Using LLMs to improve RL policies in personalized health adaptive interventions

Karine Karine
University of Massachusetts Amherst
karine@cs.umass.edu

Benjamin M. Marlin
University of Massachusetts Amherst
marlin@cs.umass.edu

Abstract

Reinforcement learning (RL) is increasingly used in the healthcare domain, particularly for the development of personalized adaptive health interventions. However, RL methods are often applied to this domain using small state spaces to mitigate data scarcity. In this paper, we aim to use Large Language Models (LLMs) to incorporate text-based user preferences and constraints, to update the RL policy. The LLM acts as a filter in the action selection. To evaluate our method, we develop a novel simulation environment that generates text-based user preferences and incorporates corresponding constraints that impact behavioral dynamics. We show that our method can take into account the text-based user preferences, while improving the RL policy, thus improving personalization in adaptive intervention.

1 Introduction

Reinforcement learning (RL) is increasingly used in the healthcare domain, particularly for the development of personalized adaptive health interventions (Coronato et al., 2020; Liao et al., 2020; Gönül et al., 2021; Yu et al., 2021; Spruijt-Metz et al., 2022; Karine et al., 2024). However, RL methods are often applied to adaptive intervention problems using small state spaces to mitigate the data scarcity that results from practical limitations on adaptive intervention trial designs, including limited numbers of participants, limited numbers of interventions per day, and limited study durations.

Moreover, there can be issues in the decision rule or policy that result in incorrectly contextualized messages sent to the participant (e.g., user preference not aligning with the policy). These messages may annoy the participant or cause participant disengagement. Therefore, it is critical to consider participant preferences before it is too late or irreversible (e.g., the participant exits the study).

One solution to prevent disengagement is to allow the participant to specify their preferences in the form of free-text descriptions and immediately take them into account to influence the action selection. This is especially relevant in today's generation, where people use chats and social media to communicate. For example, the user preference can be: "I twisted my ankle" or "my leg is sore". The user can enter their preference in a daily survey in the mobile health app.

In this paper, we explore leveraging the natural language understanding ability and reasoning capabilities of Large Language Models (LLMs) to influence RL action selection based on participant descriptions of preferences. We evaluate an approach where an RL agent proposes a candidate action at each time step. Next, given the text-based participant preference, we use the LLM to decide whether the candidate action (sending one of several message types message) should be allowed or not allowed. The LLM is used as a filter in the action selection with the goal of better aligning the RL policy with the user preferences and constraints. We use Thompson sampling as a data-efficient base RL algorithm (see Appendix A.2 for relevant background). We refer to the resulting method as LLM+TS.

To evaluate our approach, we build on a recently introduced simulation environment for an adaptive messaging physical activity intervention that simulates key aspects of behavioral dynamics including intervention habituation and disengagement risk (Karine and Marlin, 2024). We add to this system a simulation of participants responding to a daily query about their general health state. We generate the responses based on the true underlying health state of the simulated participant, and incorporate constraints that impact behavioral dynamics.

Our preliminary results show that different families of LLMs reason about the simulated participant preferences with different accuracies, but that using

Proceedings of the Second Workshop on Patient-Oriented Language Processing (CL4Health), pages 137–147

any of the evaluated LLMs results in improved performance relative to standard Thompson Sampling. We explore the effect of leveraging intermediate reasoning and domain-specific knowledge within the prompt, mirroring promising LLM approaches such as chain-of-thought reasoning and retrieval-augmented generation (Zheng et al., 2023; Wei et al., 2022; Lewis et al., 2020).

Our contributions are:

1. **LLM+TS.** We introduce an "LLM as judge" approach to enhancing personalized adaptive health interventions. LLM+TS leverages the natural language understanding and reasoning capabilities of LLMs to improve the limited state representation of a Thompson Sampler, while maintaining data efficiency and providing intervention designers with better control over intervention content. This is a promising approach for significantly augmenting the intelligence of personalized adaptive health interventions. We provide an overview of our method in Figure 1.

2. **StepCountJITAI for LLM.** We create a novel simulation environment to evaluate the proposed method. Our simulation environment extends an existing base simulator to add the support for LLMs. It generates text-based user preferences and incorporates constraints that impact behavioral dynamics. Our simulation environment has significant potential to enable the development of new RL algorithms for adaptive interventions that incorporate text-based user preferences.

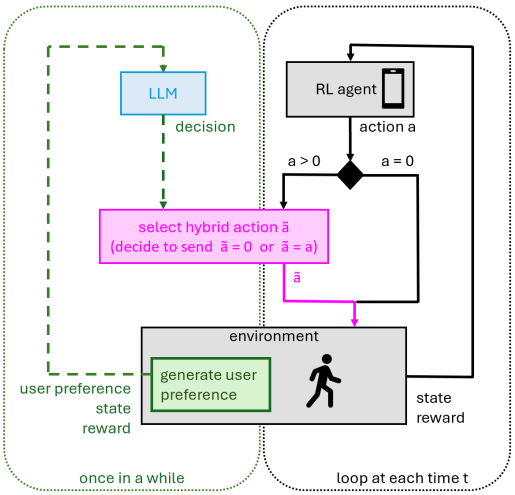


Figure 1: Overview of the LLM+TS method. LLM+TS is a hybrid method that combines LLM inference and RL policy learning to improve action selection. The RL agent proposes a candidate action a. The LLM prompt that is used to guide inference includes a description of the behavioral dynamics and the participant preferences along with questions that prompt chain of thought-like reasoning. Finally, the prompt asks the LLM to decide whether the candidate action (sending one of several message types message) should be allowed or not allowed (i.e., $\tilde{a} = 0$ or $\tilde{a} = a$). Thus, the LLM acts as a judge, filtering the candidate actions.

