Best of Both Worlds: Harmonizing LLM Capabilities in Decision-Making and Question-Answering for Treatment Regimes

Hongxuan Liu*
Department of Chemical Engineering
Tsinghua University
Beijing 100084, China
liuhx21@mails.tsinghua.edu.cn

Zhiyao Luo†
Institute of Biomedical Engineering
University of Oxford
Oxford OX3 7DQ, United Kingdom
zhiyao.luo@eng.ox.ac.uk

Tingting Zhu‡
Institute of Biomedical Engineering
University of Oxford
Oxford OX3 7DQ, United Kingdom
tingting.zhu@eng.ox.ac.uk

Abstract

This paper introduces a framework that incorporates fine-tuning large language models (LLM) with reinforcement learning (RL) in the application of the dynamic treatment regime (DTR). Within the RL training framework, our bilevel-LLM framework makes use of indications from the DTR environment for 'RL with Environment Feedback' (RLEF) fine-tuning to achieve best-of-both-world results. Experimental results show that LLM-RLEF agent outperforms both existing RL policies and pure LLM policies on the *SimGlucoseEnv* treatment regime task, improving sampling efficiency, generalizability, and interpretability. In addition to improving DTR performance, RLEF improves LLM's question-answering ability on the MMLU-Med, MedQA, and MedMCQA benchmarks.

1 Introduction

As the demand for effective treatment of complex diseases increases, personalized therapies tailored to patient-specific characteristics have become a critical topic in modern healthcare [1]. Traditional population-based treatment methods are shifting toward more individualized and dynamic approaches, particularly in the management of chronic diseases and multimorbidity, where therapeutic interventions can vary significantly between individuals. In this context, Dynamic Treatment Regimes (DTRs) [2] have emerged as a critical component of personalized medicine. The DTRs are designed to develop sequential decision-making policies that adapt to changes in patient status and disease progression, providing opportunities for individualized and precise disease management.

Reinforcement learning (RL) has been effectively applied in DTR and showed promising potential. Zhu et.al [3] applied RL algorithms to closed-loop blood glucose control for type-I diabetes patients and witnessed the outstanding performance, whereas Raghu et.al [4] investigated RL algorithm's effect on Sepsis treatment. However, RL remains limited for clinical deployment as it faces several

*First author.
†Corresponding author.
‡Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

challenges: (i) **Sampling Efficiency and Safe Exploration** RL algorithms often require sufficient exploration, and sometimes include potentially harmful actions to learn from failure. This trial-and-error learning framework is unharmful in game settings, while unacceptable in clinical applications [5]; (ii) **Generalization to Unseen Patients** RL algorithms trained on a stationary environment are prone to the change of underlying Markov Decision Process (MDP), making it difficult to generalize to patients who have distinctive Pharmacokinetic—Pharmacodynamic (PK/PD) dynamics [6]; (iii) **Decision Interpretability** –although there have been recent advances in improving the interpretability of RL [7, 8], interpreting AI-generated recommendations for clinicians with domain knowledge remains non-trivial, leading to hesitation in trusting and utilizing RL as a decision-support tool.

In recent years, LLMs have demonstrated their potential in general knowledge understanding and reasoning [9], drawing significant attention from various fields of deep learning applications, including medical and healthcare domains [10, 11]. In terms of applying foundation models to decision-making tasks, existing methods such as ReAct [12], Reflexion [13] and Retroformer [14] effectively leverage LLM's capability in few-shot reasoning, reflecting and summarizing to assist decision making, but the LLMs used are either pre-trained models or fine-tuned in advance, independent to the RL task interactions, which implies that these methods do not effectively make use of RL environment feedback to improve the inherent capability of foundation models, but rather limited to the scope of prompt engineering based on prior knowledge and episodic memory.

When applying foundation models to healthcare scenarios such as general medical knowledge tasks, two common approaches are Supervised Fine-tuning (SFT) [9] and Reinforcement Learning with Human Feedback (RLHF) [15]. SFT involves fine-tuning pre-trained language models on labelled instruction-following data to enhance task-specific performance. However, this approach often requires extensive and costly data collection. RLHF, on the other hand, aligns the model with human preferences but also demands expensive human feedback annotations and computationally intensive reinforcement learning (RL) training [16, 17].

In addition to feedback from humans, environments can naturally provide reward signals that reflect preferences. We term this approach Reinforcement Learning with Environment Feedback (RLEF), which serves as an alternative to the traditional RLHF method [15]. In the medical field, clinical preference data is often expensive and not readily available. This leads us to ask: can we use environmental feedback in place of human feedback to fine-tune a language model, thereby improving both treatment performance (measured by RL rewards) and general medical question-answering ability?

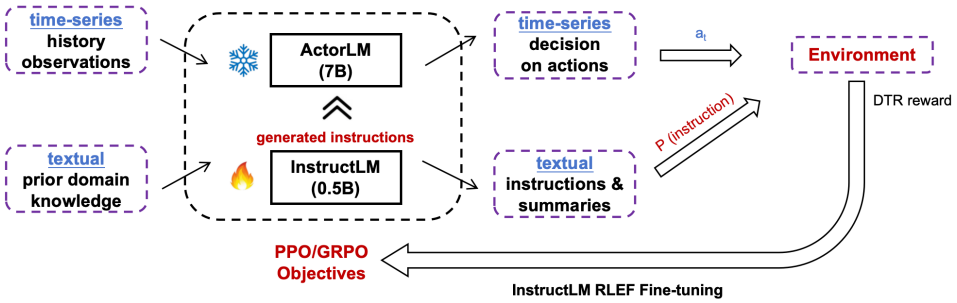

Figure 1: **Schematic Framework of Bilevel-LLM RLEF Training.** At every timestep t, InstructLM summarizes the decision rules and generates prior-knowledge embedded instruction P by analyzing time-series history, while ActorLM uses both history and instructions to take action a_t. The DTR environment-generated reward was assigned to the last token in instruction P (combined with KL penalty in PPO) to be used for RLEF fine-tuning by optimizing PPO or GRPO objectives.

This paper aims to achieve two main objectives: firstly, to **explore the potential of large language models (LLMs) in developing more effective medical treatment regimens**, and secondly, to **investigate how RLEF can enhance the question-answering capabilities of LLMs in the medical domain.** We introduce a bilevel-LLM RLEF framework that:

1. **on the decision-making side:** incorporates LLM's prior knowledge to address limitations of RL in medical dynamic treatment regimes (DTR).

2. **on the question-answering side:** utilises treatment environment feedback to enhance the LLM's question-answering capability in the medical domain.

We inherit the heuristics of Retroformer [14] in applying LLM to RL tasks, where a smaller language model generates instructions and feedback as prompts to assist a larger language model in making decisions. Unlike Retroformer or other similar frameworks, our smaller "InstructLM" is trained during the RL learning paradigm, while other frameworks leave both language models frozen during RL training. By applying this framework to DTR, we propose a novel approach that combines the strengths of both LLM and RL, aiming for optimal results in medical applications.

