StepCountJITAI: simulation environment for RL with application to physical activity adaptive intervention

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

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

The use of reinforcement learning (RL) to learn policies for just-in-time adaptive interventions (JITAIs) is of significant interest in many behavioral intervention domains including improving levels of physical activity. In a messaging-based physical activity JITAI, a mobile health app is typically used to send messages to a participant to encourage engagement in physical activity. In this setting, RL methods can be used to learn what intervention options to provide to a participant in different contexts. However, deploying RL methods in real physical activity adaptive interventions comes with challenges: the cost and time constraints of real intervention studies result in limited data to learn adaptive intervention policies. Further, commonly used RL simulation environments have dynamics that are of limited relevance to physical activity adaptive interventions and thus shed little light on what RL methods may be optimal for this challenging application domain. In this paper, we introduce StepCountJITAI, an RL environment designed to foster research on RL methods that address the significant challenges of policy learning for adaptive behavioral interventions.

1 Introduction

Reinforcement learning (RL) is increasingly being considered for the development of just-in-time adaptive interventions (JITAIs) that aim to increase physical activity [Coronato et al., 2020, Yu et al., 2021, Gönül et al., 2021, Liao et al., 2022]. In a physical activity adaptive intervention, participants typically use a wearable device (e.g., Fitbit) to log aspects of physical activity such as step counts [Nahum-Shani et al., 2018]. In an adaptive messaging-based intervention, a mobile health app is used to send messages to each participant to encourage increased physical activity. In an adaptive intervention, the selection of which messages to send at what times is personalized using context (or tailoring) variables. Context variables can include external factors such as time of day and location, as well as behavioral variables such as whether the participant is experiencing significant stress. Some context variables can be inferred from wearable sensor or other real time data, while others may be provided by participants via self-report mechanisms.

In this setting, RL methods can be used to learn what intervention options to provide to a participant in different contexts with the goal of maximizing a measure of cumulative physical activity, such as total step count over the intervention duration. The state variables used by an RL method correspond to the context variables (observed, inferred, or self-reported) relevant to selecting intervention options. The immediate reward is typically taken to be the step count in a window of time following an intervention decision point.

However, deploying RL methods in real physical activity adaptive interventions comes with challenges: the cost and time constraints of real intervention studies result in limited data to learn adaptive intervention policies. Real behavioral studies are difficult to conduct because they involve following many participants and can run for weeks or months while only allowing a handful of interactions with the participant per day [Hardeman et al., 2019]. This problem is particularly acute given the need to personalize intervention policies to individual participants.

The general problem of data scarcity in real adaptive intervention trials means that RL methods that require a large number of episodes to achieve high performance [Sutton and Barto, 1998, Mnih et al., 2013, Coronato et al., 2020] cannot typically be used in real adaptive intervention studies. Thus, there

NeurIPS 2024 Workshop on Behavioral Machine Learning.

is a need to create new simulation environments that reflect the specific challenges of the adaptive intervention domain to support the exploration of RL methods that are better tailored to meet these challenges. Indeed, commonly used benchmark simulation environments have dynamics that are not particularly relevant to the adaptive intervention domain where there can be significant variation in dynamics within individuals over time as well as between individuals.

We leverage insights from the behavioral domain to construct the proposed simulation environment. In real behavioral studies, there can be missingness and uncertainty among the context variables. For example, participants may not supply requested self-reports or use study devices as expected. Modeling the context uncertainty can provide useful information for decision-making: if the context uncertainty is too high, then a better RL policy might be to send a non-contextualized message, instead of sending an incorrectly contextualized messages that may cause a participant to lose trust in the intervention system or attend less to messages in the future.

The proposed simulation environment reflects these considerations via two primary behavioral variables: habituation level, which measures how much the participant becomes accustomed to receiving messages, and disengagement risk, which measures how likely the participant is to abandon the study. The simulation dynamics relate message contextualization accuracy and habituation level to immediate reward in terms of step count while excess disengagement risk results in early termination of the simulation. The simulation environment also includes stochasticity to represent between- and within-participant variability. In this work, we extend the simulation environment introduced in [Karine et al., 2023] to create a stochastic version of this simulator and introduce new parameters to control the level of stochasticity and between person variability.

Our contributions are:

1. We introduce StepCountJITAI, a messaging-based physical activity JITAI simulation environment that models stochastic behavioral dynamics and context uncertainty, with parameters to control stochasticity. StepCountJITAI can help to accelerate research on RL algorithms for data scarce adaptive intervention optimization by offering a challenging new simulation environment based on key aspects of behavioral dynamics.
2. We provide an open source implementation of StepCountJITAI using a standard API for RL (i.e., gymnasium) to maximize compatibility with existing RL research workflows. We provide quickstart code samples in Appendix E.2 and detail the StepCountJITAI interface in Appendix E.1. StepCountJITAI is available here: `https://github.com/reml-lab/StepCountJITAI`.

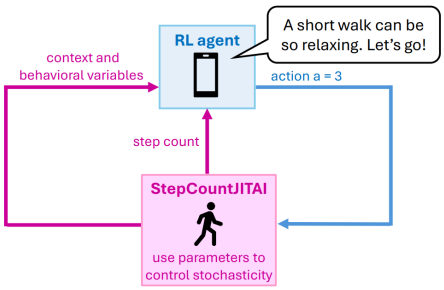
2 Methods

We provide an overview of the use of StepCountJITAI in an RL loop in Figure 1, and provide details below. We first describe the specifications, then introduce our new method. We use the same specifications and deterministic dynamics as in the base simulator [Karine et al., 2023]. For the notation, we use the following: short variable names are in upper case, and variable values are in lower case with subscript t indicating the time index, for example: we use C for the 'true context' variable name, and c_t for the 'true context' value at time t.