2 Background

We describe the base simulator below and provide more details in Appendix A.1. We also provide the background on Thompson Sampling in Appendix A.2, and related work in Appendix A.3.

StepCountJITAI: an adaptive physical activity simulation environment. There is limited prior work on simulation environments for adaptive interventions in the literature. In this work, we extend the base physical activity adaptive intervention simulator introduced in Karine and Marlin (2024). This base simulator was specifically designed to support the development of new RL algorithms applicable to the adaptive intervention domain.

A messaging-based physical activity adaptive intervention can be framed as an RL system. In this simulation environment, the state includes a context variable $c_t \in \{0, 1\}$ that can model a binary state such as 'stressed / not stressed' or 'at home / not at home,' etc. at each time t. The simulation also models the dynamics of two key behavioral state variables: habituation level h_t and disengagement risk level d_t. The different types of messages that can be sent to a participant are the possible actions. The variable a_t denotes the action at time t. The possible actions a_t are:

- $a_t = 0$ (do not send a message)
- $a_t = 1$ (send a generic message)
- $a_t = 2$ (send a message tailored to context 0)
- $a_t = 3$ (send a message tailored to context 1)

The **goal** in this domain is to **maximize** the participant's total walking step count over the duration of the intervention. Thus, **step count** serves as the **reward** r_t. Further details of the base simulator are described in Appendix A.1.2.

However this base simulator does not include the support for LLMs. Thus, we extend the base simulator to create a simulation environment that includes the support for LLMs. We describe this novel simulation environment in Section 3.2.

3 Methods

In this section, we describe our proposed method as well as our novel simulation environment. Figure 1 provides an overview of the proposed method.

3.1 Proposed Approach: LLM+TS

We propose a hybrid method where the RL agent outputs a candidate action at each time step. Then, based on the LLM prompt that includes the user preference and other information, the LLM decides whether to allow or not allow the RL candidate action. We summarize the method below.

1. Candidate Action Generation: At each time step t, the RL agent proposes a candidate action a_t based on its current parameters θ_t and the current state s_t. If the candidate action is $a_t = 0$, set $\tilde{a}_t = 0$. No message is sent. If the action is $a_t \neq 0$, apply LLM inference.
2. LLM Inference: Given the current user preference and other context information, construct the LLM prompt. Apply an LLM to perform inference given the prompt. Extract the decision from the LLM response.
3. Action Filtering: If the LLM decision is to "not send" a message, set $\tilde{a}_t = 0$. Otherwise, set $\tilde{a}_t = a_t$.
4. Policy Update: Take the action $\tilde{a}_t$. Observe the reward r_t and new state s_{t+1}. Update the RL agent's parameters based on the tuple $(s_t, \tilde{a}_t, r_t)$, obtaining θ_{t+1}.

We note that if the RL agent proposes the candidate action $a_t > 0$ (indicating a candidate message to be sent), then the LLM is prompted to decide if this message should actually be sent or not. If the RL agent proposes the candidate action $a_t = 0$ (indicating no message) or if no user preference was generated, then there is no need to call LLM inference, so the RL loop continues as usual. We note that the RL agent does not have knowledge of the text-based user preferences.

We construct the LLM prompt by including a description of the specific adaptive intervention domain, the hypothesized behavioral dynamics, intermediate reasoning questions to guide the LLM, a statement of the user preferences, and a final question asking the LLM to make a decision to "send" or "not send" a message. We provide an example of a constructed LLM prompt in Appendix B.1.

To evaluate the proposed method, we create a simulation environment to generate the text-based user preferences and incorporate additional latent physical health states as described in the next section. Importantly, the LLM inference step used to filter action selection is completely separated from the application of LLMs to simulate participant generation of text descriptions of preferences. In a real-world application of the proposed method, the preference text would, of course, be generated by the participant via an intervention app.

3.2 StepCountJITAI for LLM

We extend the base simulator introduced in Karine and Marlin (2024) to create a new simulation environment that generates participant preferences and constraints conditioned on an additional state dimension that is not observable by the RL agent. Specifically, we introduce a new state variable $w_t \in \{0, 1\}$ indicating whether the user is able to walk or not.

We implement the dynamics for w_t using a Markov chain where the value for w_t is sampled conditioned on w_{t-1}. This allows "can walk" and "cannot walk" states to persist for different average lengths of time. These dynamics are described in detail in Appendix Figure 4 and Table 3.

We use two different LLM prompts to simulate the generation of participant text conditioned on the variable w_t. When transitioning from $w_{t-1} = 1$ to $w_t = 0$, we emit text produced by prompting the LLM to generate a short description of a reason why a person might not be able to walk. When transitioning from $w_{t-1} = 0$ to $w_t = 1$, we emit text produced by prompting the LLM to generate a message describing that the participant is "feeling fine." When staying in the $w_t = 1$ state, we emit a new participant preference statement with probability 0.3. We provide further details on LLM-based user preference generation in Appendix B.

