Personalizing AI Interventions in Multiple Health Behavioral Change Settings

Samantha Marks [1] **Michelle Chang** [2] **Eura Nofshin** [2] **Weiwei Pan** [2] **Finale Doshi-Velez** [2]

Abstract

We introduce a novel reinforcement learning (RL) framework for rapidly personalizing AI interventions in multiple health behavior change (MHBC) settings. Our key contribution is a simple, interpretable model that captures empirically observed human multi-goal pursuit behaviors. Using this model, we provide insight into how the AI will intervene, including when it has varying degrees of knowledge about the human model.

1. Introduction

In mobile health (mHealth) settings where AI assists humans in behavioral changes, fast personalization of AI interventions is needed to avoid user disengagement. Existing works propose reinforcement learning (RL) frameworks for optimizing and personalizing AI interventions in single-goal mHealth settings (e.g. Trella et al., 2024; Liao et al., 2020; Wang et al., 2021; Yom-Tov et al., 2017; Menictas et al., 2019; Zhou et al., 2018; Ghosh et al., 2024), in particular, by modeling humans as Markov Decision Process (MDP) planners (e.g. Nofshin et al., 2024). However, many applications occur in multiple health behavioral change (MHBC) settings, where humans pursue many health goals simultaneously, e.g. by working on diet and exercise to reduce cardiovascular risk (e.g. Sharma et al., 2021). Currently, there is no RL framework for personalizing AI interventions for MHBC. Designing interventions for multiple-goal settings is more challenging, because human behaviors here are more complex – humans must prioritize goals and adaptively switch between them (e.g. Scholer et al., 2024).

In this work, we build on an interpretable RL framework for single-goal settings (Nofshin et al., 2024), where humans are modeled as MDP planners and AI interventions modify parameters of the human MDPs. We extend this framework to MHBC settings. Our contributions are: (1) We propose a novel, behaviorally grounded RL framework for an AI intervening on a human in multi-goal settings. We propose a simple model of human decision-making that captures fundamental behavioral mechanisms in multi-goal pursuit, such as feasibility-based goal prioritization (e.g. Ballard et al., 2016) and the impact of progress-making on disengagement (e.g. Roose & Williams, 2018). Importantly, we show that our human model captures three classes of realistic human behaviors observed in the empirical literature (e.g. Ballard et al., 2016; Schmidt & Dolis, 2009). (2) Using our framework, we intuitively characterize optimal AI interventions for different classes of human behaviors. (3) Finally, we show that we our method can personalize rapidly even when the latent construct of disengagement is unknown and must be estimated, as well as provide evidence that model-free learning fails to personalize rapidly in the MHBC setting. We also discuss how our work can be extended to provide practical insights for MHBC programs in real applications.

Related Works Previous work uses RL to personalize mHealth interventions in single health behavioral change settings where the context of the user determines what AI intervention (often a push notification) to deliver and when (e.g. Trella et al., 2024; Liao et al., 2020; Wang et al., 2021; Yom-Tov et al., 2017; Menictas et al., 2019; Zhou et al., 2018; Ghosh et al., 2024; Park & Lee, 2023; Tabatabaei et al., 2018). There are two significant challenges for RL applied to mHealth settings. First, one must achieve effective personalization within relatively few user interactions (e.g. Kankanhalli et al., 2021). Secondly, RL models and policies need to be interpretable to domain experts who interact with real patients (e.g. Nofshin et al., 2024; Liao et al., 2020). Many existing RL methods, when naively applied to mHealth, may learn slowly or produce policies that are not easily interpretable, as they model users as black-boxes (Nofshin et al., 2024).

Prior work addresses both challenges by modeling the human as a MDP planner, and AI interventions as changes to

[1]Harvard College, Harvard University, Cambridge, MA, USA [2] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. Correspondence to: Samantha Marks <samanthamarks@college.harvard.edu>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

the human MDP. Nofshin et al. introduce a flexible and behaviorally grounded framework, called "Behavior-Model RL" (BMRL), for single-goal mHealth settings. BMRL proposes a human MDP model that can incorporate behavioral change mechanisms. Through linking human behaviors to MDP parameters, BMRL allows for interpretation of the factors driving both human and AI decision-making. Nofshin et al. also show that the inductive bias provided by the human MDP can speed up online personalization. In this work, we extend the BMRL framework to multiple-goal settings.

Multiple health behavioral change (MHBC) programs promote multiple health behaviors at once and can be more effective than single-goal programs (Zhang et al., 2024; Dai et al., 2020; Sunderrajan et al., 2021; Wilson et al., 2014; Prochaska et al., 2008; Prochaska, 2008; Geller et al., 2017). Extending RL frameworks for single-goal settings to MHBC can be challenging, as human behavioral dynamics differ between the two settings. In MHBC, humans must prioritize goals and adaptively switch them (e.g. Scholer et al., 2024). Behavioral studies show that for MHBC, goal feasibility and resiliance (to set-backs) are important factors driving human decision-making (Scholer et al., 2024; Ballard et al., 2016; Schmidt & Dolis, 2009; Alister et al., 2024; Roose & Williams, 2018; Neal et al., 2017). We capture these two factors in our extension of BMRL.

2. Behavioral Model RL Framework for the Multi-Goal Setting

We assume that the human plans under a MDP, to which the AI has full access. The optimal policy of the human MDP may not reach the real-life goal of completing the treatment plan. To help, the AI can intervene upon the human MDP parameters.

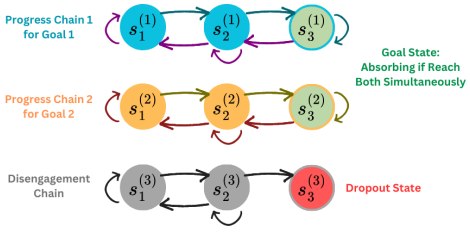


Figure 1. Toy human MDP: has 2 progress chains of length 3, each corresponding to a different subgoal, and 1 disengagement chain of length 3. The absorbing goal state s_g is at $(s_3^{(1)}, s_3^{(2)})$ (in green). The absorbing dropout state s_d is at $s_3^{(3)}$ (in red). The initial state $s_0 = (s_1^{(1)}, s_1^{(2)}, s_1^{(3)})$.

The Human MDP This is a *multi-progress chainworld*: $\mathcal{M}_h = \langle \mathcal{S}_h, \mathcal{A}_h, \mathcal{T}_h, \mathcal{R}_h, \gamma_h, s_g, s_d \rangle$. States represent progress on a goal, $s_h^{(P)}$, or a step towards disengagement, $s_h^{(D)}$. As in Figure 1, progress towards goals and towards disengagement are chains of 1-D, discrete states. The absorbing *goal state* s_g is at the end of all progress chains, representing completing all treatment goals. The absorbing *dropout* state s_d is at the end of the disengagement chain, representing quitting treatment. The initial state s_0 is at the start of each chain.

We limit the human to working on one goal at a time (or do nothing). *If the human does nothing*, it moves back 1 on each progress chain i with probability (w.p.) b_i, and moves forward 1 on the disengagement chain w.p. f_d. *If it works on progress chain* i, it moves forward on it w.p. f_i (success), and back on each other progress chain k w.p. b_k. Success on a progress chain results in a step back on the disengagement chain w.p. b_d (failure results in advancement on disengagement w.p. f_d). Details are in Appendix A.1. Importantly, *goal feasibility* is modeled by forward probabilities on progress chains and resilience is captured by f_d. Finally, the human gets a reward r_g for reaching s_g and a cost r_d for dropping out. See A.2 for modeling justification.

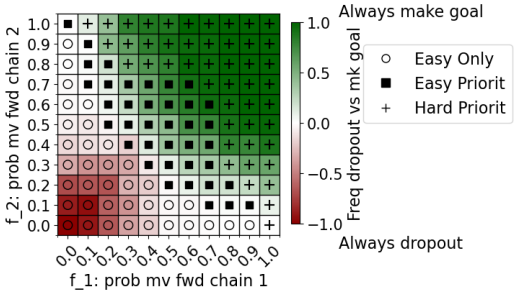


Figure 2. Human agent uses hard prioritization when easy to reach goal; easy prioritization when possible, but hard to reach goal; easy-only when too hard. Human's policy over outcome (frequency of reaching goal vs. drop out) for varying f_1 (x-axis) and f_2 (y-axis).