2 Related Work

RL-based Dynamic Treatment Regime and Benchmarks Reinforcement learning (RL) applications in dynamic treatment regimes (DTRs) are divided into two primary approaches: simulation-based and real-world data-based methods, each with distinct advantages and challenges. Real-world data-based DTRs leverage observational healthcare data to train and evaluate RL models, relying heavily on off-policy evaluation, as seen in various studies [18, 19, 20], due to ethical and logistical barriers to directly testing experimental policies in clinical settings [21, 22, 23]. However, the lack of online testing in real-world data-based DTRs complicates the validation and iterative improvement of RL algorithms under genuine clinical conditions, often creating a gap between theoretical advancements and actual clinical efficacy.

In contrast, simulation-based DTRs provide a controlled environment for testing RL algorithms in various healthcare scenarios without ethical concerns about patient involvement, as discussed in [3, 24, 25]. These simulations enable exhaustive testing across multiple hypothetical scenarios, allowing for unlimited trial-and-error iterations on virtual patients, an approach impractical in real-world settings. DTR-Bench [6] established a unified framework to simulate various healthcare DTRs and compare the effectiveness of various RL algorithms in DTR applications including cancer chemotherapy, radiotherapy, glucose management in diabetes, and sepsis treatment. The research findings confirm the non-robustness and lack of adaptability of many RL algorithms in DTR applications, and underscore the necessity in the healthcare community to shed light on new perspectives for solving the pitfalls of pure RL algorithms.

Language Model RLHF-tuning Recent advancements in Large Language Models (LLMs) have significantly benefited from Reinforcement Learning from Human Feedback (RLHF) [15], an approach that fine-tunes pre-trained models by incorporating human preferences into the training process. In terms of on-policy RLHF approaches (for a detailed discussion on the classification of on-policy and off-policy RL methods, see Appendix A), ReMax [26] abandons the critic model and uses the optimal action under the current strategy as the baseline to reduce variance, achieving lower computation and memory cost in fine-tuning than conventional Proximal Policy Optimization (PPO) approach. Group-relative Policy Optimization (GRPO) [16] inherits PPO RLHF's optimization objective but substitutes critic model-related advantage estimation with group-relative Monte Carlo estimation, thereby saving training costs and achieving better performance.

Despite these advances, there are challenges to scaling RLHF efficiently, especially when collecting massive amounts of high-quality human preference data and training a reward model requires great costs [27]. It is necessary and urgent to propose new methods for generating rewards more efficiently and effectively to guide language model fine-tuning.

Autonomous Language Model Agent for Decision Making Recent advancements in autonomous decision-making frameworks utilizing LLMs have demonstrated significant potential across various tasks. These frameworks, which leverage the generalizability and knowledge-rich capabilities of LLMs, can be broadly categorized into open-loop and closed-loop approaches. Open-loop LLM-based frameworks, such as ReAct [12], Reflexion [13], and ADaPT [28], employ LLMs to generate "thoughts" of problem solving based on observations without real-time feedback from the environment. For instance, ReAct enables the model to dynamically adjust its strategies, while Reflexion integrates verbal feedback to enhance decision-making capabilities by augmenting the model's episodic memory. However, these approaches often do not incorporate direct environmental rewards, limiting their adaptability. In contrast, closed-loop LLM-based frameworks, such as Refiner [29], Retroformer [14],

and REX [30], incorporate feedback mechanisms that facilitate iterative learning. Refiner employs a fine-tuned LLM for policy decision feedback, whereas Retroformer utilizes a smaller trainable LM to provide verbal feedback based on received rewards to assist a frozen policy model in decision-making. REX adopts Monte-Carlo Tree Search (MCTS) [31] to guide the language model's exploration.

Admittedly, existing closed-loop frameworks acquire strong adaptability to the characteristics and dynamics of specific RL environments, but the foundation models used are either pre-trained or fine-tuned in advance before being applied to RL interactions; therefore, the utilization of environmental feedback and rewards is only limited to the scope of prompt engineering, and the capabilities of the models do not get improved over the RL tasks. Some multi-modal LLMs like PaLM-E [32] are trained on RL tasks, such as robot control, for better capabilities in grounding languages to actions [33, 34] or decision makings. These frameworks bring improvements to foundation models only in very limited task-related domains. However, bridging the gap between improving RL-related tasks as well as language-task performance for language models during RL interaction still needs further insight.

3 Methodology

In this work, we explore an RL framework in which the traditional reward model is supplanted by direct environmental feedback. Our approach incorporates two language models with distinct roles: a smaller, trainable LLM for generating prompts from textual states and a larger, fixed LLM for making treatment decisions based on these prompts. The objective is to utilize step-wise rewards from the environment to iteratively improve the text generation capability of the smaller LLM, ultimately leading to better treatment outcomes. We formulate the RL problem in DTR, the convention RLHF, and then introduce our method, which incorporates environmental feedback into the existing RLHF framework.

3.1 Reinforcement Learning for Dynamic Treatment Regime

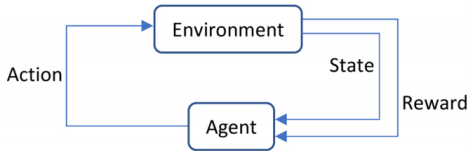

Figure 2: **The Reinforcement Learning Framework.**

A DTR Markov Decision Process is formally defined as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. The set $\mathcal{S}$ represents a finite set of states (i.e., clinical observation); $\mathcal{A}$ denotes a finite set of actions (i.e., drug dose). The state transition probability function $P_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ characterizes the PK/PD dynamics of the patient i.

The primary objective in RL is to learn an optimal treatment policy $\pi^* : \mathcal{S} \to \mathcal{A}$ to maximize the expected cumulative discounted reward:

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right], \tag{1}$$

where Π is the space of all possible treatment policies, $\tau = (s_0, a_0, s_1, a_1, \ldots)$ represents a trajectory, and $s_0 \in \mathcal{S}$ is the initial state of the patient.

These fundamental concepts form the basis for various RL algorithms. In Appendix A, we provide an overview of several well-recognized RL algorithms in DTR, including Deep Q-Network (DQN) [35] and Proximal Policy Optimization (PPO) [36].

3.2 Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback uses RL to optimize human preference given a learned reward model and an SFT model. Inspired by standard RL, RLHF uses a slightly different MDP for text generation. To differentiate from the MDP setup in dynamic treatment regimes, we use different notations to represent the state and action in RLHF.

Consider a language model π_θ with vocabulary $x \in \mathcal{V}$; the state s_t^{H} is defined as all tokens generated so far $s_t^{\text{H}} = (x_1, x_2, \ldots, x_t)$, and the action is the next possible token x_{t+1} . The policy is the next-token-prediction distribution of the language model $\pi_\theta(x_{t+1}|s_t^{\text{H}})$. The quality of text generation should be aligned with the human preference oracle $r_H(x_{1:T})$ with T tokens. Before optimizing LLMs with human preference, a learned reward model R_ϕ is needed to approximate the reward oracle based on collected human feedback $R_\phi(x_{1:T}) \approx r_H(x_{1:T})$.