When in the $w_t = 0$ or "cannot walk" state, we modify the behavioral and reward dynamics accordingly. First, if $w_t = 0$ and $\tilde{a}_t \neq 0$, the disengagement risk d_t is incremented regardless of whether the tailoring of the action was correct or not. This simulates the idea that a participant might lose significant trust in the system and be more likely to disengage from using it if walking suggestions continue to be issued despite the fact that the participant indicates a reason for not being able to walk. Second, we set the reward to $r_t = 0$ if $w_t = 0$, consistent with the idea that the participant accumulates no reward (i.e., no step) if they can not walk. The dynamics are given in Appendix B.3.1.

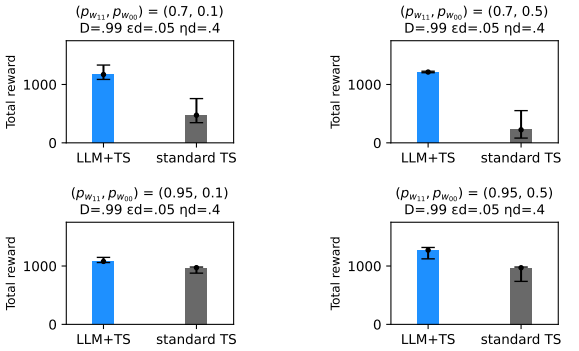


Figure 2: Example scenarios showing that LLM+TS outperforms standard TS. (top) Scenario 1: $p_{w_{11}} = 0.7$ (probability of staying in state "can walk") and various $p_{w_{00}}$ (probability of staying in state "cannot walk"). (bottom) Scenario 2: $p_{w_{11}} = 0.95$ and various $p_{w_{00}}$.

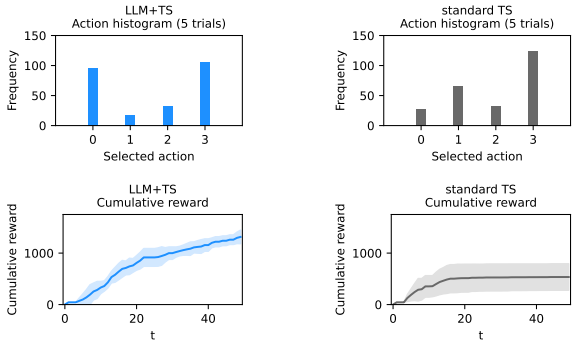


Figure 3: LLM+TS vs. standard TS. Example histograms of all selected actions (top), and plots of average cumulative reward per episode for $(p_{w_{11}}, p_{w_{00}}) = (0.7, 0.5)$, $\epsilon_d = 0.01, \eta_d = 0.05$.

4 Experiments

We conduct experiments to validate the LLM responses and compare our method to standard TS.

Validating LLM Inference. We perform experiments evaluating the ability of different LLMs to correctly classify preference statements as implying that the participant can or cannot walk. We found average inference accuracies of 0.86 for Gemma 2, 0.87 for Llama 3 8B and 0.98 for Llama 3 70B. Details are provided in Appendix C.1.

Validating LLM+TS. We conduct extensive experiments to compare LLM+TS to standard Thompson Sampling (TS). Both LLM+TS and TS use the same TS state space that does not include access to the w_t state variable. However, LLM+TS performs inference over the text of user preferences as described previously. We generate results by varying the probability of remaining in the "cannot walk" state $p_{w_{00}}$ and the probability of remaining in the "can walk" state $p_{w_{11}}$. We show results for two realistic scenarios: Scenario 1, where $p_{w_{11}} = 0.7$, and Scenario 2, where $p_{w_{11}} = 0.95$. In both scenarios, $p_{w_{00}}$ varies in the range $[0.1, ..., 0.5]$. We plot the median total reward, with the 25th and 75th percentiles, over 5 trials in Figure 2. We see that when there is a higher probability that the participant is in the "cannot walk" state, LLM+TS significantly outperforms TS, as expected. More details and results are provided in Appendix C.2.

Analysis of Selected Actions. We compare the histograms of selected actions, taking into account all actions selected by each method across 5 trials. The histograms show that LLM+TS selects more $a_t = 0$ actions, which indicates that the LLM has correctly decided to "not send" a message when the user cannot walk. We also compare the average cumulative reward per episode in Figure 3, which suggests that the average episode length for TS is significantly lower than for LLM+TS due to early disengagements. Additional results are provided in Appendix C.

5 Conclusion

We introduce LLM+TS, an "LLM as judge" approach to enhancing personalized adaptive health interventions. LLM+TS leverages the natural language understanding and reasoning capabilities of LLMs to improve the limited state representation of a Thompson Sampler, while maintaining data efficiency and providing intervention designers with better control over intervention content. To evaluate our method, we introduce StepCountJITAI for LLM, a novel simulation environment that generates user preferences and incorporates constraints that impact behavioral dynamics. Our results show that LLM+TS is a promising approach for significantly augmenting the intelligence of personalized adaptive health interventions. Our novel simulation environment has significant potential to enable the development of new RL algorithms for adaptive interventions that incorporate text-based user preferences.

Limitations

The proposed method was evaluated on selected LLMs at this time. Other LLMs could be used depending on available resources. Future work will involve inserting additional insights into the LLM prompt or using advanced LLMs to further improve the LLM inference accuracy.

Acknowledgments

This work is supported by National Institutes of Health National Cancer Institute, Office of Behavior and Social Sciences, and National Institute of Biomedical Imaging and Bioengineering through grants U01CA229445, 1P41EB028242 and P30AG073107.

References

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings.