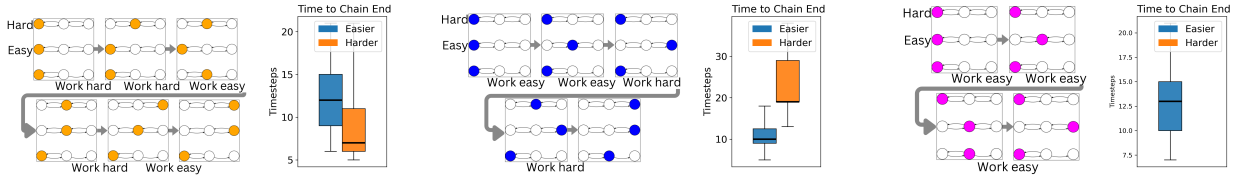
The AI Agent The AI agent modifies human MDP parameters to help the human reach the behavioral goal. The AI plans under the following MDP: $\mathcal{M}_{AI} = \langle \mathcal{S}_{AI}, \mathcal{A}_{AI}, \mathcal{T}_{AI}, \mathcal{R}_{AI}, \gamma_{AI} \rangle$. Each AI state consists of the human's *current* state and the *previous* human action. AI actions are: (1) do nothing, (2) intervene on the disengagement chain – decrease f_d (increase resilience) by $\Delta_{f_d AI}$, (3) intervene on progress chain i – increase f_i (perceived goal attainability) by $\Delta_{f_i AI}$. The AI intervenes on one chain at a time, and *the effect lasts for just the current timestep* (example in Appendix B). Note that AI actions are grounded in existing behavioral interventions (e.g. Scholer et al., 2024). The transition function of the AI is defined by: $\mathcal{T}_{AI}(s_{AI}, a_{AI}, s'_{AI}) = P(s'_h | s_h, a_h, a_{AI}) \cdot P(a'_h | s_h, a_{AI}) = \mathcal{T}_h(s_h, a_h, s'_h | a_{AI}) \cdot \pi_h(a'_h | s_h, a_{AI})$ where given the AI's intervention action, $\mathcal{T}_h$ and π_h are the human transitions and policy under that intervention. Finally, the AI receives reward r_g if the human reaches the goal state and cost r_d if the human drops out. It gets cost r_{inter} if it intervenes to limit excessive interventions.

3. Modeling Human Decision-Making in MHBC Settings

We begin by demonstrating how our proposed multi-progress chainworld captures realistic human behaviors and the parameter settings that give rise to them. In the next section, we will discuss how the AI intervenes on these behaviors.

We instantiate the human MDP with 2 progress chains and a disengagement chain of length 5. We set $r_g = -r_d = 10$ and $\gamma_h = 0.9$, $b_1 = b_2 = 0.3$, $b_d = 0.5$, $f_d = 0.4$. We solve for the optimal policy using value iteration (VI), under which we sampled 100 trajectories. See Appendix C.1 for details. The *easy chain* is the one which has a higher probability of moving forward, and the *hard chain* is the one with lower probability (see Appendix C.2 for details).

Our human model captures three realistic behavioral strategies and outcomes. (1) *Hard prioritization*: the human agent first works on the hard chain, switching to the easy one when disengagement is too high. Once disengagement decreases, the human resumes working on the hard chain until completion, then completes the easy chain. This corresponds to reaching s_g. (2) *Easy prioritization*: the human works on the easy chain until completion, then switches to the hard one. If disengagement becomes high, the human switches back to the easy chain and either uses the same strategy as in (1), or, if disengagement does not decrease, it will maintain progress on the easy chain forever. This corresponds to reaching s_g or just the end of the easy chain. (3) *Easy only*: the human only works on the easy chain. See Figures 2 and 3 for visualizations of all strategies.



(a) Hard prioritization policy. (b) Easy prioritization policy. (c) Easy only policy.

Figure 3. The 3 policy classes: each shows characteristic trajectory (left), empirical order of finishing chains (right).

Chain difficulty determines human policy class and outcome. In Figure 2 we see that the *hard prioritization* policy class corresponds to reaching s_g, and that it occurs when both chains are easy to work on (f_1 and f_2 are high). The *easy prioritization* class corresponds to reaching the goal or working on the easy chain forever to avoid dropout (see Appendix C.4 for details). It occurs when one chain is easy and the other hard (or when both are medium difficulty). Lastly, the *easy only* policy class corresponds to working on the easier chain forever or dropping out and occurs when one or both chains are very hard. These results generalize to all transition parameters (see Appendix C.3).

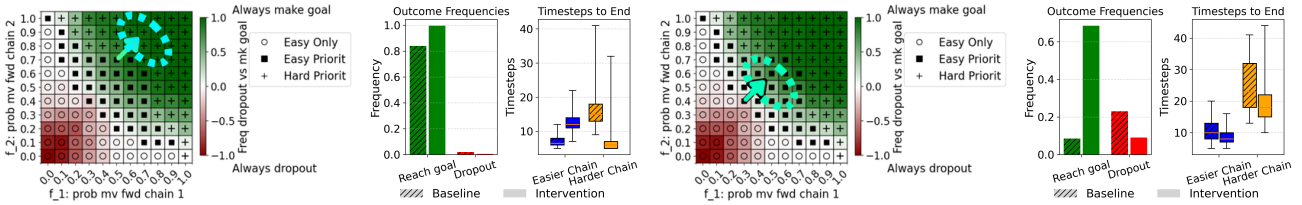
Our human MDP captures realistic, observed behavior. Behavioral science studies have found that when humans are given two goals they perceive as doable, they will allocate more time to the task further from completion, but switch to the easier task when the harder task is perceived as too difficult (i.e. when not making progress) (Schmidt & Dolis, 2009; Ballard et al., 2016; Neal et al., 2017; Alister et al., 2024). This roughly corresponds to the hard prioritization policy. When the participant perceives only one (or neither) goal as attainable, they will focus first on finishing the easier task (Schmidt & Dolis, 2009). This matches the easy prioritization and easy only policies. Studies also show that humans quit treatment when the tasks are too hard, corresponding to the agent dropping out when both tasks are very difficult (Zhang et al., 2024).

4. Designing AI Interventions for MHBC Settings

In this section, we characterize how the AI agent intervenes on the human, given the full human MDP. We instantiate the human MDP as from Section 3. We set $\Delta_{f_1 AI} = \Delta_{f_2 AI} = -\Delta_{f_d AI} = 0.2$, $r_{inter} = -0.1$, and $\gamma_{AI} = 0.9$. We solve for the optimal AI policy using VI, under which we sample the AI trajectories for up to 80 timesteps. Details in Appendix C.1.

Results: AI intervenes in intuitive ways. The human agent's three policy classes from worst to best are: easy only, easy prioritization, hard prioritization. Given these, there are two main classes of AI interventions: (1) When the human agent is close to the border of a better policy class, the AI nudges it into the better policy class. (2) Otherwise, the AI agent helps the human achieve the objective of each policy class, augmenting, but not changing, the human strategy. See Appendix D.1 for details. These results are depicted in Figure 4 for the easy prioritization policy class, see Appendix D.2 for the others.

When the human's disengagement nears s_d, the AI will intervene in the manner which has the best chance of reducing disengagement; otherwise, it intervenes on the progress chain the human is working on. Once the AI starts intervening, it continues to do so, yielding the trend shown in Figure 10. Once the human can reach s_g, the AI intervenes more. As the



Easy prioritization on the **border** of hard prioritization class: AI nudges baseline human into hard prioritization class.

Easy prioritization **not** on the border of hard prioritization class: AI does not alter human agent's policy class.

Figure 4. **The AI agent nudges a human agent bordering the next-better strategy into it, and otherwise augments the human's strategy to make it more successful.** Left: what the baseline human's policy class is updated to under intervention. Right: frequency of the human reaching s_g vs s_d and number of timesteps to the end of the easier vs harder chains with and without AI interventions.

baseline human gets better at reaching s_g, the number of AI interventions decreases. Importantly, this means that our AI provides more support when the human agent struggles to get to the (reachable) goal. But, in practice, it may intervene too much. Overall, the AI can help the human complete their treatment plan, pushing them to use a better goal-prioritization policy or making the human strategy more successful.

5. AI Robustness under Disengagement Estimation

In this section, we demonstrate that the AI can still personalize when parts of the human model are unknown. In practice, the part most likely to be unknown is the human's level of disengagement, as it is not directly observable. However, we assume the progress towards each subgoal is known, which is reasonable because health experts typically design treatment plans with clear metrics to chart progress toward each health goal (e.g. Bailey, 2017; Teal et al., 2012). For example, the progress of a patient working on shoulder rehabilitation can be measured by their range of motion (Oosterwijk et al., 2018), and the progress of a patient working on depression reduction can be measured by the PHQ-9 (Economides et al., 2019).

Correspondingly, in this section, the AI agent only observes the human agent's progress chain states and actions. It does not observe the human's disengagement state, which is instead estimated using the sequential estimator defined in Appendix E.1. (Note that in Appendix E we also explore using a simpler estimator, but find the sequential estimator to perform better). The AI agent under estimation samples its action from its optimal policy at the estimated human state. We sample 500 trajectories for each parameter setting (see Appendix C.1 for human parameters; Section 4 for AI).

Results: AI Robustly Helps Human Reach Goal under Disengagement Estimation Estimating the disengagement state only slightly reduces the AI agent's ability to help the human reach s_g and slightly reduces the number of interventions it delivers. But, it decreases the AI agent's ability to prevent the human dropping out, and causes the human to dropout faster. This is shown for the easy prioritization policy class in Figure 5 and all others in Appendix E.5, where we also see that as the baseline human struggles less to reach s_g, the more robust the AI is to disengagement estimation; but when it cannot reach s_g, the AI is not robust to helping it reach the easier chain's end.