The primary objective in RLHF is to find the optimal policy π_θ^* that maximizes the expected cumulative discounted reward. This can be formalized as follows:

$$\pi_\theta^* = \arg\max_{\pi_\theta} \mathbb{E}_{x_{1:T}\sim\pi_\theta} \left[\sum_{t=1}^{T} \gamma^{t-1} R_\phi(x_{1:t}) \right] \tag{2}$$

Examples of RL algorithms used in RLHF are Proximal Policy Optimization [15] and Group-Relative Policy Optimization (**GRPO**) [16]. PPO introduces a clipping mechanism to control the magnitude of policy updates, leading to more stable training compared to the preceding actor-critic algorithms [36, 15]. Based on PPO, GRPO uses the average rewards of multiple sampled outputs corresponding to the same question to estimate advantage, greatly reducing the memory and computational costs of the value model.

In RLHF, the objective function for PPO/GRPO policy optimization is represented as:

$$\begin{aligned} &\mathcal{J}_{PPO}(\theta) = \mathbb{E}\left[q \sim P(Q), x \sim \pi_{\theta_{\text{old}}}(X|q)\right] \\ &\frac{1}{|x|}\sum_{t=1}^{|x|} \left\{ \min\left[\frac{\pi_\theta(x_t|q,x_{<t})}{\pi_{\theta_{\text{old}}}(x_t|q,x_{<t})} A_t, \text{clip}\left(\frac{\pi_\theta(x_t|q,x_{<t})}{\pi_{\theta_{\text{old}}}(x_t|q,x_{<t})}, 1-\epsilon, 1+\epsilon \right) A_t \right] \right\}, \end{aligned} \tag{3}$$

$$\begin{aligned} &\mathcal{J}_{GRPO}(\theta) = \mathbb{E}\left[q \sim P(Q), \{x^i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(X|q)\right] \frac{1}{G}\sum_{i=1}^{G} \frac{1}{|x^i|} \\ &\sum_{t=1}^{|x^i|} \left\{ \min\left[\frac{\pi_\theta(x_t^i|q,x_{<t}^i)}{\pi_{\theta_{\text{old}}}(x_t^i|q,x_{<t}^i)} \hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(x_t^i|q,x_{<t}^i)}{\pi_{\theta_{\text{old}}}(x_t^i|q,x_{<t}^i)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{i,t} \right] - \beta \mathcal{D}_{KL}[\pi_\theta \| \pi_{\text{ref}}] \right\}, \end{aligned} \tag{4}$$

where π_θ and $\pi_{\theta_{old}}$ are the current and old policy models, and q, x are questions and outputs sampled from the question dataset and the old policy $\pi_{\theta_{old}}$, respectively. ε is a clipping-related hyper-parameter to constrain the magnitude of policy updates for stabilizing training. A_t is the advantage, which in PPO is computed by applying Generalized Advantage Estimation (GAE) [37] based on the rewards $\{r_{\geq t}\}$ and a learned value model V_ψ trained alongside the policy model. In GRPO, A_t is determined by the difference of new policy's reward from the normalized group rewards $\mathbf{r} = \{r_1, r_2, \cdots, r_G\}$, i.e., $\hat{A}_{i,t} = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$, where $\{r_1, r_2, \cdots, r_G\}$ represents rewards of a group of sampling outputs $\{x^1, x^2, \cdots, x^G\}$ from the old policy $\pi_{\theta_{old}}$.

To mitigate over-optimization of the reward model, PPO adds a KL penalty per token $\log \frac{\pi_\theta(x_t|q,x_{<t})}{\pi_{\text{ref}}(x_t|q,x_{<t})}$ to the reward term, controlled by hyper-parameter β. While in GRPO, an unbiased KL divergence estimator [38] is added as an independent term. Both approaches maintain the similarity between the trained policy and the reference policy, guaranteeing the regularity of the new policy's output.

3.3 DTR Environment - *SimGlucoseEnv*

We investigated *SimGlucoseEnv* - a simulation-based insulin administration environment for Type-1 diabetic patients. In *SimGlucoseEnv*, the blood glucose dynamics are determined based on real-world

data from 300 patients, covering a range of metabolic parameters and demographic characteristics. The dynamics are formulated in ordinary differential equations (ODEs), which are developed based on computational models that simulate interactions between insulin dosing, carbohydrate intake, and glucose metabolism. *SimGlucoseEnv* depicts how glucose and insulin levels in the bloodstream are influenced by various processes such as glucose absorption, renal excretion, insulin fluxes, and insulin degradation [39]. The rationale behind choosing *SimGlucoseEnv* is: (i) as a simulation environment, it enables us to conveniently conduct large amounts of training and testing on it with different settings without concerns on data volume or balance; (ii) it has a limited number of variables, allowing us to examine the model's behavior in a more controlled setting.

In this work, we follow the Luo et al. environment setting [6]: the environment updates at 5-minute intervals, and the termination occurs if the basal plasma glucose level falls below 10 or exceeds 600. If neither condition is met, the environment continues for 24 hours (i.e., 288 steps). We set the environmental reward based on risk indices that encourage the agent to take action to reduce the risks related to hyperglycemia or hypoglycemia. The formulation of the risk function along with the entire *SimGlucoseEnv* The formulation of ODE and the descriptions of each true environment state are detailed in Appendix B.

3.4 Fine-tuning LLM in the DTR Environment using Reinforcement Learning with Environment Feedback (RLEF)

Reinforcement Learning from Environment Feedback (RLEF) inherits many heuristics of Reinforcement Learning from Human Feedback (RLHF) as they all leverage RL to optimize upon reward signals. But instead of taking a learned reward model trained on human feedback datasets as guidance, RLEF takes the reward signals generated from an outer RL environment, with which the LLM interacts, to guide policy optimization. Thus, the framework of RLEF training is a nested MDP composed of an "inner" LLM RLEF training MDP and an "outer" medical MDP to provide environment rewards based on LLM-generated clinical interventions.

Consider the fine-tuning objective, a language model π_θ with vocabulary $x \in \mathcal{V}$, state $s_t^{\mathrm{H}} = (x_1, x_2, \ldots, x_t)$, and action x_{t+1}. For the text generated by the language model $x_{1:T} \sim \pi_\theta$ sampled from the next token prediction distribution of the policy $\pi_\theta(x_{t+1}|s_t^{\mathrm{H}})$, its quality is not estimated by a learned reward model based on collected human feedback but by the "outer" returned reward of the medical MDP, denoted by $R_{Env}(x_{1:t})$. The language model generated text could be directly used as a policy to interact with the medical MDP, or it can also participate in medical DTR indirectly (e.g., in Refiner [29] and Retroformer[14], a language model is used to generate verbal feedback for the actor model).

