIRON, B., TANG, B., CHERN, B., CAUCHETEUX, C., NAYAK, C., BI, C., MARRA, C., MCCONNELL, C., KELLER, C., TOURET, C., WU, C., WONG, C., FERRER, C. C., NIKOLAIDIS, C., ALLONSIUS, D., SONG, D., PINTZ, D., LIVSHITS, D., ESIOSU, D., CHOUDHARY, D., MAHAJAN, D., GARCIA-OLANO, D., PERINO, D., HUPKES, D., LAKOMKIN, E., ALBADAWY, E., LOBANOVA, E., DINAN, E., SMITH, E. M., RADENOVIC, F., ZHANG, F., SYNNAEVE, G., LEE, G., ANDERSON, G. L., NAIL, G., MIALON, G., PANG, G., CUCURELL, G., NGUYEN, H., KOREVAAR, H., XU, H., TOUVRON, H., ZAROV, I., IBARRA, I. A., KLOUMANN, I. M., MISRA, I., EVTIMOV, I., COPET, J., LEE, J., GEFFERT, J., VRANES, J., PARK, J., MAHADEOKAR, J., SHAH, J., VAN DER LINDE, J., BILLOCK, J., HONG, J., LEE, J., FU, J., CHI, J., HUANG, J., LIU, J., WANG, J., YU, J., BITTON, J., SPISAK, J., PARK, J., ROCCA, J., JOHNSTUN, J., SAXE, J., JIA, J., ALWALA, K. V., UPASANI, K., PLAWIAK, K., LI, K., HEAFIELD, K., STONE, K., AND ET AL. 2024. The llama 3 herd of models. *Advance online publication. https://doi.org/10.48550/arXiv.2407.21783*.
- EFREMOV, A., GHOSH, A., AND SINGLA, A. 2020. Zero-shot learning of hint policy via reinforcement learning and program synthesis. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, A. N. Rafferty, J. Whitehill, C. Romero, and V. Cavalli-Sforza, Eds. International Educational Data Mining Society.

- FAUCON, L., KIDZINSKI, L., AND DILLENBOURG, P. 2016. Semi-Markov Model for Simulating MOOC Students. In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*. International Educational Data Mining Society (IEDMS), 358–363.
- HE, J., CHEN, J., HE, X., GAO, J., LI, L., DENG, L., AND OSTENDORF, M. 2016. Deep Reinforcement Learning with a Natural Language Action Space. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. The Association for Computer Linguistics.
- HE-YUEYA, J. AND SINGLA, A. 2021. Quizzing policy using reinforcement learning for inferring the student knowledge state. In *Proceedings of the 14th International Conference on Educational Data Mining, EDM 2021, virtual, June 29 - July 2, 2021*, S. I. Hsiao, S. S. Sahebi, F. Bouchet, and J. Vie, Eds. International Educational Data Mining Society.
- HIRUNYASIRI, D., THOMAS, D. R., LIN, J., KOEDINGER, K. R., AND ALEVEN, V. 2023. Comparative Analysis of GPT-4 and Human Graders in Evaluating Praise Given to Students in Synthetic Dialogues. *CoRR abs/2307.02018*.
- ICHTER, B., BROHAN, A., CHEBOTAR, Y., FINN, C., HAUSMAN, K., HERZOG, A., HO, D., IBARZ, J., IRPAN, A., JANG, E., JULIAN, R., KALASHNIKOV, D., LEVINE, S., LU, Y., PARADA, C., RAO, K., SERMANET, P., TOSHEV, A., VANHOUCHE, V., XIA, F., XIAO, T., XU, P., YAN, M., BROWN, N., AHN, M., CORTES, O., SIEVERS, N., TAN, C., XU, S., REYES, D., RETTINGHOUSE, J., QUIAMBAO, J., PASTOR, P., LUU, L., LEE, K., KUANG, Y., JESMONTH, S., JOSHI, N. J., JEFFREY, K., RUANO, R. J., HSU, J., GOPALAKRISHNAN, K., DAVID, B., ZENG, A., AND FU, C. K. 2022. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, K. Liu, D. Kulic, and J. Ichnowski, Eds. *Proceedings of Machine Learning Research*, vol. 205. PMLR, 287–318.