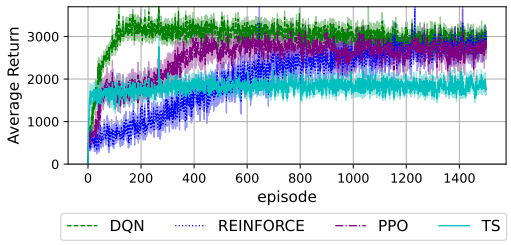
2.1 StepCountJITAI states variables

We describe the state variables below. We show some examples of traces (i.e., how the variables change over time) in Appendix Figure 10. We also provide the deterministic dynamics equations for generating the environment states in Appendix C.

- **True Context (C).** We include an abstract binary context $c_t \in \{0, 1\}$. This context can represent a binary state such as 'stressed'/'not stressed', 'fatigued'/'not fatigued', or 'home'/'not home'.
- **Context probability (P).** This variable represents an inferred probability that the true context takes value 1, where $p_t \in [0, 1]$. It models the fact that in real-world studies, we typically do not have access to the true context, but can make inferences about the context using probabilistic machine learning models.
- **Most likely context (L).** The most likely context $l_t \in \{0, 1\}$ is defined as the context value with the highest inferred probability according to p_t. It can be used to model situations where the context uncertainty is discarded when learning intervention policies.



(a) Overview of StepCountJITAI in an RL loop.



(b) Example of average returns using StepCountJITAI with RL methods: DQN, REINFORCE, PPO and TS.

Figure 1: Overview of StepCountJITAI in an RL loop. StepCountJITAI is a simulation environment for physical activity adaptive interventions. StepCountJITAI models stochastic behavioral dynamics and context uncertainty, with parameters to control stochasticity. Step count is used as the reward. The actions correspond to physical activity motivational messages with different contextualization levels. The messages can be non-contextualized, or customized to a binary context. The behavioral variables are: habitation and disengagement risk.

- **Habituation level (H).** As the participant receives more messages, the participant becomes more accustomed to the messages, thus the habituation level h_t will increase, with $h_t \in [0, 1]$. An increase in h_t also reduces the step count s_t because the messages become less effective.
- **Disengagement risk (D).** If the participant keeps receiving messages that are not useful (e.g., the context for the customized messages do not match the true context), then the disengagement risk d_t will increase, with $d_t \in [0, 1]$. If d_t exceeds a preset threshold $D_{threshold}$ the episode ends and all future rewards are 0.
- **Step Count (S).** The participant's step count is also the observed reward in the RL system.

When performing simulations, the observed state variables accessible to an RL method can be selected from the list of state variables maintained by the simulator depending on the desired experiment design, for example: $[C, H, D]$, $[L, H, D]$, or $[H, D]$.

2.2 StepCountJITAI dynamics

The environment dynamics depend on four actions: $a = 0$ indicates no message is sent. $a = 1$ indicates a non-contextualized message is sent. $a = 2$ indicates a message customized to context 0 is sent. $a = 3$ indicates a message customized to context 1 is sent to the participant.

The environment dynamics can be summarized as follows: 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 the given threshold, the episode ends. The reward is the surplus step count, beyond a baseline count, attenuated by the habituation level.

The base simulator implements deterministic dynamics, which we summarize in Appendix C. In this work, we extend the base simulator to create a simulation environment with additional stochasticity by introducing noise into the existing deterministic dynamics. We let h_t be the habituation level, d_t be the disengagement risk level, s_t be the step count at time t. The dynamics of habituation and disengagement are governed by increment and decay parameters including the habituation decay δ_h, the habituation increment ϵ_h, the disengagement risk decay δ_d, and the disengagement risk increment ϵ_d. In the base simulator the dynamics parameters δ_h, ϵ_h, δ_d, and ϵ_d are fixed. We make them stochastic at the episode level to model between person variation in the dynamics of habituation and disengagement risk. We also make the state variables h_t, d_t and s_t stochastic.

We construct two different versions of the stochastic dynamics based on the uniform and beta distributions. The uniform uncertainty-based dynamics are summarized below where the a parameters control the width of a uniform distribution about the mean values. The step counts themselves are positive reals and are sampled from a Gamma distribution parameterized by its mean and standard deviation σ_s. The alternative beta distribution-based stochastic dynamics sample the values in $[0, 1]$

from beta distributions with the same means as noted below, but with the spread specified via a concentration parameter, as shown in Appendix D.

In the equations below, the new equations for the uniform uncertainty-based dynamics are shown in blue. The deterministic dynamics are shown in black. The base dynamic parameters and output of the deterministic dynamics are indicated using a symbol $\hat{}$, for example: $\hat{\delta_h}$, $\hat{\epsilon_h}$, $\hat{\delta_d}$, $\hat{\epsilon_d}$, $\hat{h}_{t+1}$ and $\hat{d}_{t+1}$.

$$\delta_h \sim Uniform\big((1-\frac{a_{de}}{2})\hat{\delta_h}, (1+\frac{a_{de}}{2})\hat{\delta_h}\big) \qquad \delta_d \sim Uniform\big((1-\frac{a_{de}}{2})\hat{\delta_d}, (1+\frac{a_{de}}{2})\hat{\delta_d}\big)$$

$$\epsilon_h \sim Uniform\big((1-\frac{a_{de}}{2})\hat{\epsilon_h}, (1+\frac{a_{de}}{2})\hat{\epsilon_h}\big) \qquad \epsilon_d \sim Uniform\big((1-\frac{a_{de}}{2})\hat{\epsilon_d}, (1+\frac{a_{de}}{2})\hat{\epsilon_d}\big)$$

$$c_{t+1} \sim Bernoulli(0.5), \quad x_{t+1} \sim \mathcal{N}(c_{t+1}, \sigma^2) \qquad p_{t+1} = P(C=1|x_{t+1}), \quad l_{t+1} = p_{t+1} > 0.5$$

$$\hat{h}_{t+1} = \begin{cases} (1-\delta_h)\cdot h_t & \text{if } a_t = 0 \\ \min(1, h_t + \epsilon_h) & \text{otherwise} \end{cases} \qquad \hat{d}_{t+1} = \begin{cases} d_t & \text{if } a_t = 0 \\ (1-\delta_d)\cdot d_t & \text{if } a_t \in \{1, c_t+2\} \\ \min(1, d_t + \epsilon_d) & \text{otherwise} \end{cases}$$