Therefore, the primary objective in the RLEF can be formalized as follows:

$$\pi_\theta^* = \arg\max_{\pi_\theta} \mathbb{E}_{x_{1:T}\sim\pi_\theta}\left[\sum_{t=1}^{T} \gamma^{t-1} R_{Env}(x_{1:t})\right] \tag{5}$$

In this work, in order to embed the RLEF fine-tuning method to *SimGlucoseEnv* RL tasks as well as incorporating the fine-tuned objective's generated text $x_{1:t}$ with environment rewards R_{Env}, we introduce the bilevel-LLM RLEF architecture (denoted as **LLM-RLEF agent**), which contains two pre-trained LLMs for RL decision making: **ActorLM** and **InstructLM**, as shown in Figure 3. This architecture was inspired by Retroformer [14], which introduced a verbal feedback model that assists the decision-making of the actor model, showing distinct improvement for LLM's performance in decision-making related tasks. ActorLM acts as a decision maker with considerable capabilities in prior knowledge embedding and reasoning, while InstructLM is designed to extract specific dynamic characteristics of the MDP environment and summarize decision rules from time-series interaction history with *SimGlucoseEnv*, providing expert prompts to instruct ActorLM toward better decision making.

In practice, the size of ActorLM (with 7B parameters) is larger than InstructLM (with 0.5B parameters), and ActorLM is set to be frozen for the following reasons: (i) this large pre-trained model already contains strong zero-shot reasoning capability; (ii) fine-tuning on this model is too costly to conduct, diminishing the framework's advantage on sampling efficiency; (iii) inappropriate fine-tuning approaches might also ruin ActorLM's originally powerful general capability, causing

"catastrophic forgetting" [40] or over-fitting on restricted tasks. Thus, we try to improve ActorLM's performance not by fine-tuning ActorLM but by learning a better prompt. In recent years, many learning-free methods have been proposed to create more effective prompts, such as prompt tuning [41]. However, these approaches do not fundamentally improve model performance, and the prompts themselves cannot be generalized to other tasks.

For this consideration, we set the much smaller InstructLM LoRA-tunable [42], thus: (i) the model after optimizing for this specific RL task could generalize well on broader and more diverse tasks; (ii) using the much smaller model as a fine-tuning objective could avoid the high costs associated with extensive training.

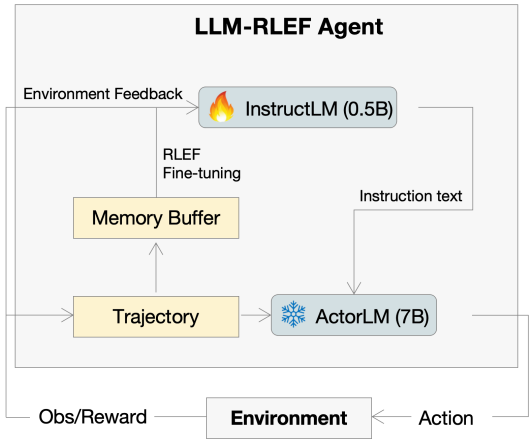

Figure 3: **Diagram of LLM-RLEF Agent in DTR Environment.** The frozen ActorLM (labeled with a snowflake) acts as the direct policy to interact with the DTR environment, while the tunable InstructLM (labeled with a flame) leverages the trajectory buffer to RLEF fine-tune itself, aiming to generate better instructions in assisting of ActorLM's decision-making.

Therefore, in the scope of learning RL, as shown in Figure 1, the sequential interaction between ActorLM and *SimGlucoseEnv* could be regarded as a text generation task for an action a at time t:

$$a_t = \pi_{\theta_A}(\mathcal{S}_A, \mathcal{H}_{t-T}^t, \mathcal{P}), \tag{6}$$

where π_{θ_A} denotes the policy of ActorLM parameterized by θ_A, $\mathcal{H}_{t-T}^t = (s_t, a_{t-1}, s_{t-1}, ..., a_{t-T}, s_{t-T})$ denotes T-step time-series interaction history with *SimGlucoseEnv*, and $\mathcal{S}_A$ and $\mathcal{P}$ denote ActorLM's system prompt (containing domain-specific prior knowledge and patient's metadata) and instruction prompt generated by InstructLM, respectively. The process of InstructLM in generating instructions is defined as

$$\mathcal{P} = \pi_{\theta_I}(\mathcal{S}_I, \mathcal{H}_{t-T}^t), \tag{7}$$

as shown in Figure 1, where π_{θ_I} denotes the policy of InstructLM parameterized by θ_I, and $\mathcal{S}_I$ denotes InstructLM's system prompt. The overall RL optimization objective for training LLM-RL^{2}EF could be formulated as:

$$\begin{aligned} &\arg\max_{\theta_I} \sum_{t=1}^{\infty} \mathbb{E}_{s_t \sim P(s_{t-1}, a_{t-1})} \left[\gamma^t R(s_t, \pi_{\theta_A}(\mathcal{S}_A, \mathcal{H}_{t-T}^t, \mathcal{P}))\right] \\ &\text{subject to } \mathcal{P} = \pi_{\theta_I}(\mathcal{S}_I, \mathcal{H}_{t-T}^t). \end{aligned} \tag{8}$$

In practice, LLM-RLEF is trained in a DTR environment (*SimGlucoseEnv*) with PPO RLEF or GRPO RLEF. The reward generated after each step of interaction with DTR is assigned to the last token of the generated sequence. RLEF with the GRPO-manner training algorithm is formulated as an Algorithm 1. Detailed designs of system prompts for ActorLM and InstructLM are listed in Appendix C.

Algorithm 1 RLEF via GRPO

Input: initial InstructLM policy $\pi_{\theta_{\text{init}}}$; instruction generating prompts $\mathcal{D}$; hyper-parameters ϵ, β, μ

Output: InstructLM policy π_θ
for iteration = 1, ..., I **do**
  Sample a batch $\mathcal{D}_b$ from RL buffer $\mathcal{D}$
  Update the old InstructLM policy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|q)$ for each prompt $q \in \mathcal{D}_b$
  Instruct ActorLM to interact with DTR to get environment reward $\{r_i\}_{i=1}^{G}$ for each sampled output o_i.
  Compute $\hat{A}_{i,t}$ for the t-th token of o_i through group relative advantage estimation
  for GRPO iteration = 1, ..., μ **do**
    Update the InstructLM policy π_θ by maximizing the GRPO objective Equation 4
  end for
end for

4 Experimental Results

Here we introduce our experiment setup and present empirical results on both decision-making and question-answering tasks using the RLEF training framework.

4.1 Experiment Setup

Treatment Policy Evaluation on DTR Environment (Decision-making Side) For the *Simglucose* DTR environment, we set the following training mode: Each policy was only trained on one adult patient, and then underwent the final test on 8 patients (four adults and four adolescents).