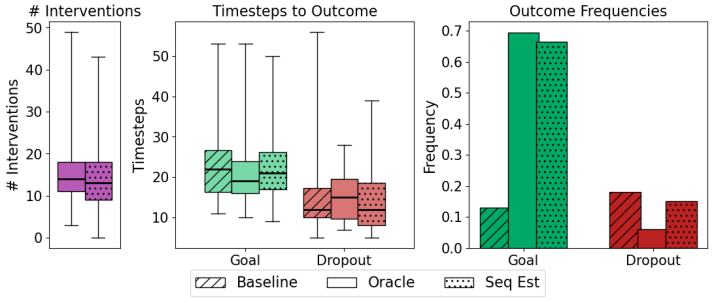


Figure 5. **Impact of disengagement estimation is reducing AI's ability to prevent human agent dropping out.** AI with full knowledge (oracle) vs. disengagement estimation (seq est) by number of interventions (left), impact on how fast the human hits s_g and s_d as against its baseline (middle), and frequency of human dropping out vs. reaching goal (right). This is for the easy prioritization policy class, the class in which the AI has the most impact on the human agent.

6. Model-free learning does not personalize in MHBC

The AI agent must learn online in reality. We show that when it uses model-free learning, it fails to personalize rapidly, using an AI agent which does not know the human model and learns its policy via epsilon-greedy Q-learning (set-up in Appendix F.1) and 500 trials of 1000 episodes each. The oracle AI's optimal policy is found via VI as in Section 4. We note that unlike the oracle AI and AI under disengagement estimation, the AI agent under Q-learning does not know the true human transitions and rewards, it simply observes the full human state (including disengagement level) and action at each timestep.

Model-free learning fails to rapidly personalize. Here, we show our results when the true human agent belongs to the easy prioritization policy class, as the optimal AI agent helps human agents of this class reach s_g the most, enabling us to see the impact of Q-learning. Our results of Q-learning against the oracle AI are shown in Figure 6.

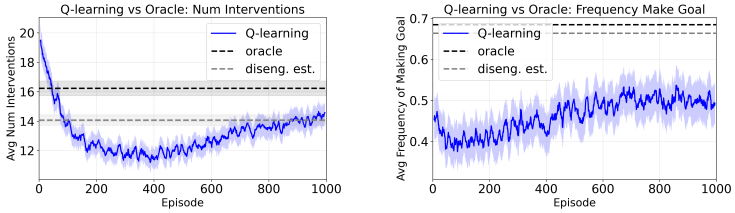


Figure 6. **Q-learning fails to effectively personalize rapidly.** Average number of interventions (left) and frequency of human reaching goal state under intervention (right) using AI policy from Q-learning vs oracle and disengagement estimation.

Our human model captures real multi-goal pursuit behavior. Yet, it is a simplification of a true human. If Q-learning fails to personalize on it–as it does here (and for all other policy classes, as shown in Appendix F.3)–, it is unlikely to personalize to an actual human. Hence it is important to provide inductive bias that captures real human mutli-goal pursuit dynamics, like our model, for enabling the AI to learn effectively and rapidly.

7. Discussion and Conclusion

We propose a novel, behaviorally grounded RL framework for optimizing and personalizing AI interventions in multi-goal mHealth settings. We show that our model of human decision-making captures observed behaviors of real humans. Using our model, we demonstrate that optimal AI interventions can be intuitively characterized given the human behavior class. We show that our analysis is robust to when the human's disengagement level is estimated. Lastly, we provide evidence that our framework provides a useful inductive bias for online learning by showing that model-free learning does not personalize well in the MHBC setting.

Limitations. Our multi-progress chainworlds captures a few (but realistic) apsects of human decision-making in MHBC settings. We are potentially missing other important aspects of multi-goal pursuit dynamics, e.g. goal valuation (Ballard et al., 2016). Furthermore, we assume that the parameters of the human MDP is constant over time; but in practice, human traits can experience changes. We also treat each behavioral change goal as independent, but it has been found that progress towards one goal affects progress on another (Scholer et al., 2024). Moreover, our empirical results are from a relatively simple multi-progress chainworld. We also do not include the impact of over-intervening on disengagement, an important factor to consider in the mHealth setting (Trella et al., 2022). Lastly, although we explore what happens when we do not know the human's level of disengagement, we assume knowledge of the transition dynamics and reward functions. Deeper empirical analysis is needed into how rapid personalization fares when these are unknown.

Future Works. Future work should formally validate our modeling assumptions through user studies. One key future adjustment is incorporating goal valuation, as well as scaling to more complicated dynamics such as by exploring direct dependencies of progress-making between chains and transition parameters which vary over time, in addition to having more progress chains. We are also interested in the sensitivity of our analysis to transition misspecification. Finally, we are excited to explore our framework for fully online personalization. We believe we can leverage the three human policy classes to inform priors on human MPD parameters to enable rapid personalization when transition dynamics and reward functions are unknown in MHBC settings.

Software and Data

The code can be found at the GitHub repository here: `https://github.com/samantha-marks/multiprogress-chainworld-workshop/tree/main`. All data was synthetically generated using this code.

Acknowledgements

We thank Susan Murphy and Siddharth Swaroop for their valuable feedback and guidance during the development of this work.

References

Alister, M., Herbert, S. L., Sewell, D. K., Neal, A., and Ballard, T. The impact of cognitive resource constraints on goal prioritization. *Cognitive Psychology*, 148:101618, 2 2024. ISSN 0010-0285. doi: 10.1016/J.COGPSYCH.2023.101618.

Bailey, R. R. Goal setting and action planning for health behavior change. *American Journal of Lifestyle Medicine*, 13(6):615–618, September 2017. doi: 10.1177/1559827617729634. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC6796229/.

Ballard, T., Yeo, G., Loft, S., Vancouver, J. B., and Neal, A. An integrative formal model of motivation and decision making: The mgpm. *Journal of Applied Psychology*, 101:1240–1265, 9 2016. ISSN 00219010. doi: 10.1037/apl0000121.

Bouton, M. E. Why behavior change is difficult to sustain, 11 2014. ISSN 10960260.

Brandstätter, V. and Bernecker, K. Persistence and disengagement in personal goal pursuit. *Annual Review of Psychology*, 73:271–299, 2022. doi: 10.1146/annurev-psych-020821-110710. URL https://doi.org/10.1146/annurev-psych-020821-110710.

Dai, W., Palmer, R., Sunderrajan, A., Durantini, M., Sánchez, F., Glasman, L. R., Chen, F. X., and Albarracín, D. More behavioral recommendations produce more change: A meta-analysis of efficacy of multibehavior recommendations to reduce nonmedical substance use. *Psychology of Addictive Behaviors*, 34(7):709–725, 2020. doi: 10.1037/adb0000586. URL https://doi.org/10.1037/adb0000586.

Economides, M., Ranta, K., Nazander, A., Hilgert, O., Goldin, P. R., Raevuori, A., and ... Long-term outcomes of a therapist-supported, smartphone-based intervention for elevated symptoms of depression and anxiety: Quasiexperimental, pre–postintervention study. *JMIR mHealth and uHealth*, 7(8):e14284, 2019. doi: 10.2196/14284. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6733157/.

Geller, K., Lippke, S., and Nigg, C. R. Future directions of multiple behavior change research. *Journal of Behavioral Medicine*, 40:194–202, 2 2017. ISSN 15733521. doi: 10.1007/S10865-016-9809-8/FIGURES/1. URL https://link.springer.com/article/10.1007/s10865-016-9809-8.

Ghosh, S., Guo, Y., Hung, P.-Y., Coughlin, L., Bonar, E., Nahum-Shani, I., Walton, M., and Murphy, S. Miwaves reinforcement learning algorithm, 2024. URL https://arxiv.org/abs/2408.15076.

Kankanhalli, A., Xia, Q., and Zhao, X. Understanding personalization for health behavior change applications: A review and future directions. *AIS Transactions on Human-Computer Interaction*, pp. 316–349, 9 2021. ISSN 19443900. doi: 10.17705/1thci.00152. URL https://aisel.aisnet.org/thci/vol13/iss3/3.

Liao, P., Greenewald, K., Klasnja, P., and Murphy, S. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4, 3 2020. ISSN 24749567. doi: 10.1145/3381007.

Menictas, M., Rabbi, M., Klasnja, P., and Murphy, S. 4.0 (cc by-nc-nd) Artificial Intelligence Artificial intelligence decision-making in mobile health, 2019. URL http://portlandpress.com/biochemist/article-pdf/41/5/20/858255/bio041050020.pdf.

Neal, A., Ballard, T., and Vancouver, J. B. Dynamic self-regulation and multiple-goal pursuit. *Annual Review of Organizational Psychology and Organizational Behavior*, 4(Volume 4, 2017):401–423, 2017. ISSN 2327-0616. doi: https://doi.org/10.1146/annurev-orgpsych-032516-113156. URL https://www.annualreviews.org/content/journals/10.1146/annurev-orgpsych-032516-113156.

Nofshin, E., Swaroop, S., Pan, W., Murphy, S., and Doshi-Velez, F. Reinforcement learning interventions on boundedly rational human agents in frictionful tasks. 2024. URL https://arxiv.org/abs/2401.14923.