Antonio Coronato, Muddasar Naeem, Giuseppe De Pietro, and Giovanni Paragliola. 2020. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109:101964.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023. Guiding Pretraining in Reinforcement Learning with Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 8657–8677.

K. J. Kevin Feng, Xander Koo, Lawrence Tan, Amy Bruckman, David W. McDonald, and Amy X. Zhang. 2024. Mapping the Design Space of Teachable Social Media Feed Experiences. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, page 890–896.

Gemma Team. 2024. Gemma: Open Models Based on Gemini Research and Technology. *arXiv:2403.08295*.

Suat Gönül, Tuncay Namlı, Ahmet Coşar, and İsmail Hakkı Toroslu. 2021. A reinforcement learning based algorithm for personalization of digital, just-in-time, adaptive interventions. *Artificial Intelligence in Medicine*, 115:102062.

Karine Karine, Predrag Klasnja, Susan A. Murphy, and Benjamin M. Marlin. 2023. Assessing the impact of context inference error and partial observability on RL methods for Just-In-Time Adaptive Interventions. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216, pages 1047–1057.

Karine Karine and Benjamin M. Marlin. 2024. StepCountJITAI: simulation environment for RL with application to physical activity adaptive intervention. In *Workshop on Behavioral Machine Learning, Advances in Neural Information Processing Systems*.

Karine Karine, Susan A. Murphy, and Benjamin M. Marlin. 2024. BOTS: Batch Bayesian Optimization of Extended Thompson Sampling for Severely Episode-Limited RL Settings. In *Workshop on Bayesian Decision-making and Uncertainty, Advances in Neural Information Processing Systems*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. 2020. Personalized HeartSteps: A Reinforcement Learning Algorithm for Optimizing Physical Activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22.

Llama Team. 2024. The Llama 3 Herd of Models. *arXiv:2407.21783*.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Ren Chen Si Zhang, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. LLM-Rec: Personalized Recommendation via Prompting Large Language Models. In *North American Association for Computational Linguistics*.

Sheshera Mysore, Mahmood Jasim, Andrew McCallum, and Hamed Zamani. 2023. Editable User Profiles for Controllable Text Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 993–1003.

Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2018. A Tutorial on Thompson Sampling. *Found. Trends Mach. Learn.*, 11(1).

Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large Language Models are Competitive Near Cold-start Recommenders for Language- and Item-based Preferences. In *Proceedings of the 17th ACM Conference on Recommender*, page 890–896.

Donna Spruijt-Metz, Benjamin M Marlin, Misha Pavel, Daniel E Rivera, Eric Hekler, Steven De La Torre, Mohamed El Mistiri, Natalie M Golaszweski, Cynthia Li, Rebecca Braga De Braganca, et al. 2022. Advancing behavioral intervention and theory development for mobile health: the HeartSteps II protocol. *International journal of environmental research and public health*, 19(4):2267.

William R. Thompson. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. In *Biometrika*, volume 25, pages 285–294.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

A Background and Related Work

We provide the background on StepCountJITAI and Thompson Sampling, and the related work.

A.1 StepCountJITAI simulation environment

The base simulator introduced in Karine et al. (2023); Karine and Marlin (2024) mimics a participant's behaviors in a mobile health study, where the interventions (actions) are the messages sent to the participant, with the goal of increasing the participant walking step count (reward), given the participant's context and behaviors (states). We summarize the base simulator specifications in Tables 1 and 2, and provide details below.

A.1.1 StepCountJITAI specifications

For the notation, we use an uppercase letter for the variable name, and a lowercase letter for the variable value, for example: the context variable C has value $c_t = 0$ at time t.

Below we describe some of the simulation environment variables and parameters that are used in the behavioral dynamics: c_t is the true context, p_t is the probability of context 1, l_t is the inferred context, h_t is the habituation level, d_t is the disengagement risk, s_t is the step count (s_t is the participant's walking step count), and a_t is the action at time t. The base simulator also includes behavioral parameters: δ_d and ϵ_d are decay and increment parameters for the disengagement risk, and δ_h and ϵ_h are decay and increment parameters for the habituation level.

The goal is to increase the participant's walking step count. Thus, the walking step count is also the RL reward.

Action	Description
$a = 0$	No message is sent to the participant.
$a = 1$	A non-contextualized message is sent.
$a = 2$	A message customized to context 0 is sent.
$a = 3$	A message customized to context 1 is sent.

Table 1: Possible action values

Variable	Description	Values
c_t	true context	$\{0, 1\}$
p_t	probability of context 1	$[0, 1]$
l_t	inferred context	{0,1}
d_t	disengagement risk level	$[0, 1]$
h_t	habituation level	$[0, 1]$
s_t	step count	$\mathbb{N}$

Table 2: State variables

We use the same default parameter values as in the base simulator: context uncertainty $\sigma = 0.4$, behavioral parameters $\delta_h = 0.1$, $\epsilon_h = 0.05$, $\delta_d = 0.1$, $\epsilon_d = 0.4$, $m_s = 0.1$, $\rho_1 = 50$, $\rho_2 = 200$. For our experiments, we set the disengagement threshold $D_{threshold} = 0.99$. The maximum study length is 50 days, with daily data. We describe the behavioral dynamics below, in Appendix A.1.2.

A.1.2 StepCountJITAI behavioral dynamics

The behavioral dynamics are as follow: Sending a message causes the habituation level to increase. Not sending a message causes the habituation level to decrease. An incorrectly tailored message causes the disengagement risk to increase. A correctly tailored message causes the disengagement risk to decrease. When the disengagement risk exceeds a given threshold, the behavioral study ends. The reward is the surplus step count, beyond a baseline count, attenuated by the habituation level.