- JIANG, A. Q., SABLAYROLLES, A., ROUX, A., MENSCH, A., SAVARY, B., BAMFORD, C., CHAPLOT, D. S., DE LAS CASAS, D., HANNA, E. B., BRESSAND, F., LENGYEL, G., BOUR, G., LAMPLE, G., LAVAUD, L. R., SAULNIER, L., LACHAUX, M., STOCK, P., SUBRAMANIAN, S., YANG, S., ANTONIAK, S., SCAO, T. L., GERVET, T., LAVRIL, T., WANG, T., LACROIX, T., AND SAYED, W. E. 2024. Mixtral of experts. *Advance online publication*. <https://doi.org/10.48550/arXiv.2401.04088>.
- JU, S., CHI, M., AND ZHOU, G. 2020. Pick the Moment: Identifying Critical Pedagogical Decisions Using Long-Short Term Rewards. In *Proceedings of the International Conference on Educational Data Mining (EDM)*. International Educational Data Mining Society.
- KÄSER, T. AND ALEXANDRON, G. 2023. Simulated learners in educational technology: A systematic literature review and a turing-like test. *International Journal of Artificial Intelligence in Education*, 1–41.
- KUMAR, A. P., NAYAK, A., K, M. S., CHAITANYA, AND GHOSH, K. 2023. 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)*, 1–44.
- KWON, M., XIE, S. M., BULLARD, K., AND SADIGH, D. 2023. Reward Design with Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net.
- LEE, U., LEE, S., KOH, J., JEONG, Y., JUNG, H., BYUN, G., LEE, Y., MOON, J., LIM, J., AND KIM, H. 2023. 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)*.

- LI, N., COHEN, W. W., KOEDINGER, K. R., AND MATSUDA, N. 2011. A Machine Learning Approach for Automatic Student Model Discovery. In *Proceedings of the International Conference on Educational Data Mining (EDM)*. www.educationaldatamining.org, 31–40.
- LI, S., PUIG, X., PAXTON, C., DU, Y., WANG, C., FAN, L., CHEN, T., HUANG, D., AKYÜREK, E., ANANDKUMAR, A., ANDREAS, J., MORDATCH, I., TORRALBA, A., AND ZHU, Y. 2022. Pre-Trained Language Models for Interactive Decision-Making. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. Neural Information Processing Systems Foundation.
- LIN, B. Y., FU, Y., YANG, K., BRAHMAN, F., HUANG, S., BHAGAVATULA, C., AMMANABROLU, P., CHOI, Y., AND REN, X. 2023. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds.
- MACLELLAN, C. J. AND GUPTA, A. 2021. Learning Expert Models for Educationally Relevant Tasks using Reinforcement Learning. In *Proceedings of the International Conference on Educational Data Mining (EDM)*. International Educational Data Mining Society, Online.
- MACLELLAN, C. J., HARPSTEAD, E., PATEL, R., AND KOEDINGER, K. R. 2016. The Apprentice Learner Architecture: Closing the Loop between Learning Theory and Educational Data. In *Proceedings of the International Conference on Educational Data Mining (EDM)*. International Educational Data Mining Society (IEDMS), 151–158.
- MAJUMDER, B. P., MISHRA, B. D., JANSEN, P. A., TAFJORD, O., TANDON, N., ZHANG, L., CALLISON-BURCH, C., AND CLARK, P. 2023. Clin: A continually learning language agent for rapid task adaptation and generalization. *CoRR abs/2310.10134*.
- MARKEL, J. M., OPFERMAN, S. G., LANDAY, J. A., AND PIECH, C. 2023. GPTeach: Interactive TA Training with GPT-based Students. In *Proceedings of the Conference on Learning @ Scale (L@S)*. ACM, 226–236.
- MCILROY-YOUNG, R., SEN, S., KLEINBERG, J. M., AND ANDERSON, A. 2020. Aligning superhuman AI with human behavior: Chess as a model system. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 1677–1687.
- MCNICHOLS, H., FENG, W., LEE, J., SCARLATOS, A., SMITH, D., WOODHEAD, S., AND LAN, A. 2023. Automated Distractor and Feedback Generation for Math Multiple-choice Questions via In-context Learning. *NeurIPS'23 Workshop on Generative AI for Education (GAIED)*.