Oosterwijk, A., Nieuwenhuis, M., van der Schans, C., and Mouton, L. Shoulder and elbow range of motion for the performance of activities of daily living: A systematic review. *Physiotherapy Theory and Practice*, 34(7):505–528, 2018. doi: 10.1080/09593985.2017.1422206. URL https://doi.org/10.1080/09593985.2017.1422206. PMID: 29377745.

Park, J. and Lee, U. Understanding disengagement in just-in-time mobile health interventions. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7, 6 2023. ISSN 24749567. doi: 10.1145/3596240.

Prochaska, J. J., Spring, B., and Nigg, C. R. Multiple health behavior change research: An introduction and overview. *Preventive medicine*, 46:181, 3 2008. ISSN 00917435. doi: 10.1016/J.YPMED.2008.02.001. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC2288583/.

Prochaska, J. O. Multiple health behavior research represents the future of preventive medicine. *Preventive Medicine*, 46: 281–285, 3 2008. ISSN 0091-7435. doi: 10.1016/J.YPMED.2008.01.015.

Roose, K. M. and Williams, W. L. An evaluation of the effects of very difficult goals. *Journal of Organizational Behavior Management*, 38:18–48, 2018. ISSN 1540-8604. doi: 10.1080/01608061.2017.1325820. URL https://www.tandfonline.com/action/journalInformation?journalCode=worg20.

Schmidt, A. M. and Dolis, C. M. Something's got to give: The effects of dual-goal difficulty, goal progress, and expectancies on resource allocation. *Journal of Applied Psychology*, 94:678–691, 5 2009. ISSN 00219010. doi: 10.1037/A0014945.

Scholer, A. A., Hubley, C., and Fujita, K. A multiple-goal framework for exploring goal disengagement. *Nature Reviews Psychology* —, 3:741–753, 2024. doi: 10.1038/s44159-024-00363-4. URL https://doi.org/10.1038/s44159-024-00363-4.

Sharma, A. K., Baig, V. N., Ahuja, J., Sharma, S., Panwar, R. B., Katoch, V. M., and Gupta, R. Efficacy of ivrs-based mhealth intervention in reducing cardiovascular risk in metabolic syndrome: A cluster randomized trial. *Diabetes Metabolic Syndrome: Clinical Research Reviews*, 15(5):102182, 2021. ISSN 1871-4021. doi: https://doi.org/10.1016/j.dsx.2021.06.019. URL https://www.sciencedirect.com/science/article/pii/S1871402121002022.

Sunderrajan, A., White, B., Durantini, M., Sanchez, F., Glasman, L., and Albarracín, D. Complex solutions for a complex problem: A meta-analysis of the efficacy of multiple-behavior interventions on change in outcomes related to hiv. *Health Psychology*, 40(9):642–653, 2021. doi: 10.1037/hea0001088. URL https://doi.org/10.1037/hea0001088.

Tabatabaei, S. A., Hoogendoorn, M., and van Halteren, A. Narrowing reinforcement learning: Overcoming the cold start problem for personalized health interventions. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11224 LNAI, pp. 312–327. Springer Verlag, 2018. ISBN 9783030030971. doi: 10.1007/978-3-030-03098-8_19.

Teal, C. R., Haidet, P., Balasubramanyam, A. S., Rodriguez, E., and Naik, A. D. Measuring the quality of patients' goals and action plans: development and validation of a novel tool. *BMC Medical Informatics and Decision Making*, 12 (152), December 27 2012. doi: 10.1186/1472-6947-12-152. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3544573/.

Trella, A. L., Zhang, K. W., Nahum-Shani, I., Shetty, V., Doshi-Velez, F., and Murphy, S. A. Designing reinforcement learning algorithms for digital interventions: Pre-implementation guidelines. *Algorithms*, 15, 8 2022. ISSN 19994893. doi: 10.3390/a15080255.

Trella, A. L., Zhang, K. W., Carpenter, S. M., Elashoff, D., Greer, Z. M., Nahum-Shani, I., Ruenger, D., Shetty, V., and Murphy, S. A. Oralytics reinforcement learning algorithm. 6 2024. URL http://arxiv.org/abs/2406.13127.

Wang, S., Zhang, C., Kröse, B., and van Hoof, H. Optimizing adaptive notifications in mobile health interventions systems: Reinforcement learning from a data-driven behavioral simulator. *Journal of Medical Systems*, 45, 12 2021. ISSN 1573689X. doi: 10.1007/s10916-021-01773-0.

Wilson, K., Senay, I., Durantini, M., Sánchez, F., Hennessy, M., Spring, B., and Albarracín, D. When it comes to lifestyle recommendations, more is sometimes less: A meta-analysis of theoretical assumptions underlying the effectiveness of interventions promoting multiple behavior domain change. *Psychological bulletin*, 141:474, 3 2014. ISSN 00332909. doi: 10.1037/A0038295. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4801324/.

Yom-Tov, E., Feraru, G., Kozdoba, M., Mannor, S., Tennenholtz, M., and Hochberg, I. Encouraging physical activity in patients with diabetes: Intervention using a reinforcement learning system. *Journal of Medical Internet Research*, 19, 10 2017. ISSN 14388871. doi: 10.2196/JMIR.7994. Looks like its the first to personalize SMS messages for treatment plan.

Zhang, A. L., Liu, S., White, B. X., Liu, X. C., Durantini, M., Chan, M.-P. S., Dai, W., Zhou, Y., Leung, M., Ye, Q., O'keefe, D., Palmese, L., and Albarracín, D. Health-promotion interventions targeting multiple behaviors: A meta-analytic review of general and behavior-specific processes of change. *Psychological Bulletin*, 2024. doi: 10.1037/bul0000427.supp.

Zhou, M., Mintz, Y., Fukuoka, Y., Goldberg, K., Flowers, E., Kaminsky, P., Castillejo, A., and Aswani, A. Personalizing mobile fitness apps using reinforcement learning hhs public access, 2018. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC7220419/.

A. Human Model Additional Details

A.1. Human Transitions

A.1.1. Example Human Trajectory

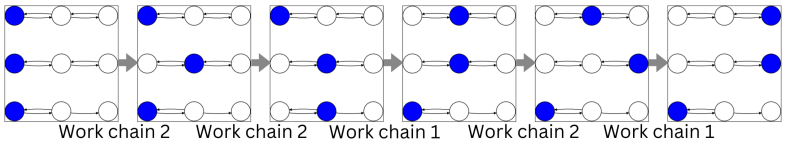


Figure 7. Example of human agent's trajectory from initial state to goal state. The state at each timestep is represented by the 3 circles colored blue in the state space square. Below the state space squares are the actions that the agent is taking.

A.1.2. Transitions Decision-Tree

Transitions. Transitions are according to the decision tree in the block below. The first split is by what action is taken.

If work on progress chain i:

(1) Will either succeed or fail to make progress on progress chain i, **and**

(2) will also have a b_k probability of moving back on each other progress chain k, independently, excluding progress chain i ($k \in \{1, \ldots, n\} \setminus \{i\}$); if do not move back on a progress chain k, will stay put on it.

Splitting up the two cases in (1):

- Will succeed to make progress (move 1 state forward on this progress chain) with probability f_i. If succeed, then one of two things will happen on the disengagement chain:
  - Will move 1 state **back** on the disengagement chain with probability b_d (getting more engaged).
  - Otherwise will stay put on the disengagement chain.
- Otherwise will fail to make progress, staying put on progress chain i. If succeed, then one of two things will happen on the disengage chain:
  - Will move 1 state **forward** on the disengagement chain with probability f_d (getting more disengaged).
  - Otherwise will stay put on the disengagement chain.

Otherwise, did no work:
(1) On each progress chain i, independently have a probability b_i of moving 1 state back; if do not move back on a progress chain i, will stay put on it, **and**

(2) will also move forward 1 state on the disengagement chain with probability f_d; otherwise stay put on it.

Note that there are some edge cases for when the human agent is at the end of a chain. If the human agent is at the start of a chain, it cannot move backwards on it. If the human is at the end of some progress chain i, then it cannot move forwards on it. When the human is at the end of progress chain i but not at the end of all progress chains, if the human agent works on progress chain i, it will stay in place on the disengagement chain.

Note the compound logic here. For example, suppose there is 1 progress chain and 1 disengagement chain.

If the human agent works on progress chain 1, the probability of succeeding to make progress on it *and* moving 1 state back on the disengagement chain is $f_1 \cdot b_d$. Another outcome from working on chain 1 is succeeding to make progress on it *and* staying put on the disengagement chain, which occurs with probability $f_1 \cdot (1 - b_d)$. There are still two more outcomes that could happen in this set-up if work on progress chain 1. The first is failing to make progress on chain 1 (staying in place on it) *and* moving forward on the disengagement chain, which occurs with probability $(1 - f_1) \cdot f_d$. The second is failing to make progress on chain 1 *and* staying put on the disengagement chain, which occurs with probability $(1 - f_1) \cdot (1 - f_d)$.