$$h_{t+1} \sim Uniform\big((1-\frac{a_{hd}}{2})\hat{h}_{t+1}, (1+\frac{a_{hd}}{2})\hat{h}_{t+1}\big) \qquad d_{t+1} \sim Uniform\big((1-\frac{a_{hd}}{2})\hat{d}_{t+1}, (1+\frac{a_{hd}}{2})\hat{d}_{t+1}\big)$$

$$\hat{s}_{t+1} = \begin{cases} m_s + (1-h_{t+1})\cdot\rho_1 & \text{if } a_t = 1 \\ m_s + (1-h_{t+1})\cdot\rho_2 & \text{if } a_t = c_t + 2 \\ m_s & \text{otherwise} \end{cases} \qquad s_{t+1} \sim Gamma\big((\frac{\hat{s}_{t+1}}{\sigma_s})^2, \frac{\sigma_s^2}{\hat{s}_{t+1}}\big)$$

where c_t is the true context, x_t is the context feature, σ is the context uncertainty, 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 number of walking steps), and a_t is the action value at time t. $\sigma, \rho_1, \rho_2, m_s$ are fixed parameters. The default parameters are provided in Appendix C. The spreads of the distributions are controlled by the parameters a_{de}, a_{hd} and σ_s.

We describe where the environmental dynamics occur in a typical RL loop: at each time t, the agent observes the current state (e.g., $[c_t, h_t, d_t]$), and selects an action a_t based on the observed state variables. Then, the environment responds by transitioning to a new state (e.g., $[c_{t+1}, h_{t+1}, d_{t+1}]$), and providing a reward (e.g., the participant step count s_{t+1}).

3 Experiments

We perform RL experiments using StepCountJITAI including learning action selection policies with various RL methods: REINFORCE and PPO as examples of policy gradient methods, and DQN as an example of a value function method [Williams, 1987, Schulman et al., 2017, Mnih et al., 2013]. We also consider a standard Thompson sampling (TS) [Thompson, 1933]. We provide the RL implementation settings in Appendix F.4, and code samples in Appendix E.2.3. In Figure 1 (b), we show the mean and standard deviation of the average return over 10 trials, with 1500 episodes per trial, when using StepCountJITAI, with observed data $[C, H, D]$, and using the stochastic parameters for Uniform distributions: $a_{hd} = 0.2$, $a_{de} = 0.5$, $\sigma_s = 20$, and context uncertainty $\sigma = 2$. In this setting, we show that the RL and TS agents are able to learn, with a maximum average return of around 3000 for RL and 1500 for TS. As expected, TS shows a lower average return than RL when using a complex environment such as StepCountJITAI.

We show additional results including generating traces in Appendix F.3 and stochastic variables histograms in Appendix F.2. We perform additional RL experiments using various parameter settings to control stochasticity in Appendix F.5.

4 Conclusion

We introduce StepCountJITAI, a simulation environment for physical activity adaptive interventions. StepCountJITAI is implemented using a standard RL API to maximize compatibility with existing RL research workflows. StepCountJITAI models key aspects of behavioral dynamics including habituation and disengagement risk, as well as context uncertainty and between person variability in dynamics. We hope that StepCountJITAI will help to accelerate research on new RL algorithms for the challenging problem of data scarce adaptive intervention optimization.

Acknowledgements

This work was 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 and 1P41EB028242.

References

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

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

Wendy Hardeman, Julie Houghton, Kathleen Lane, Andy Jones, and Felix Naughton. A systematic review of just-in-time adaptive interventions (jitais) to promote physical activity. *International Journal of Behavioral Nutrition and Physical Activity*, 16(1):1–21, 2019.

Karine Karine, Predrag Klasnja, Susan A. Murphy, and Benjamin M. Marlin. 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, 2023.

Peng Liao, Zhengling Qi, Runzhe Wan, Predrag Klasnja, and Susan A. Murphy. Batch policy learning in average reward Markov decision processes. *The Annals of Statistics*, 50(6):3364 – 3387, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NeurIPS Deep Learning Workshop*, 2013.

Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462, 2018.

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

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT press, Cambridge, MA, 1998.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. In *Biometrika*, volume 25, pages 285–294, 1933.

Ronald J. Williams. A class of gradient-estimation algorithms for reinforcement learning in neural networks. In *Proceedings of the International Conference on Neural Networks*, pages II–601, 1987.

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

A Appendix

Below we provide the table of content for the main paper, as well as for the appendix.

Contents

B Overview of StepCountJITAI

We provide the summaries of the actions, environment states, and parameters for StepCountJITAI.

In Appendix B.1, B.2, B.3, we provide a summary of the same specifications as the base simulation environment introduced in [Karine et al., 2023].

In Appendix B.4, we provide a summary of the new stochastic parameters that we introduce in this work. We describe the stochastic dynamics in the main paper in Section 2.2.

B.1 Summary of the possible action values used by StepCountJITAI

Table 1: Possible action values

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

B.2 Summary of the variables generated by StepCountJITAI

Table 2: StepCountJITAI simulation variables

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.	$\mathbb{R}^+$

B.3 Summary of StepCountJITAI parameters for deterministic dynamics

Table 3: StepCountJITAI parameters for deterministic dynamics

Parameter	Description
σ	Context uncertainty. The default value is $\sigma = 0.4$
δ_d	Disengagement risk decay. The default value is $\delta_d = 0.1$.
δ_h	Habituation decay. The default value is $\delta_h = 0.1$.
ϵ_d	Disengagement risk increment. The default value is $\epsilon_d = 0.4$.
ϵ_h	Habituation increment. The default value is $\epsilon_h = 0.05$.
ρ_1	$a_t = 1$ base step count. The default value is $\rho_1 = 50$.
ρ_2	$a_t = c_t + 2$ base step count. The default value $\rho_2 = 200$.
m_s	Base step count. The default value is $m_s = 0.1$.