These behavioral dynamics can be translated into equations:

$$c_{t+1} \sim Bernoulli(0.5), \quad x_{t+1} \sim \mathcal{N}(c_{t+1}, \sigma^2) \tag{1}$$

$$p_{t+1} = P(C = 1|x_{t+1}), \quad l_{t+1} = p_{t+1} > 0.5 \tag{2}$$

$$h_{t+1} = \begin{cases} (1 - \delta_h) \cdot h_t & \text{if } a_t = 0 \\ \min(1, h_t + \epsilon_h) & \text{otherwise} \end{cases} \tag{3}$$

$$d_{t+1} = \begin{cases} d_t & \text{if } a_t = 0 \\ (1 - \delta_d) \cdot d_t & \text{if } a_t \in \{1, c_t + 2\} \\ \min(1, d_t + \epsilon_d) & \text{otherwise} \end{cases} \tag{4}$$

$$s_{t+1} = \begin{cases} m_s + (1 - h_{t+1}) \cdot \rho_1 & \text{if } a_t = 1 \\ m_s + (1 - h_{t+1}) \cdot \rho_2 & \text{if } a_t = c_t + 2 \\ m_s & \text{otherwise} \end{cases} \tag{5}$$

where σ is the context uncertainty, x_t is the context feature, $\sigma, \rho_1, \rho_2, m_s$ are fixed parameters. We use the same default parameter values as the base simulator, which we summarize in Appendix A.1.1.

A.2 Thompson Sampling

Thompson Sampling (TS) is a probabilistic method for decision-making under uncertainty. It can be used to address contextual multi-armed bandit problems (Russo et al., 2018; Chu et al., 2011; Thompson, 1933).

Typical TS for contextual bandit settings uses a reward model of the form $\mathcal{N}(r; \theta_a^\top v_t, \sigma_{Ya}^2)$, where v_t is the state vector at time t, θ_a is a vector of weights, and σ_{Ya}^2 is the reward variance for action a. Thus, $\theta_a^\top v_t$ represents the mean reward for action a.

The reward model weights θ_a are random variables of the form $\mathcal{N}(\theta_a; \mu_{ta}, \Sigma_{ta})$. Actions are selected at each time t by sampling $\hat{\theta}_a$ from $\mathcal{N}(\theta_a; \mu_{ta}, \Sigma_{ta})$ and choosing the action with the largest value $\hat{\theta}_a^\top v_t$. The prior distribution for θ_a is of the form $\mathcal{N}(\theta_a; \mu_{0a}, \Sigma_{0a})$. The distribution over θ_a for the selected action is updated at time t based on the observed reward r_t and v_t using Bayesian inference. We provide the update equations for the mean and covariance matrix below.

$$\Sigma_{(t+1)a} = \sigma_{Ya}^2 \left(v_t^\top v_t + \sigma_{Ya}^2 \, \Sigma_{ta}^{-1} \right)^{-1} \tag{6}$$

$$\mu_{(t+1)a} = \Sigma_{(t+1)a} \left((\sigma_{Ya}^2)^{-1} \, r_t \, v_t + \Sigma_{ta}^{-1} \, \mu_{ta} \right) \tag{7}$$

A.3 Related work

Recent works use LLMs in RL, where the RL agent selects actions based on natural language inputs, and apply to games (Du et al., 2023). Note that in our work, we leverage LLMs as foundational models and focus on online decision-making for episode-limited RL settings. Recent research on RL from human feedback, and from AI feedback, typically require some form of reward modeling, and a large number of episodes to perform well. Other works have also explored using natural language inputs, but apply to recommender systems for items such as movies, social media, recommendation algorithms (Lyu et al., 2024; Feng et al., 2024; Mysore et al., 2023; Sanner et al., 2023). However, these approaches also require a large number of iterations to work well. In contrast, we use Thompson Sampling which is a Bayesian approach that can perform well in a lower number of iterations than typical deep RL methods.

Recent works use LLM as a judge, intermediate reasoning and retrieval-augmented generation, to generate better LLM responses (Zheng et al., 2023; Wei et al., 2022; Lewis et al., 2020). We use similar ideas, but focus on creating a single LLM prompt, where the LLM makes a decision, based on the user preference and reasoning in the prompt.

B Method details

We first provide an example of an LLM prompt that is used in our method, as described in Section 3. Then, we provide further details about our novel environment simulator that supports LLMs.

B.1 Example of LLM prompt

In our new method, the LLM prompt contains the following blocks of text (description of behavioral dynamics, participant preference, reasoning), as described in Section 3.1.

Example of LLM prompt.

```
A mobile health app can send a message to the user to encourage the user to walk.
...
Sending a message causes the habituation level to increase.
Not sending a message causes the habituation level to decrease.
An incorrectly tailored message causes the disengagement risk to increase.
A correctly tailored message causes the disengagement risk to decrease.
If the user is sick, injured or cannot walk, then the mobile health app should not send a message.
...
This morning, when we asked the user how they felt, the user reply was: "I twisted my ankle".
...
Given the user reply, answer the following questions:
provide the reason for sending a message,
provide the reason for not sending a message,
is there any risk to the user?
will the user disengage from the study?
is there some long term consequence?
...
Given these answers, provide the final answer to this question: should the mobile health app send a message to the user?
```

We detail the `text in purple`. The text for the

user reply (e.g., "I twisted my ankle") is chosen randomly from the lists provided in Appendix B.3.