Otherwise, if the human does no work, they might move back 1 on progress chain 1 *and* move forward on the disengagement chain, which would occur with probability $b_1 \cdot f_d$. Note that as per our decision tree above, there are still 3 more cases of what could happen (the 4 cases are all combinations of moving back or staying put on progress chain 1 and moving forward or staying put on the disengagement chain, which each have their own compound probabilities as per our decision tree).

For more examples of how our transitions work, see Figure 7 for a tracing out of a trajectory.

A.2. Discussion of Modeling Decisions in the Context of MHBC

Our modeling decisions were based on behavioral literature about how humans undergo behavioral change and pursue on multiple goals. As this is a multi-goal setting, we made one progress chain per goal, representing progress along that goal. This is interpretable to physicians, for example, physical therapists routinely measure patient progress along goals.

Furthermore, behavioral studies have found that resilience to not making progress are important in goal-pursuit decision-making (e.g. Scholer et al., 2024). This is because not making progress towards a goal tends to lead to not pursuing that goal and making progress tends to lead to continued pursuit (Roose & Williams, 2018), but different people perceive the impact of progress-making differently (Scholer et al., 2024). We model this using the disengagement chain, which tracks how close the human is to disengaging. To incorporate the impact of making versus not making progress on the disengagement level, we chose that when the human agent makes progress on a subgoal, they have a chance of becoming more engaged, and when they fail to make progress, they have a chance of becoming more disengaged. This also represents the fact the disengagement is a process (Brandstätter & Bernecker, 2022). To represent that the impact of progress-making on levels of frustration and disengagement vary by individual, we parameterize the probabilities of moving forward and back on the disengagement chain with f_d. We note that f_d captures human resilience because it represents the probability of becoming more *discouraged* (closer to dropping out) when not making progress. Hence human agents with higher f_d have lower resilience because they become discouraged more readily when not making progress.

Additionally, if the human agent does not work on a subgoal, they have some probability of losing progress on it—moving back 1 on that subgoal's progress chain—to represent that when working on a behavioral change goal, humans have a tendency to revert back to their original behavior if they do not continue to work on it and reinforce the new behavior (Bouton, 2014).

To give rise to goal-prioritization behavior, we limited the human agent to working on one goal at a time, which also corresponds to the fact the humans can only work on a limited number of tasks at a time. And to incorporate feasibility-based goal-prioritization behavior, we parameterized the probability of moving forward on each progress chain (making progress on each subgoal). A higher probability of moving forward on a progress chain corresponds to an easier task, and a lower probability corresponds to a harder task which requires more effort.

B. AI Agent Additional Details: Transition Example

Recall that the effect of the AI agent's action on the human agent's transition parameters lasts for just the current timestamp. For example, in our setting, if the AI agent increases the human's probability of moving forward on the first subgoal, f_1, in the current time step to yield $f_1^{\Delta_{AI}} = f_1 + \Delta_{AI}$, the human's probability of moving forward on this goal will have reverted to its original value f_1 at the next time step. Hence what occurs in one timestep is the AI intervening, the human choosing its action according to its policy under that intervention, going to the next as drawn from the transition probabilities under intervention, yielding the next AI state. Then the cycle repeats, creating the trajectory under AI intervention.

C. Characterizing Human Model Behavior Additional Details

C.1. Human Policy Class Visualizations Parameters

The parameters used in this paper for visualizing the hard prioritization policy class are $f_1 = 0.6, f_2 = 0.8, f_d = 0.4$, the easy prioritization policy class are $f_1 = 0.3, f_2 = 0.4, f_d = 0.4$, and the easy only are $f_1 = 0.1, f_2 = 0.2, f_d = 0.4$. For the easy only policy class bordering the easy-prioritization policy class, we use $f_1 = 0.2, f_2 = 0.4, f_d = 0.4$. For the easy prioritization policy class bordering the hard prioritization policy class, we use $f_1 = 0.5, f_2 = 0.7, f_d = 0.4$.

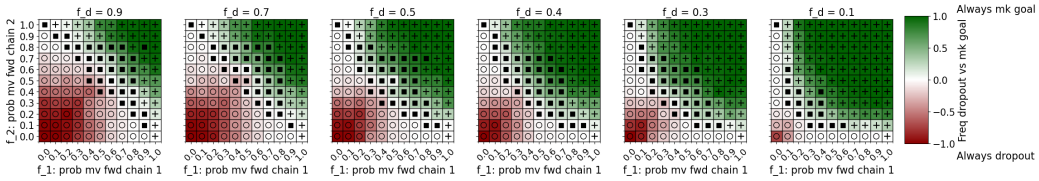
C.2. Easy Chain when Progress Chains are of the Same Length

We define the **easy chain** (or easier chain) as the progress chain which incurs the fewest expected number of movements forward on the disengagement chain when working from the start to the end of it. This is because the easier subgoal, the one corresponding to the easy chain, is the one that is the easiest for a human to complete without burning out.

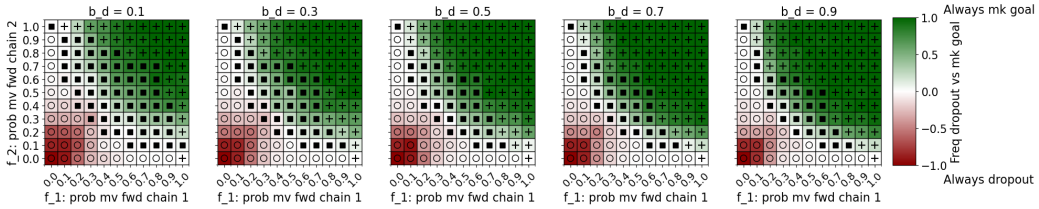
In this paper, we use progress chains of the same length. In this case, *the easy chain is the chain with the highest probability of moving forward on it*, f_i. This is because moving forward on the disengagement chain only occurs when failing to make progress. When the progress chains are of the same length, all that matters for expected number of movements forward on the disengagement chain when working on it is what the probability of moving forward on the stuck chain is at each step. This is $(1 - f_i) \cdot f_d$. So the progress chain with highest f_i leads to minimizing the probability of moving forward on the stuck chain when working on it, thus minimizing the expected number of movements forward on the stuck chain when working from the start to the end of it.

C.3. Impacts of Varying f_d, b_d, b_1, and b_2 on Human Model Behavior

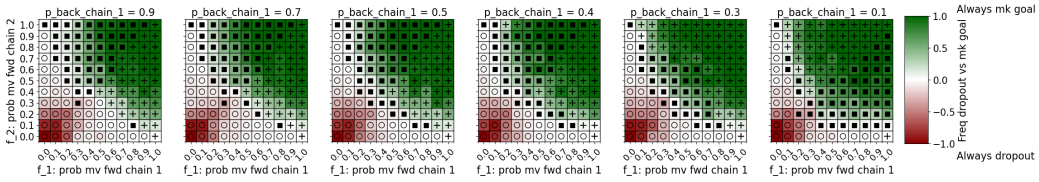
Here we show the impacts of varying the probability of moving forward on the disengagement chain f_d as well as moving back on the disengagement chain b_d, on chain 1 b_1, and on chain 2 b_2 are on the human agent's policy class and outcome.



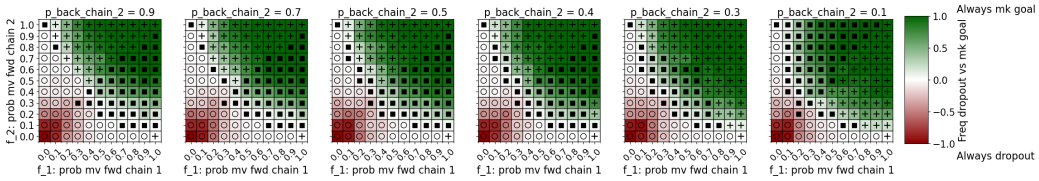
(a) **Low resilience means dropping out more, high resilience means reaching goal more.** Varying probability of moving forward on disengagement chain, f_d, from 1 (left) to 0 (right).



(b) **Getting re-engaged more readily means reaching goal slightly more.** Varying probability of moving back on disengagement chain, b_d, from 0 (left) to 1 (right).



(c) **Losing progress less on the first subgoal means reaching goal slightly more, especially when first subgoal is hard.** Varying probability of moving back on progress chain 1, b_1, from 1 to 0.



(d) **Losing progress less on the second subgoal means reaching goal slightly more, especially when second subgoal is hard.** Varying probability of moving back on progress chain 2, b_2, from 1 to 0.

Figure 8. **Probability of moving forward on all chains has the biggest impact on whether the human reaches the goal state, drops out, or works on the easier chain forever.** Human agent's policy (easy only (◯), easy prioritization (■), and hard prioritization (+)) overlaid onto its outcome (frequency of making goal vs. dropping out) by varying human MDP parameters. Each grid has varying f_1 (x-axis) and f_2 (y-axis). Each row of plots is for varying some other transition probability: the first is by decreasing f_d, the second is by increasing b_d, the third is by decreasing b_1, and the fourth is by decreasing b_2, showing the effect of all transition probability parameters on the human's outcome. Note that by default $b_1 = b_2 = 0.3$ and $b_s = 0.5$, unless they are otherwise explicitly varied.