Baseline policies are divided into three categories: **naive policies**, **RL-only policies** and **LLM-only policies**. Naive policies included "random-0.1" and "pulse-0.05" (defined in Appendix D), RL-only policies included DQN and PPO, while LLM-only policies contained LLM and LLM w/ self-instruct, which calls itself (the larger pretrained ActorLM) to instruct decision-making. The LLM model used was Qwen2-7B [43]. For RL policies including DQN, PPO, and LLM-RLEF, we trained 288k steps each and then underwent the final test on 8 patients each for 20 episodes. For learning-free policies, including two naive policies and two LLM-only policies, we directly underwent the final test without any training. The hyper-parameter configurations for all policies are listed in Table 5 in Appendix D. We collected each algorithm's best performance over all hyper-parameter configurations.

Language Task Benchmarking on Medical Datasets (Question-answering Side) We evaluated InstructLM's linguistic performance under the RLEF training framework on general and diabetes-related medical tasks by selecting three mainstream LLM medical knowledge evaluation benchmarks: MMLU-Med [44], MedQA [45], and MedMCQA [46] and their subsets of questions related to diabetes. The statistics of the benchmarks are shown in Table 1.

Table 1: **Medical Knowledge Evaluation Datasets**

Benchmark	Question Number	Glucose-related Question Number
MMLU-Med	272	63
MedQA	1273	194
MedMCQA	4183	144
Total	5728	401

For the LLM-RLEF agent, the ActorLM chosen was Qwen2-7B, while the InstructLM was Qwen2-0.5B [43]. ActorLM remained frozen while InstructLM was fine-tuned using LoRA [42] with rank 8. We compared InstructLM's performance after RLEF with pre-trained Qwen2-0.5B and Qwen2-1.5B [43]. LLM-RLEF was trained for a total of 11,520 steps in the environment.

4.2 Decision-making Side Results on *SimGlucoseEnv* Tasks

Table 2: **Decision-making Side Test Results.** We mark the 1st highest returns on each patient cohort in red, and the 2nd highest in blue.

Patient Cohort	Naive baselines		RL-only baselines		LLM-only baselines		LLM-RLEF
	random	pulse	DQN	PPO	LLM	LLM w/ self-instruct	
adult - mean	201.01	130.48	275.62	244.45	207.94	247.62	254.84
adult - std	±100.25	±48.34	±1.68	±18.24	±31.16	±19.20	±27.52
adolescent - mean	96.97	131.11	217.09	202.55	149.96	181.69	197.01
adolescent - std	±64.77	±30.19	±61.81	±77.02	±48.48	±55.26	±66.24
training steps	-	-	288000	288000	-	-	5760

We collect results from different policies trained on *SimGlucoseEnv*, including naive baselines, RL-only baselines, LLM-only baselines, and LLM-RLEF, as shown in Table 2. We observed that LLM-RLEF outperformed most of the baseline algorithms by a considerable margin. Compared to LLM-only baselines, LLM-RLEF surpassed not only LLM policy but also LLM w/ self-instruct, which uses a more powerful LLM (Qwen2-7B) than the original InstructLM to generate summaries and instructions for ActorLM, implying that the RLEF effectively augmented InstructLM's summarization capability during fine-tuning. LLM-RLEF shows on-par performance to RL-only baselines with only 5k steps of training. This implies that RLEF achieved much higher sampling efficiency compared to typical RL algorithms. Furthermore, for RL-only policies, we observed that with an increasing number of patients in training, the generalizability of the algorithm deteriorated significantly, while this problem was almost non-existent with LLM-based policies.

To better evaluate the robustness and generalizability of different algorithms, we counted the average return of each algorithm on the patient that performed the worst out of 8 patients in the final test set. We found that LLM-RLEF, despite being trained on "*SimGlucoseEnv-adult1*" (only one patient seen during training), still generalized well on other unseen patients in the final test.

4.3 Question-answering Side Results on MMLU-Med, MedQA and MedMCQA

Table 3: **Question-answering Side Test Results.** T_{glu} for the MedMCQA benchmark is blank because MedMCQA does not disclose the ground truth of the test set questions and only supports remote submission and evaluation. For each benchmark, we picked out and added highlights on the highest T_{all} and T_{glu} scores among 3 LLMs.

Benchmark	Qwen-2-0.5B		Qwen-2-0.5B-RLEF		Qwen-2-1.5B	
	T_{all}	T_{glu}	T_{all}	T_{glu}	T_{all}	T_{glu}
MMLU-Med	0.206	0.254	**0.217**	**0.254**	0.217	0.254
MedQA	**0.310**	0.284	0.306	0.289	0.301	**0.299**
MedMCQA	0.230	-	0.232	-	**0.267**	-

For the effect of RLEF training on the RLEF side, we benchmarked Qwen2-0.5B-RLEF, Qwen2-0.5B, and Qwen2-1.5B and collected their overall accuracy T_{all} and diabetes-related accuracy T_{glu}. The results are shown in Table 3. We observed that after training inside the LLM-RLEF agent, Qwen2-0.5B-RLEF's medical knowledge performance level maintained similar to its pre-trained version Qwen2-0.5B, with a slight drop behind Qwen2-1.5B. For the diabetes-related subset of questions, the performance of Qwen2-0.5B-RLEF was comparable to that of Qwen2-1.5B.

This demonstrates the effectiveness of LLM-RLEF framework in enhancing the foundation model's performance in domain-specific language tasks. We suppose that by leveraging the rewards of the DTR environment to guide the fine-tuning of LLM, InstructLM has learned to better summarize the relationship and potential patterns between *SimGlucoseEnv*'s meta-information and the time-series interactive history. This allows LLM to learn the regulations for blood glucose control, thus improving its performance in general medical knowledge, especially in the domains related to diabetes and insulin control.

5 Conclusions and Future Work

Our proposed bilevel-LLM framework using RLEF offers a promising approach to DTR, potentially providing the best of both worlds in terms of performance. Our framework represents a significant step towards using LLM to address the key limitations of traditional RL methods in medical DTR, including sampling inefficiency, poor generalization, and lack of interpretability. It is also the first work in the community to apply LLM-based policies in the healthcare DTR and achieve distinct improvements in various performance metrics. Moreover, it introduces a novel approach that leverages DTR environment feedback to effectively fine-tune foundation models, enhancing their performance in medical language tasks. We believe that by migrating the DTR environment to other medical RL environments or those in broader domains, we can further improve the language performance of foundation models in these realms.

For the future work of this project, we plan to: (i) further complement our experiments by testing more DTR environments such as *AhnChemoEnv* [47], *GhaffariCancerEnv* [48], and *OberstSepsisEnv* [49], and evaluate whether the results remain consistent across a broader range of foundation models; (ii) theoretically explore the potential of integrating the nested MDPs in RLEF consisting of the token-generating MDP and the DTR MDP, which we believe will further enhance the efficacy of this training paradigm.