B.4 Summary of StepCountJITAI parameters for stochastic dynamics

In this section, we provide a summary of the newly introduced parameters for the stochastic dynamics, as described in Section 2.2 and Appendix D.

Table 4: StepCountJITAI parameters for stochastic dynamics

Parameter	Description
σ_s	Parameter to control the spread of the Gamma distribution for s_t.
a_{hd}	Parameter to control the spread of the Uniform distributions for h_t and d_t.
a_{de}	Parameter to control the spread of the Uniform distributions for $\delta_d, \epsilon_d, \delta_h$ and ϵ_h.
κ_d	Parameter to control the spread of the Beta distribution for d_t.
κ_h	Parameter to control the spread of the Beta distribution for h_t.
κ_{δ_d}	Parameter to control the spread of the Beta distribution for δ_d.
κ_{δ_h}	Parameter to control the spread of the Beta distribution for δ_h.
κ_{ϵ_d}	Parameter to control the spread of the Beta distribution for ϵ_d.
κ_{ϵ_h}	Parameter to control the spread of the Beta distribution for ϵ_h.

C StepCountJITAI deterministic dynamics

The simulation environment introduced in the base simulator [Karine et al., 2023] models the deterministic dynamics. We summarize the specifications in Tables 1 and 2.

We provide a summary of the **deterministic dynamics** below.

$$
\begin{aligned}
c_{t+1} &\sim Bernoulli(0.5) \\
x_{t+1} &\sim \mathcal{N}(c_{t+1}, \sigma^2) \\
p_{t+1} &= P(C = 1 | x_{t+1}) \\
l_{t+1} &= p_{t+1} > 0.5 \\
h_{t+1} &= \begin{cases} (1 - \delta_h) \cdot h_t & \text{if } a_t = 0 \\ \min(1, h_t + \epsilon_h) & \text{otherwise} \end{cases} \\
d_{t+1} &= \begin{cases} d_t & \text{if } a_t = 0 \\ (1 - \delta_d) \cdot d_t & \text{if } a_t = 1 \text{ or } a_t = c_t + 2 \\ \min(1, d_t + \epsilon_d) & \text{otherwise} \end{cases} \\
s_{t+1} &= \begin{cases} m_s + (1 - h_{t+1}) \cdot \rho_1 & \text{if } a_t = 1 \\ m_s + (1 - h_{t+1}) \cdot \rho_2 & \text{if } a_t = c_t + 2 \\ m_s & \text{otherwise} \end{cases}
\end{aligned}
$$

where c_t is the true context, x_t is the context feature, σ is the context uncertainty, 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 number of walking steps), a_t is the action value at time t.

The behavioral dynamics can be tuned using the parameters: disengagement risk decay δ_d, disengagement risk increment ϵ_d, habituation decay δ_h, and habituation increment ϵ_h.

The default parameters values for the base simulator are: $\sigma = 0.4$, $\delta_h = 0.1$, $\epsilon_h = 0.05$, $\delta_d = 0.1$, $\epsilon_d = 0.4$, $\rho_1 = 50$, $\rho_2 = 200$, $m_s = 0.1$, disengagement threshold $D_{threshold} = 0.99$ (the study ends if d_t exceeds $D_{threshold}$). The maximum study length is 50 days with one intervention per day, thus the maximum episode length is 50 days.

The context uncertainty σ is typically set by the user, with value $\sigma \in [0.2, 10.]$. In Appendix F.1, we describe how to select σ.

D StepCountJITAI beta distribution-based stochastic dynamics

In the main paper, in Section 2.2, we introduce the equations for the uniform uncertainty-based dynamics. Below we introduce the beta distribution-based stochastic dynamics, using the same notations. The spread of the distribution is controlled by the κ concentration parameter.

$$
\begin{aligned}
h_{t+1} &\sim Beta\big(\kappa_h \hat{h}_{t+1}, \kappa_h (1 - \hat{h}_{t+1})\big) \\
d_{t+1} &\sim Beta\big(\kappa_d \hat{d}_{t+1}, \kappa_d (1 - \hat{d}_{t+1})\big)
\end{aligned}
\qquad
\begin{aligned}
\delta_d &\sim Beta\big(k_{\delta_d} \hat{\delta_d}, k_{\delta_d} (1 - \hat{\delta_d})\big) \\
\epsilon_d &\sim Beta\big(k_{\epsilon_d} \hat{\epsilon_d}, k_{\epsilon_d} (1 - \hat{\epsilon_d})\big) \\
\delta_h &\sim Beta\big(k_{\delta_h} \hat{\delta_h}, k_{\delta_h} (1 - \hat{\delta_h})\big) \\
\epsilon_h &\sim Beta\big(k_{\epsilon_h} \hat{\epsilon_h}, k_{\epsilon_h} (1 - \hat{\epsilon_h})\big).
\end{aligned}
$$

E How to code using StepCountJITAI

E.1 StepCountJITAI interface

We implement the StepCountJITAI interface using a standard API for RL (i.e., gymnasium), so that StepCountJITAI can simply be plugged into a typical RL loop. The description of the API functions `reset()` and `step(action)`, and the output variables `info`, `terminated`, and `truncated`, can be found in the gymnasium.Env online documentation.

When instantiating using `env = StepCountJITAI(chosen_obs_names = ...)` as shown in the code sample in Appendix E.2.1, we can specify the desired variable names in `chosen_obs_names`. For example:

`chosen_obs_names = ['C', 'H']` will generate observed data $[c_t, h_t]$ at each time t.
`chosen_obs_names = ['C', 'H', 'D']` will generate observed data $[c_t, h_t, d_t]$ at each time t.

We can also specify the parameters listed in Appendix B, by inserting the parameters as arguments in `StepCountJITAI(...)`, as shown in the code samples in Appendix E.2.

We can use a get function to extract the current variable value at time t, for example: `get_C()` will extract the current true context value at time t.

We provide a summary of the main functions for StepCountJITAI in Table 5.