From Figure 8, we can see that increasing f_d leads to dropping out significantly more and that increasing b_d, b_1, or b_2 leads to reaching the goal slightly more. Of these four variables, we can see that varying f_d has the largest impact on whether the human agent drops out versus reaches the goal. We also consistently see that varying the probabilities of moving forward on the chains, f_1 and f_2, which corresponds to varying the difficulty of completing the subgoals these chains belong to, has a large impact on the human agent's policy class and outcome. We also see that varying b_1 and b_2 has an effect on skewing where the easy prioritization versus hard prioritization classes can be found. We note that when we vary the probability of moving back on a progress chain, we hold the other fixed at 0.3. So when $b_1 \neq b_2$, we see that when the chain that has a lower probability of moving back on it also has a higher probability of moving forward on it, the human agent completes this task first, before moving on to try the one that it has less probability of moving forward on. That is, it prefers to do the task that is both easier and it is less likely to lose progress when not working on it first.

C.4. There are exactly 3 possible absorbing human behaviors

The first two possible absorbing behaviors are reaching the goal state s_g and the dropout state s_d. These are absorbing because s_g and s_d are absorbing states.

The third possible absorbing behavior is working on a chain forever. This is because per the definition of our transition dynamics for the edge cases (see A.1.2): if the human agent works on a progress chain that it is at the end of, while being at the beginning of the other(s), it is guaranteed to stay in place on all chains: it must stay put on the progress chain it is working on as well as on the disengagement chain since it cannot move forward on the progress chain it is working on, and it must stay put on the progress chain(s) it is not working on because it cannot move backwards on it. Hence if the human agent is following a policy which says to work on a progress chain the human is at the end of while at the start of the other progress chains, it will stay in place working on this chain forever, yielding the third absorbing behavior.

Per the definition of our transition dynamics, are no other ways for the human agent to stay in place forever, as it will have some probability of moving forward or back on a chain.

C.4.1. Impact on Easy Prioritization and Easy Only Policy Classes

Recall that the easy prioritization policy works on the easier chain until completion before working on the harder chain. Given the result above, this provides it the safety of being able to work on the easier chain forever when it is close to dropping out in order to avoid the cost r_d of dropout. Note that this explains why the easy prioritization and easy-only policies occur when it is difficult for the human agent to reach the end of at least one progress chain without dropping out.

D. Characterizing the AI Agent's Interventions Additional Details

D.1. Impact of AI Interventions when Human Agent is not on the Border of a Better Policy Class

When the human agent is not on the border of the next-better policy class, the AI agent helps the human agent achieve the objective of each policy class more successfully, augmenting, but not changing, its strategy.

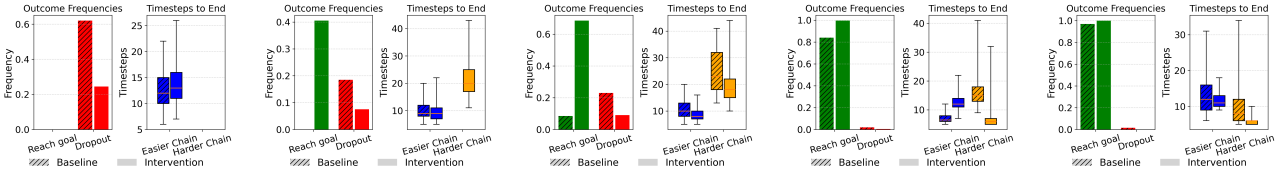
For hard prioritization: this means helping the human agent work on harder chain for longer by intervening on this chain, and then when it needs to switch to the easier chain because it's close to dropout, the AI also intervenes to help it get more engaged faster (increasing the probability of moving back on the disengagement chain).

For easy prioritization: the AI helps the human successfully reach the end of the easier chain first by intervening on this chain, then it helps the human work on the hard chain and get to its end more successfully (also intervening to help it get more engaged when it decides to switch to the easier chain when close to dropout).

For easy only: the AI helps the human reach the end of the easier chain only (intervening only on this chain), it does not attempt to nudge the human into working on the harder chain.

D.2. Result of AI's Interventions on all Policy Class Cases

Here we show the AI agent's impact on the human agent across all three policy classes, as well as when the human agent has transition parameters bordering the next-better policy class. Recall that the ordering of policy classes from worst to best in terms of the frequency of the human agent reaching the goal state are the easy only policy class, easy prioritization policy class, and hard prioritization policy class. The results of the AI's interventions on the baseline human for all these policy class cases are shown in Figure 9. See C.1 for the parameters of each policy class.



(a) Easy only not on the border of easy prioritization.

(b) Easy only on the border of easy prioritization class.

(c) Easy prioritization not on the border of hard prioritization.

(d) Easy prioritization on the border of hard prioritization.

(e) Hard prioritization.

Figure 9. **The AI agent nudges a human agent bordering the next-better strategy into it, and otherwise augments the human's strategy to make it more successful.** For each subplot: left: what the baseline human's policy class is updated to under intervention; right: frequency of the human reaching s_g vs s_d and number of timesteps to the end of the easier vs harder chains with and without AI interventions.

In Figure 9, we see that for all policy classes not on the border of the next-better policy class, the AI does not change their policy class, it just augments their strategy to help them more successfully complete their objective. But when they are on the border, the AI nudges them into the next-better policy class.

For the easy only policy class not on the border of the easy prioritization class, the human agent under AI intervention more readily reaches the end of the easier chain, but does not attempt the harder chain. But when it is on the border of the easy prioritization class, we see that the AI nudges the human to try working on the harder chain after finishing the easier chain, pushing it into the easy prioritization policy class.

For the easy prioritization policy class not on the border of the hard prioritization class, the human agent under AI intervention reaches the goal more frequently and drops out less frequently, but its strategy does not change: it still works on the easier chain until completion before working on the harder chain. But when it is on the border of the hard prioritization class, we see that the AI nudges the human to work on the harder chain first, pushing it into the hard prioritization policy class.

Lastly, for the hard prioritization policy class, we see the AI agent helps the human reach the goal state slightly more frequently, and do so faster.

D.3. Number of AI Interventions by Human Policy Class

See figure 10.

E. AI Robustness under Disengagement Estimation Additional Details

When we estimate the disengagement state, we assume that the AI has full knowledge of the human agent's progress chains states and actions. Hence we use the information about the human agent's previous states and actions to estimate the current

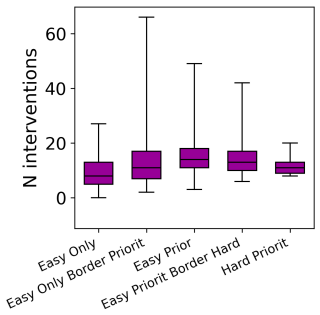


Figure 10. **AI intervenes more when the human struggles to, but can, get to s_g.** Number of AI interventions by policy class from least (left) to most (right) likely to reach s_g.

disengagement state. Note that we are assuming that there is one disengagement chain.

E.1. The Sequential Estimator

Given the sequence of human progress chain states and actions as well as its policy, the sequential estimator finds the disengagement state with the highest probability for the current timestep:

$$\hat{s}_k^{(\mathrm{D})}{}_{\mathrm{seq}} = \arg\max_{s_k^{(D)}} P\left[s_k^{(D)} = x_k \,\middle|\, \mathcal{H}_{k-1}, \pi_h\right] \tag{1}$$

where $\mathcal{H}_{k-1} = (s_1, a_1), (s_2^{(P)}, a_2), \ldots, (s_{k-1}^{(P)}, a_{k-1}), s_k^{(P)}$ is the history of human progress chain states and actions from the initial state s_1 leading up to the current timestep k by following policy π_h.

E.1.1. Derivation of the Sequential Estimator into Functional Form

Here we show how we can expand expression 1 into functional form.

We start by expressing the joint probability function for $s_2^{(D)}, \ldots, s_k^{(D)}$ under the human's deterministic policy π_h, given the previous progress states $s_2^{(P)}, \ldots, s_2^{(P)}$, initial state s_1, and human's transitions $\mathcal{T}$. We express it using:

$$\begin{aligned}
P(s_2^{(D)} &= x_2, \ldots, s_k^{(D)} = x_k | (s_1, a_1), (s_2^{(P)}, a_2), \ldots, (s_{k-1}^{(P)}, a_{k-1}), \pi_h) \\
&= P(s_1^{(D)} | s_1) \Pi_{i=2}^{k} P(s_i^{(D)} = x_i | s_i^{(P)}, a_{i-1}, s_{i-1}) I(a_{i-1} = \pi_h(s_{i-1})) \\
&= \Pi_{i=2}^{k} P(s_i^{(D)} = x_i | s_i^{(P)}, a_{i-1}, s_{i-1}) I(a_{i-1} = \pi_h(s_{i-1}))
\end{aligned}$$

where each $s_i^{(P)}$ is the progress chains states (a tuple consisting of the state on each progress chain) and each s_i is a full state: a tuple consisting of the known state on each progress chain and the unknown disengagement state $s_i^{(D)}$, with the exception of s_1, where we know $s_1^{(D)} = 0$. Hence $P(s_1^{(D)} | s_1) = 1$ since s_1 is our defined initial full state: a state which consists of the first state on each chain.