B.2 Creating auxiliary variable W (cannot walk / can walk)

We first augment the simulation environment states with a binary state variable W with value: 0 "cannot walk" or 1 "can walk". The variable W is not observed by the RL agent. It reflects a hidden state of the user, and is used to generate the user preference, and trigger the constraints. We implement a Markov chain to simulate w_t, the values of W at time t. The Markov chain sketch and transition function for W are shown in Figure 4 and Table 3.

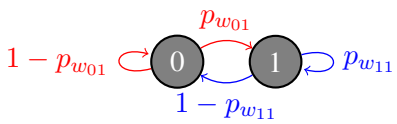


Figure 4: Markov chain sketch.

w_t	w_{t+1}	$P(w_{t+1}\|w_t)$
0	0	$1 - p_{w_{01}}$
0	1	$p_{w_{01}}$
1	0	$1 - p_{w_{11}}$
1	1	$p_{w_{11}}$

Table 3: Transition Function.

We define the new parameters: $p_{w_{01}}$ the probability of transitioning from $w_t = 0$ to $w_{t+1} = 1$, and $p_{w_{11}}$ the probability of remaining in the "can walk" state.

$$p_{w_{01}} = P(w_{t+1} = 1 | w_t = 0) \quad (8)$$
$$p_{w_{11}} = P(w_{t+1} = 1 | w_t = 1) \quad (9)$$

Setting $p_{w_{11}}$ to a lower (or higher) value allows for a lower (or higher) probability of remaining in the "can walk" state. Similarly, setting $p_{w_{01}}$ to a lower (or higher) value allows for a lower (or higher) probability of transitioning from $w_t = 0$ to $w_{t+1} = 1$.

We note that the parameters $p_{w_{01}}$ and $p_{w_{11}}$ can be used to simulate the user state "cannot walk" over a variety of ranges, from shorter to longer time intervals, and thus enabling a variety of scenarios for our experiments.

In Section 4, we run our experiments and show the results for two realistic scenarios: Scenario 1, where $p_{w_{11}} = 0.7$, and Scenario 2, where $p_{w_{11}} = 0.95$. In both scenarios, $p_{w_{00}}$ varies in the range $[0.1, ..., 0.5]$, where $p_{w_{00}} = 1 - p_{w_{01}}$.

B.3 Generating a text-based user preference "cannot walk".

Following the Markov chain and transition function in Figure 4 and Table 3, W can take values 1 "can walk" or 0 "cannot walk".

When W transitions from 1 "can walk" to 0 "cannot walk", a user preference is randomly chosen from a list of pre-defined reasons for "cannot walk". The "cannot walk" list was previously created by asking ChatGPT to give reasons why a user cannot walk.

When W transitions from 0 "cannot walk" to 1 "can walk", a user preference is randomly chosen from a list of pre-defined texts of type "other". The "other" list was previously created by asking ChatGPT to give examples of how a healthy participant feels today.

When W remains at 1 "can walk", we generate the user preference of type "other", based on a Bernoulli distribution: either generate the "other" preference with probability 0.3, or do nothing with probability $1 - 0.3 = 0.7$.

We show some examples of user preferences of type "cannot walk":

```
I am tired, I do not want to walk, I got an injury,
I have a headache, My legs are sore, I twisted my
ankle, I'm feeling dizzy, I'm out of breath, I
have a cold, I'm feeling weak, I pulled a muscle,
My knee hurts, I have blisters, I feel nauseous,
I have stomach cramps, I can't find my shoes, I
don't have time, I'm waiting for someone, It's
too hot outside, It's too cold outside, ...
```

We show some examples of user preferences of type "other":

```
I am feeling good, I'm in a great mood, I feel
energized, I'm feeling positive, I'm doing well
today, I feel great, I'm in high spirits, I feel
focused, I'm feeling relaxed, I feel motivated,
I'm doing fine, I feel optimistic, I'm feeling
calm, I feel balanced, I'm feeling strong, I feel
productive, I'm in a positive state of mind, I
feel healthy, I feel confident, I feel alert, ...
```

B.3.1 Inserting new constraints to impact behavioral dynamics

Below are the equations for the behavioral dynamics implemented in the StepCountJITAI simulation environment, with the new constraints.

We insert the new constraints in blue color. The default base simulator equations are in black color.

The new constraints impact d_{t+1} and s_{t+1}.

We note that $a_t = \tilde{a}$ when the LLM is called, at time t. If the LLM is not called, then a_t takes the RL candidate action value a, at time t.