Table 5: StepCountJITAI main functions

Function	Output
`reset()`	reset observation and info.
`step(action)`	next observation, reward, terminated, truncated, and info.
`get_C()`	current true context value.
`get_H()`	current habituation level value.
`get_D()`	current disengagement risk value.
`get_L()`	current inferred context value.
`get_P()`	current probability of context 1 value.
`get_S()`	current step count value.

E.2 Quickstart code samples for StepCountJITAI

Below we provide quickstart code samples for StepCountJITAI. The StepCountJITAI interface is detailed in Appendix E.1.

E.2.1 Creating a StepCountJITAI simulation environment

StepCountJITAI is available here: `https://github.com/reml-lab/StepCountJITAI`.

To create the StepCountJITAI environment, we can call `StepCountJITAI(...)`. We can set the parameters (e.g., `n_version=1` for the stochastic version using Uniform distributions), and choose the observed variable names, as shown below.

```
env = StepCountJITAI(sigma=0.4, chosen_obs_names=['C', 'H'], n_version=1)
```

E.2.2 Generating simulation variables with random actions

We can generate the simulation variables with random actions, and use the built-in get functions to extract a particular variable. Below is an example where we store c_t and h_t current values into arrays.

```
Cs=[]; Hs=[]
for t in range(50):
  Cs.append(env.get_C())
  Hs.append(env.get_H())
  action = np.random.choice(4)
  obs_, reward, terminated, truncated, info = env.step(action)
```

E.2.3 Using StepCountJITAI in an RL loop

Below is the code for a typical RL loop, where the RL `agent` selects an action at each time t.

```
obs, info = env.reset()
for t in range(50):
  action = agent.select_action(obs)
  obs_, reward, terminated, truncated, info = env.step(action)
  obs = obs_
  if terminated or truncated: break
```

F Additional Experiments and Results

F.1 Experiment: How to select the context uncertainty σ?

The context uncertainty σ is a parameter that was introduced in the base simulator [Karine et al., 2023]. The user can use σ to control the desired context error.

We perform some experiments to show the relationship between the context uncertainty σ and the context error. In our experiment, we generate the true context c_t and the inferred context l_t, using the deterministic dynamics equations in Section C, for various fixed values of σ. Then we compute the context error (percentage of true context values that match the inferred context values), for $N = 5000$.

We note that as the context uncertainty σ increases, the context error increases.

We create a lookup table in Table 6, that can be used as a reference for selecting σ. For example:

- The user can set σ to 0.2, to get a context error of 1%.
- The user can set σ to 0.4, to get a context error of 10%.
- The user can set σ to 1, to get a context error of 30%.

Table 6: Lookup table: context accuracy and context error, for various values of context uncertainty σ

context uncertainty σ	**context accuracy**	**context error**
0.2	99 %	1 %
0.3	96 %	4 %
0.4	90 %	10 %
0.8	74 %	26 %
1.	70 %	30 %
2.	60 %	40 %
3.	57 %	43 %
10.	52 %	48 %

F.2 Experiments: Creating histograms for stochastic h_t, d_t, s_t and δ_h, ϵ_h, δ_d, ϵ_d

To illustrate the effects of the parameters on the stochastic dynamics, we generate h_t, d_t, s_t, as well as δ_h, ϵ_h, δ_d, ϵ_d using the equations in Section 2.2, for various sets of parameters and fixed deterministic values. We plot the histograms below.

F.2.1 Histograms for stochastic h_t, d_t, s_t

We generate $N = 5000$ samples of h_t, d_t, s_t, using $h_t = 0.5$, $d_t = 0.75$, $s_t = 200$. We plot the histograms for h_t, d_t, s_t below. The vertical lines represent the deterministic values.

Uniform distributions. To sharpen the histogram peaks ("making the stochastic dynamics closer to deterministic"), the a_{hd} and σ_s values can be reduced.

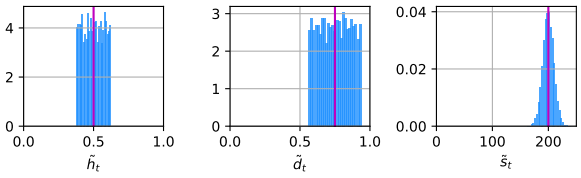


Figure 2: Histograms for stochastic h_t, d_t, s_t using $a_{hd} = 0.5$, $\sigma_s = 10$.

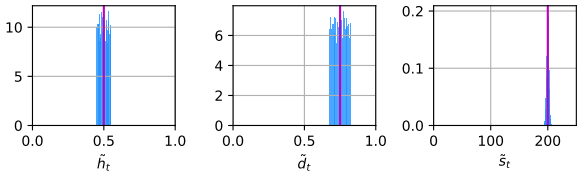


Figure 3: Histograms for stochastic h_t, d_t, s_t using $a_{hd} = 0.2$, $\sigma_s = 2$.

Beta distributions. To sharpen the histogram peaks ("making the stochastic dynamics closer to deterministic"), the κ parameters can be set to larger values, and the σ_s value can be reduced.

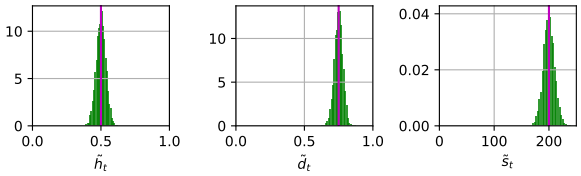


Figure 4: Histograms for stochastic h_t, d_t, s_t using $\kappa_h = 200$, $\kappa_d = 200$, $\sigma_s = 10$.

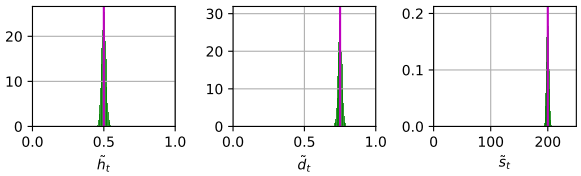


Figure 5: Histograms for stochastic h_t, d_t, s_t using $\kappa_h = 1000$, $\kappa_d = 1000$, $\sigma_s = 2$.