- MNIH, V., KAVUKCUOGLU, K., SILVER, D., GRAVES, A., ANTONOGLU, I., WIERSTRA, D., AND RIEDMILLER, M. A. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR abs/1312.5602*.
- MUI, J., LIN, F., AND DEWAN, M. A. A. 2021. Multi-Armed Bandit Algorithms for Adaptive Learning: A Survey. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*. Lecture Notes in Computer Science, vol. 12748. Springer, 273–278.
- NAZARETSKY, T., HERSHKOVITZ, S., AND ALEXANDRON, G. 2019. 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)*. International Educational Data Mining Society (IEDMS).
- NGUYEN, M. H., TSCHIATSCHKEK, S., AND SINGLA, A. 2024. Large language models for in-context student modeling: Synthesizing student's behavior in visual programming. In *Proceedings of the 17th International Conference on Educational Data Mining, EDM 2024, Atlanta, Georgia, USA, July*

- 14-17, 2024, D. A. Joyner, B. Paaßen, and C. D. Epp, Eds. International Educational Data Mining Society.
- NIKISHIN, E., SCHWARZER, M., D'ORO, P., BACON, P., AND COURVILLE, A. C. 2022. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds. Proceedings of Machine Learning Research, vol. 162. PMLR, 16828–16847.
- NOTTINGHAM, K., AMMANABROLU, P., SUHR, A., CHOI, Y., HAJISHIRZI, H., SINGH, S., AND FOX, R. 2023. 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)*. PMLR, 26311–26325.
- OPENAI. 2023. GPT-4 technical report. *Online publication*. <https://doi.org/10.48550/arXiv.2303.08774>.
- PADUREAN, V., TZANNETOS, G., AND SINGLA, A. 2024. Neural Task Synthesis for Visual Programming. *Transactions of Machine Learning Research (TMLR)*.
- PAN, A., SHERN, C. J., ZOU, A., LI, N., BASART, S., WOODSIDE, T., NG, J., ZHANG, H., EMMONS, S., AND HENDRYCKS, D. 2023. Do the Rewards Justify the Means? Measuring Trade-Offs Between Rewards and Ethical Behavior in the MACHIARELLI Benchmark. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 26837–26867.
- PANKIEWICZ, M. AND BAKER, R. S. 2023. Large Language Models (GPT) for Automating Feedback on Programming Assignments. *CoRR abs/2307.00150*.
- PARDOS, Z. A. AND BHANDARI, S. 2023. Learning Gain Differences between ChatGPT and Human Tutor Generated Algebra Hints. *CoRR abs/2302.06871*.
- PARETO, L. 2014. 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.
- PHUNG, T., PADUREAN, V., CAMBRONERO, J., GULWANI, S., KOHN, T., MAJUMDAR, R., SINGLA, A., AND SOARES, G. 2023a. 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)*. ACM.
- PHUNG, T., PADUREAN, V., CAMBRONERO, J., GULWANI, S., KOHN, T., MAJUMDAR, R., SINGLA, A., AND SOARES, G. 2023b. Generative AI for programming education: Benchmarking chatgpt, gpt-4, and human tutors. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2, ICER 2023, Chicago, IL, USA, August 7-11, 2023*, K. Fisler, P. Denny, D. Franklin, and M. Hamilton, Eds. ACM, 41–42.
- PHUNG, T., PADUREAN, V., SINGH, A., BROOKS, C., CAMBRONERO, J., GULWANI, S., SINGLA, A., AND SOARES, G. 2024. 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)*. ACM, 333–344.
- RADMEHR, B., SINGLA, A., AND KÄSER, T. 2024. Towards generalizable agents in text-based educational environments: A study of integrating rl with llms. In *Proceedings of the 17th International Conference on Educational Data Mining*, B. Paaßen and C. D. Epp, Eds. International Educational Data Mining Society, Atlanta, Georgia, USA, 181–193.
- RAFFERTY, A. N., BRUNSKILL, E., GRIFFITHS, T. L., AND SHAFTO, P. 2016. Faster Teaching via POMDP Planning. *Cognitive Science* 40, 6, 1290–1332.

- RAFFERTY, A. N., WILLIAMS, J. J., AND YING, H. 2019. Statistical Consequences of Using Multi-Armed Bandits to Conduct Adaptive Educational Experiments. *Journal of Educational Data Mining (JEDM)* 11, 47–79.