References

[1] Isaac S Chan and Geoffrey S Ginsburg. Personalized medicine: progress and promise. *Annual review of genomics and human genetics*, 12(1):217–244, 2011.

[2] Bibhas Chakraborty and Susan A Murphy. Dynamic treatment regimes. *Annual review of statistics and its application*, 1(1):447–464, 2014.

[3] Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation. *IEEE Journal of Biomedical and Health Informatics*, 25(4):1223–1232, 2020.

[4] Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017.

[5] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[6] Zhiyao Luo, Mingcheng Zhu, Fenglin Liu, Jiali Li, Yangchen Pan, Jiandong Zhou, and Tingting Zhu. Dtr-bench: An in silico environment and benchmark platform for reinforcement learning based dynamic treatment regime. *arXiv preprint arXiv:2405.18610*, 2024.

[7] Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*, pages 77–95. Springer, 2020.

[8] Claire Glanois, Paul Weng, Matthieu Zimmer, Dong Li, Tianpei Yang, Jianye Hao, and Wulong Liu. A survey on interpretable reinforcement learning. *Machine Learning*, pages 1–44, 2024.

[9] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[10] Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. A study of generative large language model for medical research and healthcare. *NPJ digital medicine*, 6(1):210, 2023.

[11] Zabir Al Nazi and Wei Peng. Large language models in healthcare and medical domain: A review. In *Informatics*, volume 11, page 57. MDPI, 2024.

[12] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[13] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*, 2023.

[15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[16] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[17] Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences. *arXiv preprint arXiv:2402.08925*, 2024.

[18] Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: A survey and comparison. *The Journal of Machine Learning Research*, 15(1):809–883, 2014.

[19] Cameron Voloshin, Hoang M Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

[20] Shengpu Tang and Jenna Wiens. Model selection for offline reinforcement learning: Practical considerations for healthcare settings. In *Machine Learning for Healthcare Conference*, pages 2–35. PMLR, 2021.

[21] Shengpu Tang, Maggie Makar, Michael Sjoding, Finale Doshi-Velez, and Jenna Wiens. Leveraging factored action spaces for efficient offline reinforcement learning in healthcare. *Advances in Neural Information Processing Systems*, 35:34272–34286, 2022.

[22] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11):1716–1720, 2018.

[23] XiaoDan Wu, RuiChang Li, Zhen He, TianZhi Yu, and ChangQing Cheng. A value-based deep reinforcement learning model with human expertise in optimal treatment of sepsis. *NPJ Digital Medicine*, 6(1):15, 2023.

[24] Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. Deep reinforcement learning for closed-loop blood glucose control. In *Machine Learning for Healthcare Conference*, pages 508–536. PMLR, 2020.

[25] Kritib Bhattarai, Sivaraman Rajaganapathy, Trisha Das, Yejin Kim, Yongbin Chen, Alzheimer's Disease Neuroimaging Initiative, Australian Imaging Biomarkers, Lifestyle Flagship Study of Ageing, Qiying Dai, Xiaoyang Li, Xiaoqian Jiang, et al. Using artificial intelligence to learn optimal regimen plan for alzheimer's disease. *Journal of the American Medical Informatics Association*, 30(10):1645–1656, 2023.

[26] Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

[27] Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *arXiv preprint arXiv:2404.08555*, 2024.

[28] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. Adapt: As-needed decomposition and planning with language models. *arXiv preprint arXiv:2311.05772*, 2023.

[29] Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*, 2023.

[30] Rithesh Murthy, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Le Xue, Weiran Yao, Yihao Feng, Zeyuan Chen, Akash Gokul, Devansh Arpit, et al. Rex: Rapid exploration and exploitation for ai agents. *arXiv preprint arXiv:2307.08962*, 2023.

[31] Guillaume Maurice Jean-Bernard Chaslot Chaslot. Monte-carlo tree search. 2010.

[32] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[33] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022.

[34] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[35] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[37] J. Schulman. Approximating kl divergence. `http://joschu.net/blog/kl-approx.html`, 2020.

[38] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[39] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34, 2014.

[40] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[41] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[42] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[43] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[44] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[45] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[46] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022.

[47] Inkyung Ahn and Jooyoung Park. Drug scheduling of cancer chemotherapy based on natural actor-critic approach. *BioSystems*, 106(2-3):121–129, 2011.

[48] A Ghaffari, B Bahmaie, and M Nazari. A mixed radiotherapy and chemotherapy model for treatment of cancer with metastasis. *Mathematical methods in the applied sciences*, 39(15):4603–4617, 2016.

[49] Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

A Reinforcement Learning Algorithms

A.1 Classifications of RL Algorithms

Reinforcement learning (RL) algorithms can be categorized based on how the agent interacts with data and how it updates its strategy. Two primary classifications are Online RL (active RL) and Offline RL (passive RL). In Offline RL, the agent trains on a fixed dataset of pre-collected data without interacting directly with the environment. This approach is often used in scenarios where real-time interaction is not feasible. In contrast, in online RL, the agent continuously interacts with the environment, collecting data dynamically, and updating its strategy in real time.

Another important distinction in RL algorithms lies in On-policy vs Off-policy methods. On-policy algorithms update the policy the agent is currently using, while Off-policy methods allow the agent to improve its policy using data collected from other policies or past explorations. Off-policy methods, while more data-efficient, can suffer from instability and exploration challenges due to the mismatch between the current policy and the data-collecting policy.

The state value function $V^\pi : \mathcal{S} \to \mathbb{R}$ for a policy π is defined as the expected cumulative discounted reward when starting from state s and following policy π thereafter:

$$V^\pi(s) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\,\middle|\, s_0 = s\right]. \tag{9}$$

The state-action value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, also known as the Q-function, extends the notion of value to state-action pairs:

$$Q^\pi(s, a) = R(s, a) + \gamma\mathbb{E}_{s'\sim P(s'|s,a)}\left[V^\pi(s')\right], \tag{10}$$

where $P(s'|s, a)$ is the state transition probability function.

A.2 Deep Q-learning (DQN)

Deep Q-learning (DQN) is a prominent Off-policy RL algorithm that extends classical Q-learning by using deep neural networks to approximate the Q-value function. The Q-value function, denoted $Q(s, a)$, represents the expected reward when taking action a in the state s. DQN updates this function using the temporal difference (TD) learning update:

$$Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma\max_{a'} Q(s', a') - Q(s, a)\right]. \tag{11}$$

DQN optimizes this update over a batch of experience samples (s_i, a_i, r_i, s'_i) by minimizing the following loss function:

$$\arg\min_{\omega} \frac{1}{2N} \sum_{i=1}^{N} \left[Q_\omega(s_i, a_i) - \left(r_i + \gamma \max_{a'} Q(s_i', a') \right) \right]^2. \tag{12}$$

To stabilize learning, DQN introduces two key mechanisms:

1. **Experience Replay**: A memory buffer stores past experiences, and random samples from this buffer are used for training. This helps break the correlation between consecutive experiences and improves the stability of the training.