F.2.2 Histograms for stochastic δ_h, ϵ_h, δ_d, ϵ_d

We generate $N = 1000$ samples of δ_h, ϵ_h, δ_d, ϵ_d, using $\delta_h = 0.1, \epsilon_h = 0.05, \delta_d = 0.1, \epsilon_d = 0.4$. We plot the histograms for δ_h, ϵ_h, δ_d, ϵ_d below. The vertical lines represent the deterministic values.

Uniform distributions. To sharpen the histogram peaks ("making the stochastic dynamics closer to deterministic"), the a_{de} values can be reduced.

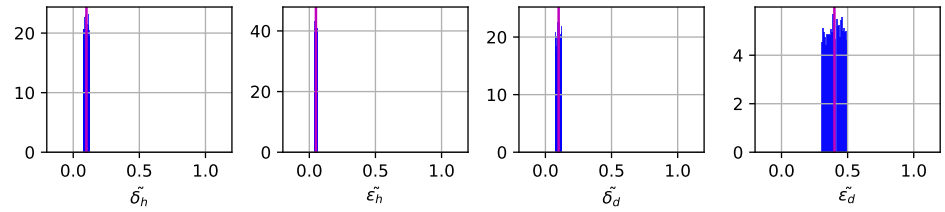


Figure 6: Histograms for δ_h, ϵ_h, δ_d, ϵ_d using $a_{de} = 0.5$.

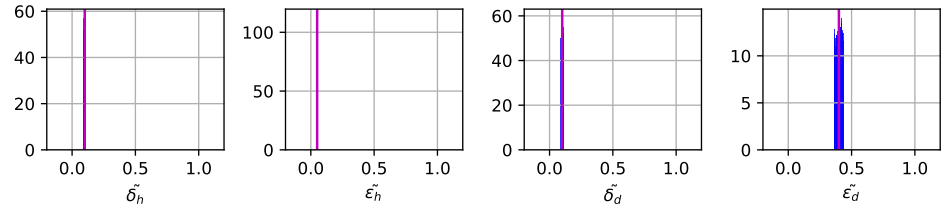


Figure 7: Histograms for δ_h, ϵ_h, δ_d, ϵ_d using $a_{hd} = 0.2$.

Beta distributions. To sharpen the histogram peaks ("making the stochastic dynamics closer to deterministic"), the κ parameters can be set to larger values.

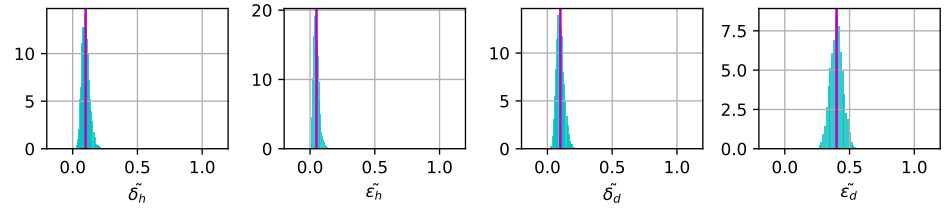


Figure 8: Histograms for δ_h, ϵ_h, δ_d, ϵ_d using $k_{\delta_h} = 100, k_{\epsilon_d} = 100, k_{\delta_d} = 100, k_{\epsilon_d} = 100$.

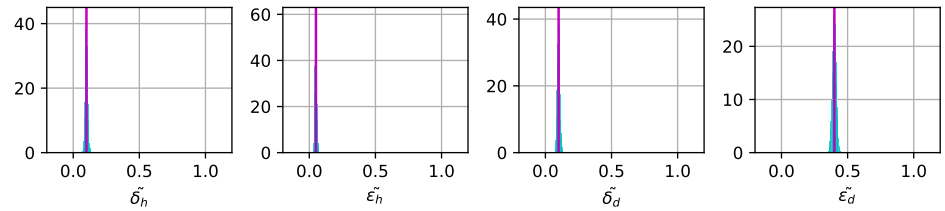


Figure 9: Histograms for δ_h, ϵ_h, δ_d, ϵ_d using $k_{\delta_h} = 1000, k_{\epsilon_h} = 1000, k_{\delta_d} = 1000, k_{\epsilon_d} = 1000$.

F.3 Experiments: Generating traces using StepCountJITAI with fixed actions or random actions

We provide examples of traces using deterministic StepCountJITAI, stochastic StepCountJITAI with Uniform distributions, and stochastic StepCountJITAI with Beta distributions, when using one of the following policies. We implement two policies: policy **"always a $= 3$"** where at each time t, the selected action is fixed, with value $a = 3$, and policy **"random action"** where at each time t, the selected action has a random value $a \in [0, 3]$.

We provide the code sample for policy "random action" in Section E.2.2. The code sample for policy "always $a = 3$" is the same except that the action is fixed to the value 3.

In our experiments, for each version of StepCountJITAI, we generate observed data $[C, P, L, H, D]$ in a loop, using one of the two policies described above, for 30 time steps. We plot the traces of $[C, P, L, H, D]$, the actions and the cumulative rewards at each time step t.

To get the traces for the full 30 steps, we set $D_{threshold} > 1$ (e.g., 1.5), so that $d_t \in [0, 1]$ will never exceeds $D_{threshold}$.

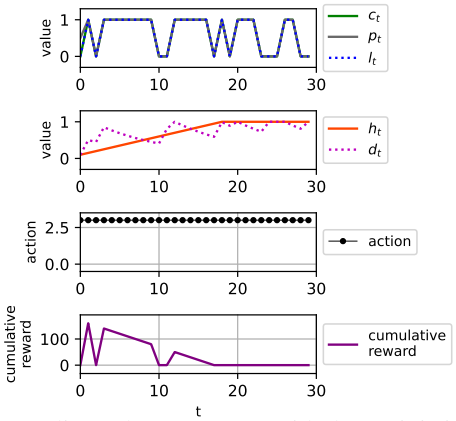
For deterministic StepCountJITAI, we use: context uncertainty $\sigma = 0.01$ (i.e., nearly 0 context error) and the same default parameters as in the base simulator, as described in Appendix C.