$$c_{t+1} \sim Bernoulli(0.5), \quad x_{t+1} \sim \mathcal{N}(c_{t+1}, \sigma^2) \tag{10}$$

$$p_{t+1} = P(C = 1|x_{t+1}), \quad l_{t+1} = p_{t+1} > 0.5 \tag{11}$$

$$h_{t+1} = \begin{cases} (1 - \delta_h) \cdot h_t & \text{if } a_t = 0 \\ \min(1, h_t + \epsilon_h) & \text{otherwise} \end{cases} \tag{12}$$

$$d_{t+1} = \begin{cases} d_t & \text{if } a_t = 0 \\ & \text{and } w_t = 0 \text{ or } 1 \\ (1 - \delta_d) \cdot d_t & \text{if } a_t \in \{1, c_t + 2\} \text{ and} \\ & w_t = 1 \text{ (can walk)} \\ \min(1, d_t + \eta_d) & \text{if } a_t \in \{1, c_t + 2\} \text{ and} \\ & w_t = 0 \text{ (cannot walk)} \\ \min(1, d_t + \epsilon_d & \text{otherwise} \\ \quad +(1 - w_t)\, \eta_d) & \end{cases} \tag{13}$$

$$s_{t+1} = \begin{cases} m_s + (1 - h_{t+1}) \cdot \rho_1 & \text{if } a_t = 1 \\ & \text{and } w_t = 1 \text{ (can walk)} \\ m_s + (1 - h_{t+1}) \cdot \rho_2 & \text{if } a_t = c_t + 2 \\ & \text{and } w_t = 1 \text{ (can walk)} \\ m_s\, w_t & \text{otherwise} \end{cases} \tag{14}$$

Below we explain in more detail how the new constraints impact d_{t+1} and s_{t+1}.

- **No message is sent**. If $a_t = 0$, and $w_t = 0$ or 1, then $d_{t+1} = d_t$. When no message is sent to the participant, then it does not matter if the participant can or cannot walk, and the disengagement risk remains the same.
- **Correct message, and can walk**. If $a_t \in \{1, c_t + 2\}$, and $w_t = 1$ (can walk), then $d_{t+1} = (1 - \delta_d)d_t$: we decrement d_t.
- **Correct message, but cannot walk**. If $a_t \in \{1, c_t + 2\}$, and $w_t = 0$ (cannot walk), then $d_{t+1} = \min(1, d_t + \eta_d)$: we increase the disengagement risk. In other words, we insert a single penalty.
- **Incorrect message**. If $a_t > 1$ and $a_t \neq (c_t + 2)$, then $d_{t+1} = \min(1, d_t + \epsilon_d + (1 - w_t)\, \eta_d)$: we have a single penalty (incorrect message, and can walk) or a double penalty (incorrect message, and cannot walk).
- **Cannot walk**. If $w_t = 0$ (cannot walk) then we set $s_{t+1} = 0$.

The reward is the walking step count. It is set to 0 when the user cannot walk. We use the same default parameter values as in the base simulator. We introduce the new constraint parameter for "cannot walk" $\eta_d \in [0, 1]$.

C Experiment details

We provide more details about the experiments described in Section 4.

C.1 Validating LLM response.

We perform experiments to check if the LLM response is correct. For each experiment, we generate prompts for "cannot walk". Ideally, the LLM response to these prompts should be "not send" a message. We also ask the LLM to provide a reason for its decision. Below, we provide examples of user preference, LLM decision, and the LLM reason for its decision.

Examples of user preference, LLM decision, and LLM reason.

Example 1:
The user preference is: `my leg is sore.`

The LLM correctly decides "not send" with the reason: `The user has a sore leg, indicating they cannot walk. The app should not send a message.`

The LLM manages to provide a valid reason.

Example 2:
The user preference is: `I'm feeling dizzy.`

The LLM correctly decides "not send" with the reason: `The user is feeling dizzy, which means they cannot walk, so the mobile health app should not send a message.`

Again, the LLM manages to provide a valid reason.

Example 3:
The user preference is `I am dealing with anxiety.`

The LLM incorrectly decides "send", with the reason: `The user is expressing anxiety, and a tailored message could be helpful.`

In a few cases, the LLM incorrectly decides to send a message, when the user cannot walk. To check the LLM accuracy, we perform preliminary experiments on a dataset of 100 different text-based user preferences (50 "cannot walk" and 50 "other" user preferences), and compare the LLM decisions with the true human decisions. We use an LLM temperature of 0.2 (level of randomness in the LLM response). We use different LLMs, including Gemma 2, Llama 3, Llama 3.1, Llama 3.2, etc. (Gemma Team, 2024; Llama Team, 2024). We found the average accuracies are 0.86 for Gemma 2, 0.87 for Llama 3 8B and 0.98 for Llama 3 70B.

Further investigation reveals that the LLM incorrect decision occurs when the text-based user preference is ambiguous, thus does not clearly indicate if the user can or cannot walk. However, since these ambiguous text-based user preferences appear in less than 6% of the time steps during our experiment, and since the hybrid action falls back to the RL candidate action, LLM+TS still outperforms the standard TS agent.

Above, we have shown how to check if the LLM response is correct, thanks to our simulation environment, by tracking exactly where the LLM decision is incorrect. Future work would involve inserting additional insights into the LLM prompt to further improve the LLM response.

C.2 Validating LLM+TS.

We conduct extensive experiments to compare our novel method LLM+TS to the standard TS. An experiment (a.k.a., trial) corresponds to the behavioral study of one participant, where the maximum study length is 50 days, with daily data. We repeat each experiment 5 times.

We run our experiments for various combinations of the parameters $(p_{w_{11}}, p_{w_{00}})$, where $p_{w_{00}} = 1 - p_{w_{01}}$, to cover different scenarios. For example, the participant often sustains a light injury and thus often cannot walk for short periods, or the participant sometimes twists their ankle and thus sometimes cannot walk for longer periods.