We then estimate $s_k^{(D)}$ as the value of $s_k^{(D)}$ which maximizes the marginalized joint probability function, where we marginalize out $s_2^{(D)}, \ldots, s_{k-1}^{(D)}$. To do so, we must sum over all possible sequences of $s_2^{(D)}, \ldots, s_{k-1}^{(D)}$, yielding:

$$\begin{aligned}
P(s_k^{(D)} &= x_k | (s_1, a_1), (s_2^{(P)}, a_2), \ldots, (s_{k-1}^{(P)}, a_{k-1}), \pi_h) \\
&= \sum_{(s_2^{(D)}, \ldots, s_{k-1}^{(D)}) \in M} \Pi_{i=2}^{k} P(s_i^{(D)} | s_i^{(P)}, a_{i-1}, s_{i-1}^{(D)}, s_{i-1}^{(P)}) I(a_{i-1} = \pi_h(s_{i-1}))
\end{aligned}$$

where M is the set of all sequences $(s_2^{(D)}, \ldots, s_{k-1}^{(D)})$ such that $s_i^{(D)} \in \{1, \ldots, l_d\}, \forall i \in \{2, \ldots, k-1\}$.

Then, our estimate for $s_k^{(D)}$ is the value for it which maximizes the probability function above:

$$\hat{s}_k^{(D)}{}_{\text{seq}} = \text{argmax}_{s_k^{(D)}} \sum_{(s_2^{(D)},\ldots,s_{k-1}^{(D)})\in M} \Pi_{i=2}^{k} P(s_i^{(D)}|s_i^{(P)}, a_{i-1}, s_{i-1}^{(D)}, s_{i-1}^{(P)}) I(a_{i-1} = \pi_h(s_{i-1})) \tag{2}$$

E.1.2. Naive (Brute Force) Time Complexity

If we were to compute expression 2 exactly, it involves plugging all possible valid sequences of $s_2^{(D)}, \ldots, s_n^{(D)}$ into the expression (letting $n = k$). There are 3^{n-1} such sequences, because there are only three options for what each $s_i^{(D)}$, $i \in \{2, \ldots, k\}$ can be: $s_i^{(D)} \in \{\max(s_{i-1}^{(D)} - 1, 1), s_{i-1}^{(D)}, \min(s_{i-1}^{(D)} + 1, l_d)\}$, noting that $s_i^{(D)}$ must be bounded between the start of the disengagement chain (1) and the end of the disengagement chain (l_d). Each time we try a sequence, we multiply $n - 1$ terms together. Lastly, we iterate over all 3^{n-1} sequences to find the sequence which maximizes the likelihood function (which is $O(1)$ work each time, as we just check if an expression is greater than the greatest one we already found). This results in $3^{n-1} + n - 1 \cdot 3^{n-1} = O(3^n)$ computations. This matches our empirical runtime shown in Figure 11.

We note that this is computationally expensive and is especially problematic for long trajectories since we must compute this estimator at every timestep n of the trajectory.

E.1.3. A Dynamic Programming Algorithm for Efficiently Computing the Sequential Estimator

To resolve the time complexity issue, we created an algorithm for efficiently computing the disengagement state with the highest probability of being at in the current timestep. This utilizes a similar vein of logic as the forward-background algorithm used in the context of Hidden Markov Models. It is written in Algorithm 1. We let $s^{(D)}$ be the current timestep's disengagement state, l_d the length of the disengagement chain, $a_1, \ldots, a_{k-1}$ be the sequence of actions taken by the human agent up to timestep k, $s_1^{(P)}, \ldots, s_k^{(P)}$ be the sequence of the human agent's progress chain states up to timestep k, π_h be the human agent's policy, and $\mathcal{T}_h$ be the human agent's transitions.

Algorithm 1 Disengagment State Estimation: Sequential DP

Definition: Let $P[t][s^{(D)}]$ be at time t, the probability of being at disengagement state $s^{(D)}$.
Base Case: $P[1][1] = 1$ because at the initial timestep, we know the initial disengagement state is 0 by definition (so with probability 1), 1-indexing the array.
procedure DP(k, l_d, $a_1, \ldots, a_{k-1}, s_1^{(P)}, \ldots, s_k^{(P)}, \pi_h, \mathcal{T}_h$)
    Initialize $P[2..k][1..l_d] \leftarrow 0$
    $P[1][1] \leftarrow 1$
    for $t \leftarrow 1$ to k **do**
        for $s^{(D)} \leftarrow 1$ to $\min(t, l_d)$ **do**
            for $ps^{(D)}$ in $[s^{(D)} - 1, s^{(D)}, s^{(D)} + 1]$ **do**
                if $ps^{(D)} < 1$ or $ps^{(D)} > t - 1$ or $ps^{(D)} > l_d$ **then**
                    skip ▷ Out of bounds
                end if
                $P[t][s^{(D)}]$ += $\Pr(s^{(D)}|ps^{(D)}, \text{time}= t - 1) \cdot P[t-1][ps^{(D)}] \cdot I(\pi_h(ps) = a_{t-1})$ ▷ (see Expressions 3, 4 for non-shorthand first term)
            end for
        end for
    end for
    return $\text{argmax}_{s^{(D)}}(P[k, s^{(D)}])$ for $s^{(D)} = 1$ to l_d ▷ Find max value over all possible disengagement states
end procedure

Intuitive Explanation of DP Algorithm

Recall from Equation (2) that finding $P(s_k^{(D)}) = x$ for some integer $x \in \{1, \ldots, l_d\}$, where l_d is the length of the disengagement chain, requires summing over all possible sequences of $s_2^{(D)}, \ldots, s_{k-1}^{(D)}$ which end in $s_k^{(D)} = x$. We can use this structure to reduce the number of computations needed to find this probability. This is by breaking down the sequence

at each timestep i, $i \in \{1, \ldots, k\}$ to find $P(s_i^{(D)} = y)$ for some integer $y \in \{1, \ldots, l_d\}$. Once we know $P(s_i^{(D)} = y)$ $\forall y \in \{1, \ldots, l_d\}$, we can find $P(s_{i+1}^{(D)} = z)$ for some integer $z \in \{1, \ldots, l_d\}$ using Law of Total Probability (LOTP), conditioning on each possible value of $s_i^{(D)}$:

$$P(s_{i+1}^{(D)} = z) = \sum_{y \in B} P(s_{i+1}^{(D)} = z | s_i^{(D)} = y) \cdot P(s_i^{(D)} = y)$$

where B is the set $\{\max(1, z-1), z, \min(l_d, z+1)\}$. This follows from something we previously noted from how we defined transitions for the disengagement chain: $s_{i+1}^{(D)} \in \{s_i^{(D)} - 1, s_i^{(D)}, s_i^{(D)} + 1\}$, noting that $s_{i+1}^{(D)}$ is bounded to be within the disengagement chain (meaning it must be a value between 1 and l_d, hence the min and max in our previous expression). By symmetry, this means that $s_i^{(D)} \in \{s_{i+1}^{(D)} - 1, s_{i+1}^{(D)}, s_{i+1}^{(D)} + 1\}$.

Our algorithm takes advantage of this structure. It starts with the disengagement state whose probability is known as the base case: the initial disengagement state, $s_0^{(D)}$, which is defined to be 0. It then uses this to find the marginal probability of being at each possible disengagement state $s_t^{(D)} \in \{1, \ldots, l_d\}$ at each timestep t, starting from the second timestep, up until the current timestep k, using LOTP on what the previous disengagement states could be. At the end, the algorithm selects the disengagement state value, $s = x$, for the current timestep, k, which maximizes the probability of the disengagement state being x at timestep k under the human's policy π_h.

Note that all of our expressions above, including the probability calculations within the DP, are conditioned on knowing $(s_1, a_1), (s_2^{(P)}, a_2), (s_3^{(P)}, a_3), \ldots, (s_{k-1}^{(P)}, a_{k-1}), s_k^{(P)})$.

Examining line 12, we also note that Pr(disengagement state $s^{(D)}$|previous disengagement state $ps^{(D)}$, time$= t-1$) is computed conditionally on knowing the above, meaning that our shorthand actually signifies:

$$\Pr(s^{(D)} | ps^{(D)}, t-1) = \Pr(s_t^{(D)} | s_{t-1}^{(D)}, (s_1, a_1), (s_2^{(P)}, a_2), (s_3^{(P)}, a_3), \ldots, (s_{k-1}^{(P)}, a_{k-1}), s_k^{(P)})) \tag{3}$$

where we absorb the timestep that the disengagement state it occurs at into our standard notation for the disengagement state, so $s^{(D)}$ at time t becomes $s_t^{(D)}$ and $ps^{(D)}$ becomes $s_{t-1}^{(D)}$.