For stochastic StepCountJITAI with Uniform distributions, we run experiments for various combinations of parameters to control the stochasticity: [σ, a_{hd}, σ_s, a_{de}] values: $[.1, .05, 2.5, .05]$, $[.8, .05, 2.5, .05]$, $[1., .05, 2.5, .05]$, $[2., .05, 2.5, .05]$, $[.1, .2, 10., .2]$, $[.8, .2, 10., .2]$, $[1., .2, 10., .2]$, $[2., .2, 10., .2]$, $[.1, .2, 20., .5]$, $[.8, .2, 20., .5]$, $[1., .2, 20., .5]$, $[2., .2, 20., .5]$. We show the results for: $\sigma = 2$, $a_{hd} = 0.2$, $\sigma_s = 20$ and $a_{de} = 0.5$.

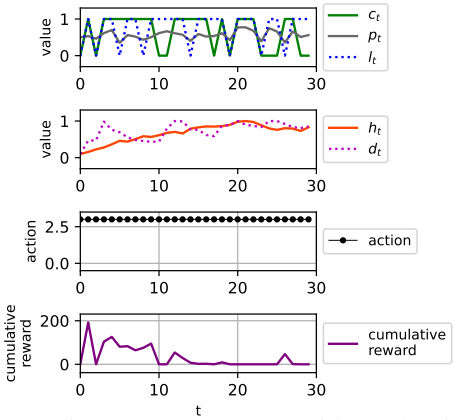
For stochastic StepCountJITAI with Beta distributions, we run experiments with κ values in $\{1, 20, 100\}$, σ_s in $\{2.5, 10, 20\}$ and σ in $\{0.1, 0.8, 2\}$. We show the results for: $\sigma = 2$, all $\kappa = 100$ and $\sigma_s = 20$.

The traces are shown in Figure 10. We can see that when using deterministic StepCountJITAI with nearly 0 context uncertainty, the true context c_t, the probability of context=1 p_t and the inferred context l_t match as expected. When using the stochastic versions of StepCountJITAI, we note that the true context c_t and inferred context l_t do not always overlap, due to the context uncertainty.

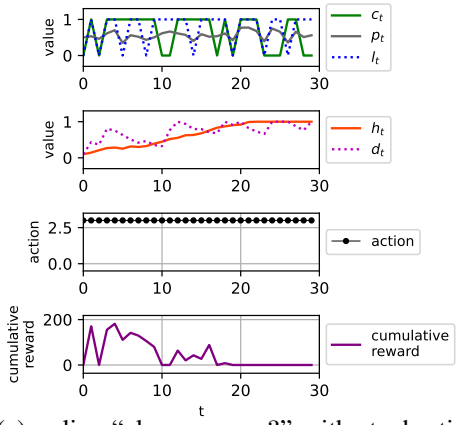
We note that these two policies are ineffective as expected. We can see that the cumulative reward decreases over time as per the environment dynamics. In the main paper, in Section 3, we describe the experiments with the RL methods, which have better policies.



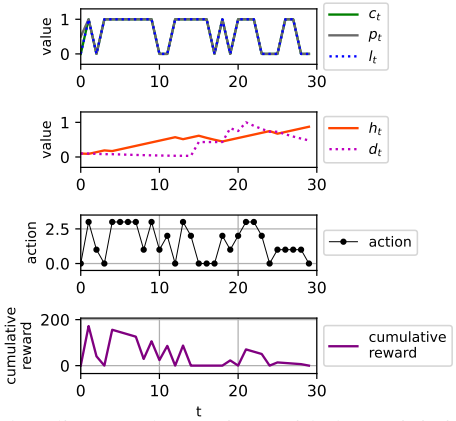
(a) policy "always $a = 3$" with deterministic StepCountJITAI.



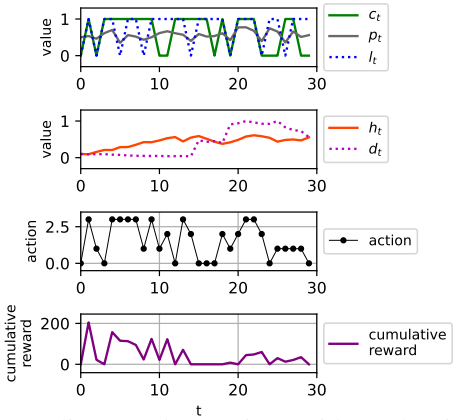
(b) policy "always $a = 3$" with stochastic StepCountJITAI with Uniform distributions.



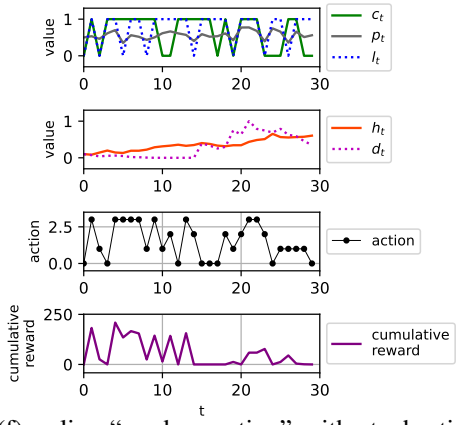
(c) policy "always $a = 3$" with stochastic StepCountJITAI with Beta distributions.



(d) policy "random action" with deterministic StepCountJITAI.



(e) policy "random action" with stochastic StepCountJITAI with Uniform distributions.



(f) policy "random action" with stochastic StepCountJITAI with Beta distributions.

Figure 10: Examples of traces using deterministic StepCountJITAI, and two versions of stochastic StepCountJITAI, with policy: (left) "always $a = 3$" and (right) "random action".