2. **Target Network**: A separate target network is maintained to provide stable Q-value estimates. This target network is updated less frequently than the main Q-network, which reduces oscillations in Q-value updates and aids in convergence.

A.3 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) is a widely-used On-policy RL algorithm based on the Policy Gradient (PG) approach and the Actor-Critic framework. PPO improves the stability of policy updates by imposing constraints on the magnitude of each update, preventing overly large steps that could destabilize learning. PPO solves the following optimization problem:

$$\arg\max_{\theta} \mathbb{E}_{s\sim\nu_{\pi_{\theta_k}}, a\sim\pi_{\theta_k}(\cdot|s)} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a) \right] \tag{13}$$

subject to a constraint on the KL divergence between the new policy π_θ and the old policy π_{θ_k}:

$$\mathbb{E}_{s\sim\nu_{\pi_{\theta_k}}} \left[D_{KL}\left(\pi_{\theta_k}(\cdot|s), \pi_\theta(\cdot|s) \right) \right] \leq \delta. \tag{14}$$

Instead of solving this constrained optimization directly, PPO approximates it using an objective function with either a penalty term:

$$\arg\max_{\theta} \mathbb{E}_{s\sim\nu_{\pi_{\theta_k}}, a\sim\pi_{\theta_k}(\cdot|s)} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a) - \beta D_{KL}\left(\pi_{\theta_k}(\cdot|s), \pi_\theta(\cdot|s) \right) \right] \tag{15}$$

or a clipping mechanism:

$$\arg\max_{\theta} \mathbb{E}_{s\sim\nu_{\pi_{\theta_k}}, a\sim\pi_{\theta_k}(\cdot|s)} \left[\min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon \right) A^{\pi_{\theta_k}}(s,a) \right) \right]. \tag{16}$$

The clipping function ensures that the policy does not change too drastically during updates, which enhances training stability.

B SimGlucoseEnv Design and Formulations

The dynamics are determined based on real-world data from 300 patients, covering a range of metabolic parameters and demographic characteristics. The dynamics are formulated in ODEs, which are developed based on computational models that simulate interactions between insulin dosing, carbohydrate intake, and glucose metabolism. The ODEs can be expressed by:

$$
\begin{cases}
\dfrac{dG_p(t)}{dt} = EGP(t) + Ra(t) - U_{ii} - E(t) - k_1 G_p(t) \\
\qquad\qquad + k_2 G_t(t) \\
\dfrac{dG_t(t)}{dt} = -U_{id}(t) + k_1 G_p(t) - k_2 G_t(t) \\
\dfrac{dX(t)}{dt} = -p_{2u} \cdot X(t) + p_{2u} \cdot [I(t) - I_b] \\
\dfrac{dI(t)}{dt} = -k_i \cdot [I'(t) - I(t)] \\
\dfrac{dX^L(t)}{dt} = -k_i [X^L(t) - I'(t)] \\
\dfrac{dS_{sto}(t)}{dt} = CHO(t) - k_{sto} \cdot S_{sto}(t) \\
\dfrac{dQ_{sto}(t)}{dt} = k_{sto} \cdot S_{sto}(t) - k_{gut} \cdot Q_{sto}(t) \\
\dfrac{dQ_{gut}(t)}{dt} = k_{gut} \cdot Q_{sto}(t) - k_{abs} \cdot Q_{gut}(t)
\end{cases} \tag{17}
$$

where $Ra(t) = f \cdot k_{abs} \cdot Q_{gut}(t)$, $E(t) = k_{e1} \cdot [G_p(t) - k_{e2}]$, $U_{id}(t) = \frac{V_{m0} + V_{mx} X(t)}{B_W (K_{m0} + G_t(t))}$, $EGP(t) = k_{p1} - k_{p2} G_p(t) - k_{p3} X^L(t)$. The variable descriptions for the SimGlucoseEnv are shown in Table 4.

Table 4: **Variables of the SimGlucoseEnv ODEs.**

Variable name	Usage	Description	Unit	Range
$G_p(t)$	O	The amount of glucose in plasma	mg/dL	(10, 600)
$G_t(t)$	S	The amount of glucose in the tissue	mg/dL	-
$I(t)$	S	The insulin concentration	U/day	-
$X(t)$	S	The insulin action on glucose utilization	-	-
$X^L(t)$	S	The delayed insulin action in the liver	-	-
$S_{sto}(t)$	S	The amount of solid carbohydrates in stomach	mg	-
$Q_{sto}(t)$	S	The amount of liquid carbohydrates in stomach	mg	-
$Q_{gut}(t)$	S	The amount of liquid carbohydrates in gut	mg	-
$Ra(t)$	S	The rate of glucose absorption in the blood	-	-
$E(t)$	S	The renal excretion of glucose	mg/dL	-
$EGP(t)$	S	The endogenous glucose production (EGP)	U/day	-
$U_{id}(t)$	S	The insulin-dependent utilization takes place in the remote compartment	-	-
$CHO(t)$	S	The amount of ingested carbohydrates	g	(0, 200)
$a(t)$	A	The insulin concentration of the insulin pump	U/h	(0, 30)

This ODE system models glucose absorption $Ra(t)$ from ingested carbohydrates $CHO(t)$, the body's glucose production $EGP(t)$, the dynamics of insulin $I(t)$, and insulin's impact on glucose utilization $X(t)$ and its delayed action in the liver $X^L(t)$. The equations track glucose concentrations in plasma $G_p(t)$ and tissue $G_t(t)$, account for renal glucose excretion $E(t)$, and quantify insulin-dependent glucose utilization $U_{id}(t)$. In addition, the model delineates the digestion process, distinguishing between the solid $S_{sto}(t)$ and liquid $Q_{sto}(t)$ carbohydrate states in the stomach before their absorption in the gut $Q_{gut}(t)$. The model directly correlates dietary intake and insulin administration with blood glucose levels through these dynamics, offering a sophisticated tool to simulate glucose-insulin interactions and aiding effective diabetes management strategies.