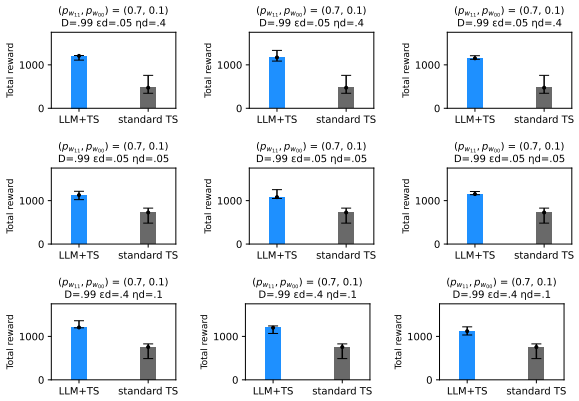
For our experiments, we set the TS prior parameters $\mu_{0a} = 0$ and $\Sigma_{0a} = 100I$ for each action a, and the reward noise variance $\sigma^2_{Ya} = 25^2$ for each action a, using the same notation as in Equations 6 and 7.

For each experiment setting, we compute the total reward as the sum of the rewards over a behavioral study (i.e., up to 50 time steps). We perform the experiments for various combinations of the disengagement parameter ϵ_d from the base simulator, and the new constraint parameter η_d.

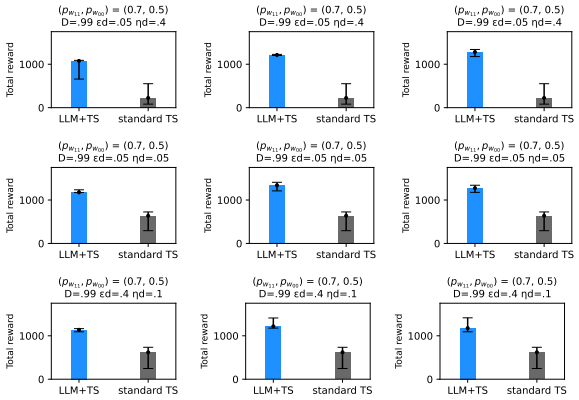
We present the results for two realistic scenarios: Scenario 1, where $p_{w_{11}} = 0.7$, and Scenario 2, where $p_{w_{11}} = 0.95$. In both scenarios, $p_{w_{00}}$ varies in the range $[0.1, ..., 0.5]$. We also set the probability of generating the "other" preference to 0.3. Recall that $p_{w_{00}}$ is the probability of remaining in the "cannot walk" state, and $p_{w_{11}}$ is the probability of remaining in the "can walk" state.

For each experiment, we also run using various LLMs, including Gemma 2, Llama 3, Llama 3.1, Llama 3.2, etc. (Gemma Team, 2024; Llama Team, 2024). When using the different LLM versions, we found similar results for the same experiment settings, as shown in Figure 5.

We run the experiments for various combinations of $(p_{w_{11}}, p_{w_{00}})$. We show the results using Llama 3 8B in Figure 6. The histograms show that LLM+TS is able to capture a larger number of actions 0, which indicates that the LLM has correctly decided to not send a message when the user cannot walk. We also compare the cumulative rewards, and show that LLM+TS outperforms standard TS.

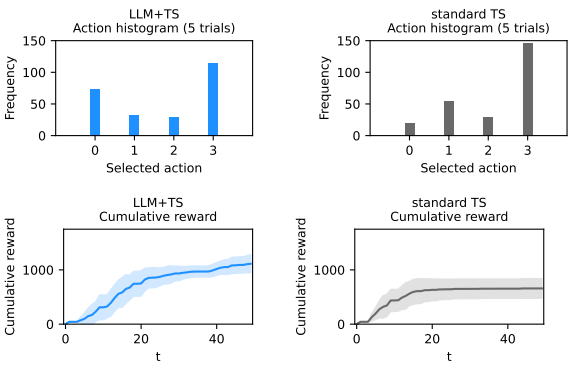


(a) Each row shows a different setting with $(p_{w_{11}}, p_{w_{00}}) = (0.7, 0.1)$.

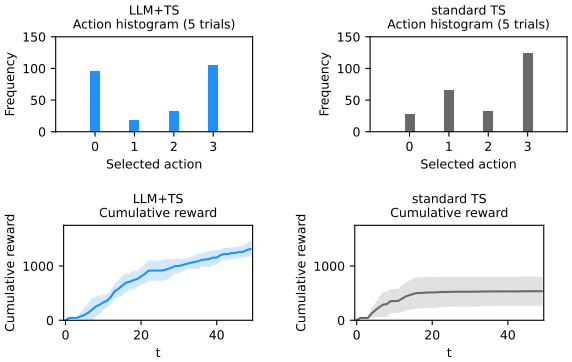


(b) Each row shows a different setting with $(p_{w_{11}}, p_{w_{00}}) = (0.7, 0.5)$.

Figure 5: Comparing LLMs: Gemma 2 9B in the left column, Llama 3 8B in the center column, and Llama 3 70B in the right column. Each row shows a different experiment setting. The results are similar for the same experiment settings.



(a) $(p_{w_{11}}, p_{w_{00}}) = (0.7, 0.1)$



(b) $(p_{w_{11}}, p_{w_{00}}) = (0.7, 0.5)$.

Figure 6: LLM+TS vs. standard TS. Example of histogram for all the selected actions, and plot of the cumulative rewards for various combinations of $(p_{w_{11}}, p_{w_{00}})$. The histograms show that LLM+TS is able to capture a larger number of actions 0, which indicates that the LLM has correctly decided to not send a message when the user cannot walk. The cumulative reward plots show that LLM+TS outperforms standard TS.