F.4 RL Experiment Details

In Section 3, we describe the experiment and results when using StepCountJITAI with RL. Below we provide the experiment details. For each RL method, we select the best hyperparameters that maximize the performance, with the lowest number of episodes: the average return is around 3000 for the RL methods, and around 1500 for basic TS. All experiments can be run on CPU, using Google Colab within 2GB of RAM.

The RL implementation details are as follows.

REINFORCE. We use a one-layer policy network. We perform a hyperparameter search over hidden layer sizes $[32, 64, 128, 256]$, and Adam optimizer learning rates from 1e-6 to 1e-2. We report the results for 128 neurons, batch size $b = 64$, and Adam optimizer learning rate $lr =$ 6e-4.

DQN. We use a two-layer policy network. We perform a hyperparameter search over hidden layers sizes $[32, 64, 128, 256]$, batch sizes $[16, 32, 64]$, Adam optimizer learning rates from 1e-6 to 1e-2, and epsilon greedy exploration rate decrements from 1e-6 to 1e-3. We report the results for 128 neurons in each hidden layer, batch size $b = 64$, Adam optimizer learning rate $lr =$ 5e-4, epsilon linear decrement $\delta_\epsilon = 0.001$, decaying ϵ from 1 to 0.01. The target Q network parameters are replaced every $K = 1000$ steps.

PPO. We use a two-layer policy network, and a three layers critic network. We perform a hyperparameter search over hidden layers sizes $[32, 64, 128, 256]$, batch sizes $[16, 32, 64]$, Adam optimizer learning rates from 1e-6 to 1e-2, horizons from 10 to 40, policy clips from 0.1 to 0.5, and the other factors from .9 to 1.0. We report the results for 256 neurons in each hidden layer, batch size $b = 64$, Adam optimizer learning rate $lr =$ 5e-3, horizon $H = 20$, policy clip $c = 0.08$, discounted factor $\gamma = 0.99$ and Generalized Advantage Estimator (GAE) factor $\lambda = 0.95$.

TS. We use a standard linear Thompson sampling with prior means $\mu_{0a} = 0$, and covariance matrices $\Sigma_{0a} = 100I$, and we set the model noise variance $\sigma^2_{Ya} = 25^2$ for all action values a.

In Section 3, we use the following StepCountJITAI parameter settings.

StepCountJITAI. We use observed data $[C, H, D]$ and the stochastic version with Uniform distributions, with parameters: $a_{hd} = 0.2$, $a_{de} = 0.5$, $\sigma_s = 20$, and context uncertainty $\sigma = 2$.

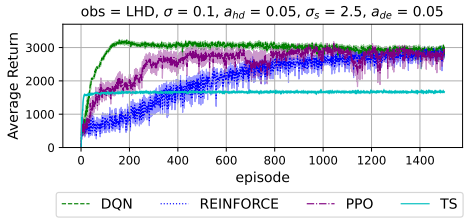
In Appendix F.5, we perform additional experiments with various StepCountJITAI parameter settings.

F.5 Additional RL Results for StepCountJITAI

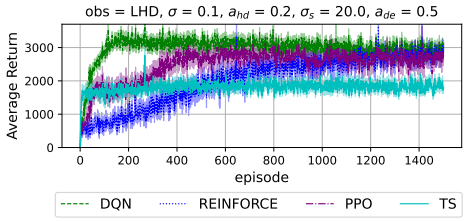
In Section 3, we show an example where the RL methods achieve a high average return of around 3000. Below we perform additional experiments for StepCountJITAI with RL, to show when a standard RL method can or cannot work. We use different observed data, and different sets of parameters to control the stochasticity in the environment dynamics and context uncertainty.

We show an example of a case study where we do not have access to the true context C, but only to the inferred context L, and we have access to the behavioral variables H and D. Thus, we use StepCountJITAI with observed data $[L, H, D]$. We use the version with stochasticity using Uniform distributions, as described in Section 2.2. We use the same RL settings as described in Appendix F.4.

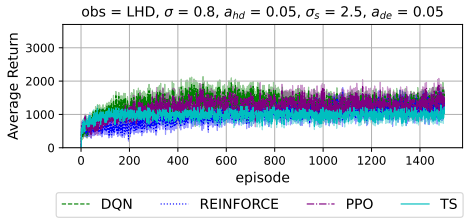
For the StepCountJITAI parameter settings, we use two settings of context uncertainty: lower context uncertainty $\sigma = 0.1$ and higher context uncertainty $\sigma = 0.8$, and two settings of parameters to control the stochasticity in the environment dynamics: lower stochasticity $[a_{hd}, \sigma_s, a_{de}] = [0.05, 2.5, 0.05]$ and higher stochasticity $[a_{hd}, \sigma_s, a_{de}] = [0.2, 20.0, 0.5]$. We show the mean and standard deviation of the average return over 10 trials, with 1500 episodes per trial. We can see that when using the settings for lower context uncertainty and lower stochasticity in the environment dynamics, all the RL methods are able to learn, and reach a high average return of around 3000. When using the settings for lower context uncertainty but with higher stochasticity in the environment dynamics, the variability in the average returns is also higher. Using the setting for higher context uncertainty, all the RL methods average returns drop to below 2000. As expected, TS shows a lower average return than the RL methods in all the experiments.



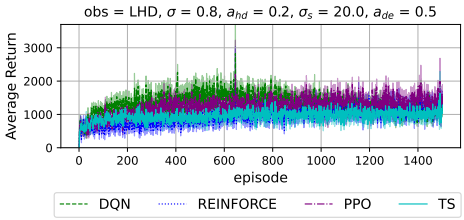
(a) Lower context uncertainty, lower stochasticity in the dynamics.



(b) Lower context uncertainty, higher stochasticity in the dynamics.



(b) Higher context uncertainty, lower stochasticity in the dynamics.



(d) Higher context uncertainty, higher stochasticity in the dynamics.

Figure 11: Examples of StepCountJITAI with RL, using StepCountJITAI with observed data $[L, H, D]$, and various settings of context uncertainty and parameters to control the stochasticity in the environment dynamics.