We set the environmental reward based on risk indices encouraging the agent to take action to reduce diabetes-related risks, formulated as follows:

$$
r_{\text{risk}}(t) = \begin{cases} -15, & \text{if } G_p(t) < 40 \\ 1 - \frac{1}{10} \left[1.509 \left(\ln(G_p(t))^{1.084} - 5.381 \right) \right]^2, & \text{otherwise.} \end{cases} \tag{18}
$$

C RLEF System Prompts

Prior Knowledge Prompt

```
"You are a clinical specialist managing patients with Type-1 Diabetes. "
Your primary objective is to maintain each patient's blood glucose levels within the range "
"of 70-180 mg/dL. "
"Blood glucose levels are observed every 5 minutes, and insulin is administered accordingly. "
"Insulin is dosed in U/min, ranging from 0 to 0.5, and is adjusted per 5 minutes. "

"[State]: We can observe the patient's blood glucose level and the insulin dose administered. "

"[Action]: Actionable drug is Basal insulin. Insulin reduces blood glucose levels, "
"but there is a time delay before its effect is observable. "
"No other drugs or insulin regimes are available. "
"Standard total daily insulin requirement is 0.4-0.6 units/kg. "
"The patient's weight is not provided."

"[Hidden variables]: Food consumption, which increases blood glucose levels, "
"is not directly observable. "
"Patients are likely to eat during the following periods: "
"Morning: 6:00-9:00, "
"Noon: 11:00-13:00, "
"Night: 17:00-19:00. "
"Occasionally, patients may consume small snacks at other times. "

"[Safety Considerations]: Hypoglycemia (low blood glucose levels) is particularly dangerous. "
"Extra caution is necessary to avoid administering excessive insulin. "
"Insulin has a long half-life, so the effects of previous doses may still be present. "
"Pay attention to the accumulated insulin dose to prevent Hypoglycemia."
```

Meta-info Prompt

```
f"[Patient]: You are treating a {age}-year-old patient with a Total Daily Insulin (TDI) "
f"requirement of {TDI:.1f} units over 24 hours. "

f"The patient's Carbohydrate Ratio (CR) is {CR}, "
f"meaning 1 unit of insulin covers {CR} grams of carbohydrate. "
f"A higher CR indicates less insulin is needed for a given amount of carbohydrates, and "
f"vice versa. "

f"The Correction Factor (CF) for this patient is {CF:.1f}, "
f"meaning 1 unit of insulin is expected to lower blood glucose by {1700/TDI:.2f} mg/dL."
```

ActorLM Instruction Prompt

```
"[Instruction]: Please generate the insulin dosage rate in U/min for the next 5 minutes. "
"Only provide a numerical value between 0 and 0.5 without any additional information."
```

ActorLM Retry Instruction Prompt

```
"Your previous answer cannot be converted to a valid action. "
"[Instruction]: Please provide a numerical value between 0 and 0.5 without any additional "
"information."
```

InstructLM Instruction Prompt

```
"[Instruction]: PLease summarize information such as indications of food intake, patient's "
"response to insulin, glucose record trend, drug dosage history, abnormal glucose signs "
"and possible misuse of insulin. "
```

```
"Summarize as much information as possible while keeping the answer short."
```

D Dataset Construction Details and Hyper-parameter Tuning Details for Training

To construct the MMLU-Med dataset, we collected 6 subjects of subsets from the full MMLU dataset: "anatomy_test", "clinical_knowledge", "college_biology", "college_medicine", "medical_genetics" and "professional_medicine", to sum up to 272 questions in the test sets.

For MedMCQA dataset, we directly took all questions from the test set and summed up to 4183 questions.

For both MMLU-Med and MedMCQA datasets, in order to filter out diabetes-control related question subsets and count LLM's performance on them separately, we filtered all questions in the 2 datasets with keywords: "diabetes", "glucose" and "insulin", resulting in 63 and 144 diabetes-control related questions in MMLU-Med and MedMCQA, respectively.

In the two naive policies "random-0.1" and "pulse-0.05", we provide a detailed definition for each them respectively. "random-0.1" means the policy gives a random amount of insulin unit, which obeys the uniform distribution of $U[0, 0.1]$ to the patient every 5 minutes, while "pulse-0.05" means the policy gives a 0.05 unit of insulin (as a pulse) to the patient every 60 minutes.

For all baselines and LLM-RL^{2}EF policy, we performed grid-search on the hyper-parameter settings as shown in Table 5.

Table 5: **Hyper-parameter Settings.**

Hyper-parameters	Naive baselines		RL-only baselines		LLM-only baselines		LLM-RLEF
	random-0.1	pulse-0.05	DQN	PPO	LLM	LLM w/ self-instruct	
seed	2732, 9845, 3264, 4859	2732, 9845, 3264, 4859	2732, 9845, 3264, 4859	2732, 9845, 3264, 4859	2732, 9845, 3264, 4859	2732, 9845, 3264, 4859	2732, 9845, 3264, 4859
lr	-	-	3e-3, 1e-3, 3e-4	3e-3, 1e-3, 3e-4	-	-	1e-4
batch_size	-	-	256	256	-	-	8
gamma	-	-	0.99	0.99	-	-	0.99
step_per_collect	-	-	1, 100	288	-	-	288
obs_mode	-	-	cat, stack	cat, stack	-	-	-
n_step	-	-	1	1	-	-	1
target_update_frequency	-	-	0, 200	-	-	-	-
is_double	-	-	True, False	-	-	-	-
eps_test	-	-	0.001	-	-	-	-
eps_train	-	-	0.1	-	-	-	-
eps_train_final	-	-	0.1	-	-	-	-
gae_lambda	-	-	0.001	-	-	-	-
vf_coef	-	-	0.001	-	-	-	-
ent_coef	-	-	0.001	-	-	-	-
eps_clip	-	-	0.001	-	-	-	-
num_try	-	-	-	-	2	2	2
total count	4	4	192	24	4	4	4

E Hardware Configuration Details for Training

For training the LLM-RLEF agent, we leveraged 1 * H800 with 2 Intel Xeon Gold 6430 32C 2.1GHz 60MB 270W CPUs for 3 days on each hyper-parameter configuration. The details of the implementation can be found in Section 4.1 or the full implementation of the code in the supplemental materials.

F Code and Data Availability

All the codes and datasets relevant to this project are available anonymously in supplemental materials, which contain two branches of codes representing the implementations of the PPO and GRPO RLEF framework.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction of the paper clearly state all the contributions made in the paper and scope of the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 5, we discuss the limitations of the work, which is the lack of more parallel experiments and the future direction of theoretically integrating the nested MDPs in the current framework.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical proofs, but we rigorously formulate the theoretical representation and objective of Reinforcement Learning with Environment Feedback, which is our newly proposed algorithmic framework.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper. In Section 4.1 and the complete codes in supplemental materials, readers can refer to the detailed descriptions and code repository to reproduce all the experimental results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the codes and datasets relevant to this project is available anonymously in supplemental materials, which contains two branches of codes representing PPO and GRPO RL^2EF framework implementations.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper fully specify all the training and test details needed to understand the experimental results. In Section 4.1, readers can refer to the detailed descriptions to understand experimental settings or details.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The error bars are not reported because it would be too computationall expensive, as referred to in Appendix E. We will try to make up with these statistics in the camera-ready version of the paper.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Appendix E, we provide sufficient information on the computer resources needed to reproduce the experiments.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: there is no societal impact of the work performed, because all the methods and experiments in this project are based on simulation environments and virtual datasets, we perform neither crowdsourcing nor research with human subjects.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original owners of all the code, data or models used in the paper, are properly credited in the main text. We also construct the license and terms of use explicitly in the code repository.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.