- ROBINSON, K., JAHANIAN, K., AND REICH, J. 2018. Using online practice spaces to investigate challenges in enacting principles of equitable computer science teaching. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE 2018)*, T. Barnes, D. D. Garcia, E. K. Hawthorne, and M. A. Pérez-Quñones, Eds. Association for Computing Machinery, New York, NY, USA, 882–887.
- SARSA, S., DENNY, P., HELLAS, A., AND LEINONEN, J. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 18th ACM Conference on International Computing Education Research (ICER 2022)*, J. Vahrenhold, K. Fisler, M. Hauswirth, and D. Franklin, Eds. Association for Computing Machinery, New York, NY, USA, 27–43.
- SCHMUCKER, R., XIA, M., AZARIA, A., AND MITCHELL, T. 2023. Ruffle&riley: Towards the automated induction of conversational tutoring systems. *Proceedings of the NeurIPS 2023 Workshop on Generative AI for Education (GAIED)*.
- SHINN, N., CASSANO, F., BERMAN, E., GOPINATH, A., NARASIMHAN, K., AND YAO, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *CoRR abs/2303.11366*.
- SINGLA, A., RAFFERTY, A. N., RADANOVIC, G., AND HEFFERNAN, N. T. 2021. Reinforcement Learning for Education: Opportunities and Challenges. *Online publication*. <https://arxiv.org/abs/2107.08828>.
- SINGLA, A. AND THEODOROPOULOS, N. 2022. From {Solution Synthesis} to {Student Attempt Synthesis} for Block-Based Visual Programming Tasks. In *Proceedings of the International Conference on Educational Data Mining (EDM)*. International Educational Data Mining Society, Durham, UK.
- SUTTON, R. S. AND BARTO, A. G. 2018. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, USA.
- TACK, A. AND PIECH, C. 2022. The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues. In *Proceedings of the 15th International Conference on Educational Data Mining (EDM 2022)*. International Educational Data Mining Society, Durham, UK.
- TEAM, G. 2024. Gemma. Available on Kaggle.
- TOWERS, M., KWIATKOWSKI, A., TERRY, J. K., BALIS, J. U., COLA, G. D., DELEU, T., GOULÃO, M., KALLINTERIS, A., KRIMMEL, M., KG, A., PEREZ-VICENTE, R., PIERRÉ, A., SCHULHOFF, S., TAI, J. J., TAN, H., AND YOUNIS, O. G. 2024. Gymnasium: A standard interface for reinforcement learning environments. *Advance online publication*. [abs/2407.17032](https://arxiv.org/abs/2407.17032).
- WANG, R., JANSEN, P. A., CÔTÉ, M.-A., AND AMMANABROLU, P. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 11279–11298.
- WANG, R., YU, H., ZHANG, W. S., QI, Z., SAP, M., BISK, Y., NEUBIG, G., AND ZHU, H. 2024. Sotopia- π : Interactive learning of socially intelligent language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 12912–12940.
- WHITEHILL, J. AND MOVELLAN, J. R. 2018. Approximately Optimal Teaching of Approximately Optimal Learners. *IEEE Transactions of Learning Technology* 11, 2, 152–164.

- WU, C., KREIDIEH, A., PARVATE, K., VINITSKY, E., AND BAYEN, A. M. 2017. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *Advance online publication*. <http://arxiv.org/abs/1710.05465>. No page numbers, article available on arXiv.
- XU, C., DING, W., LYU, W., LIU, Z., WANG, S., HE, Y., HU, H., ZHAO, D., AND LI, B. 2022. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds.
- YAO, S., NARASIMHAN, K., AND HAUSKNECHT, M. J. 2021. Reading and acting while blindfolded: The need for semantics in text game agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds. Association for Computational Linguistics, 3097–3102.
- YAO, S., ZHAO, J., YU, D., DU, N., SHAFRAN, I., NARASIMHAN, K. R., AND CAO, Y. 2023. React: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*. OpenReview.net.
- ZHOU, G., AZIZSOLTANI, H., AUSIN, M. S., BARNES, T., AND CHI, M. 2019. Hierarchical reinforcement learning for pedagogical policy induction. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*. Lecture Notes in Computer Science, vol. 11625. Springer International Publishing, Cham, Switzerland, 544–556.