And in practice, this expression is actually computed using the definition of conditional probability with extra conditioning to put it in terms of expressions that we know (dropping the given information to just what is relevant to this calculation):

$$\begin{aligned} \Pr(s_t^{(D)} | s_{t-1}^{(D)}, s_{t-1}^{(P)}, a_{t-1}, s_t^{(P)}) &= \frac{\Pr[(s_t^{(P)}, s_t^{(D)}) | a_{t-1}, (s_{t-1}^{(P)}, s_{t-1}^{(D)})]}{\sum_{i=0}^{l_d - 1} \Pr[(s_t^{(P)}, i) | a_{t-1}, (s_{t-1}^{(P)}, s_{t-1}^{(D)})]} \\ &= \frac{\mathcal{T}[(s_{t-1}^{(P)}, s_{t-1}^{(D)}), a_{t-1}, (s_t^{(P)}, s_t^{(D)})]}{\sum_{i=0}^{l_d - 1} \mathcal{T}[(s_{t-1}^{(P)}, s_{t-1}^{(D)}), a_{t-1}, (s_t^{(P)}, i)]} \end{aligned} \tag{4}$$

where a_{t-1} is the previous action (the action taken at timestep $t-1$, where the current timestep is t), $s_{t-1}^{(P)}$ is a tuple consisting of the state on each progress chain at time $t-1$, $s_t^{(P)}$ is the same but for time t, and l_d is the length of the disengagement chain. Here, we are using 0-indexing. Note that $\mathcal{T}[s', a', s] = \Pr(s | s', a')$ represents the probability of transitioning to full state s (which consists of the state on each chain) from the previous full state s' by taking action a', which is an expression that we know.

Proof of Correctness

We will now prove the correctness of our algorithm. That is, we will prove that our algorithm correctly finds the disengagement state which maximizes the probability of being at it at time k under the human agent's policy π_h; transitions $\mathcal{T}_h$; and given all its previous actions since the start of its trajectory $a_1, \ldots, a_{n-1}$, progress states $s_1^{(P)}, \ldots, s_n^{(P)}$, and disengagement chain length l_d. We first will prove by induction that at any time $n \geq 1$, the algorithm correctly instantiates row n of the DP matrix P (assuming n is at most the number of timesteps the trajectory inputted lasted for) such that each element j contains

the probability of being at disengagement state j at time n under the human's policy π_h and its transitions $\mathcal{T}_h$, given all of its previous progress states and actions up to the current timestep.

Base Case: By definition of the initial disengagement state being 1, we know that ${\hat{s}_1^{(D)}}_{\text{seq}} = 1$. This means that at time $n = 1$, $P[1][1] \leftarrow 1$ (indicating that at time $n = 1$, ${\hat{s}_1^{(D)}}_{\text{seq}} = 1$ with probability 1), and all other entries in this row should be 0. Our algorithm correctly instantiates the first row of the matrix like this. Hence our claim holds for the base case where $n = 1$.

Inductive Hypothesis Suppose the algorithm correctly instantiates row k of the DP matrix P for some arbitrary $n = k \geq 1$.

Inductive Step: We will show that our claim holds for $k + 1$. Suppose the inductive hypothesis. Then we know that row k has been correctly instantiated. Hence each element $j \in \{1, \ldots, l_k\}$ of row k in DP matrix P correctly holds the likelihood of being at that disengagement state j at timestep k given the human agent's policy, transitions, and previous sequence of progress chain states and actions from timestep 1 to $k - 1$.

Our algorithm now iterates through each element j in row $k + 1$ of DP matrix P, filling it in with

$$\sum_{ps^{(D)} \in B} \Pr(s^{(D)} | ps^{(D)},\ \text{time} = k) \cdot P[k][ps^{(D)}] \cdot I(\pi_h(ps) = a_k) \tag{5}$$

where B is the set $\{s^{(D)} - 1, s^{(D)}, s^{(D)} + 1\}$ but restricted such that it excludes any values less than 1, greater than the previous timestep k or greater than the length of the disengagement chain l_d. Note that this is all possible previous disengagement states which could occur before the current disengagement state at time $k + 1$, $j = s^{(D)}$. This is by construction of the transitions where the next disengagement state must be one greater than, equal to, or less than the previous disengagement state–but a disengagement state must be contained within the disengagement chain (be of value between 1 and l_d) and it is impossible to reach a disengagement state whose value is greater than the timestep, since at each timestep we can only move at most 1 forward on the disengagement chain (increment the disengagement state value by 1). Furthermore, in expression 5, note that the first multiplicand is shorthand notation which actually signifies what is written in expression 3. And per our inductive hypothesis, $P[t - 1][ps^{(D)}]$ correctly contains the probability of being at that disengagement state $ps^{(D)}$ (for all possible $ps^{(D)}$) at timestep k given the human agent's policy, transitions, and previous sequence of progress chain states and actions from timestep 1 to $k - 1$.

Hence this expression applies Law of Total Probability (LOTP), conditioning on each possible previous disengagement state, to correctly calculate the likelihood of being at the current disengagement state $j = s^{(D)}$ at timestep $k + 1$ under the human's policy π_h (which is why we correctly multiply by the term $I(\pi_h(ps) = a_k)$, which ensures that the previous disengagement state $ps^{(D)}$ would result in taking action a_k under the human's policy π_h) and transitions $\mathcal{T}_h$ and given the sequence of actions and progress states from timestep 1 through $k + 1$. (See expression 4 for how conditioning on the previous disengagement state allows us to compute the first multiplicand, noting that this expression is correct by definition of conditional probability with extra conditioning). Therefore, by LOTP, our algorithm correctly fills in each element j of row $k + 1$ in DP matrix P.

Induction Conclusion Since we have proven the base case and shown that if our claim holds for $n = k$, then it holds for $k + 1$, we have hence proven that our claim holds for all $n \geq 1$ (with the restriction that n is less than the number of timesteps in the human agent's trajectory). That is, we have proven that our algorithm correctly fills in each element j of the nth row with the probability that the current disengagement state at time t is equal to j given the human agent's policy π_h, transitions $\mathcal{T}_h$, previous actions $a_1, \ldots, a_{n-1}$, and progress states $s_1^{(P)}, \ldots, s_n^{(P)}$.

With this concluded, recall we are using our algorithm to find the disengagement state which maximizes the probability of being at it at time k under the human agent's policy π_h; transitions $\mathcal{T}_h$; and given all its previous actions since the start of its trajectory $a_1, \ldots, a_{n-1}$, progress states $s_1^{(P)}, \ldots, s_n^{(P)}$, and disengagement chain length l_d. We know if plug in this value of the timestep k into our algorithm, alongside these other elements, our DP algorithm correctly fills in the likelihood of being at each disengagement state $s_k^{(D)} = j, \forall j \in \{1, \ldots, l_d\}$. It then returns the value of j which maximizes this likelihood, therefore correctly returning ${\hat{s}_k^{(D)}}_{\text{seq}}$. Hence, our algorithm is correct $\square$.

Time Complexity of DP Approach for Computing Sequential Estimator

The theoretical runtime of our DP algorithm to compute the sequential estimator at timestep n is $O(n \cdot l_d^2)$ where l_d is the length of the disengagement chain. This is because for n timesteps, for each possible disengagement state (of which there

are l_d), we multiply together three terms–doing so three times, one for each previous possible disengagement state. We note that the first term being multiplied actually entails doing l_d additions to form the denominator (see Expression 4). So the amount of work done each time is $O(l_d)$. As we do this $n \cdot l_d$ times, the runtime is $O(n \cdot l_d^2)$. This matches the empirical runtime shown in Figure 11.

This is a significant reduction from our original runtime of $O(3^n)$, especially since for a trajectory of length m, we must do this m times, for $n = \{1, 2, \ldots, m\}$.

E.2. A Simpler Estimator: The Iterative Estimator

In the iterative approach, to get an estimate of the disengagement state at timestep k of a trajectory, which we call $s_k^{(D)}$, we use the estimate of the disengagement state at timestep $k-1$, which is $s_{k-1}^{(D)}$. We note that we know the disengagement state at the first timestep: $s_1^{(D)} = 0$, as per our definition of our multichain world that our initial state consists of the first state on every chain. We estimate the disengagement state at each timestep k (except for the first timestep) using our full knowledge of the transition probabilities and the progress chain states. As per the Markov Property, if we know what the last full state was (including the disengagement state) and what the action taken there was, then we have all the knowledge we need concerning the transition to the next state. Here, we treat the estimate of the disengagement state at the previous full state s_{k-1} (the tuple of the state on all progress chains and the disengagement state) as the "known" disengagement state, giving us full knowledge of the previous state. Using our transition probabilities, noting that our human agent is selecting its action from a deterministic policy π_h, we have that the probability function for the next state s_k under policy π_h is:

$$P(s_k^{(D)} = x | (s_1, a_1) \ldots, (s_{k-1}, a_{k-1}), s_k^{(P)}) = \frac{P(s_k^{(D)} = x, s_k^{(P)} | s_{k-1}, a_{k-1})}{P(s_k^{(P)} | s_{k-1}, a_{k-1})} \cdot I((a_{k-1} = \pi_h(s_{k-1})))$$

where s_{k-1} is the tuple consisting of the state on each progress chain and the estimate of the disengagement state at timestep $k-1$ (noting which treat this estimate as the "known" value of the disengagement state). Our estimator for $s_k^{(D)}$ is the value of $s_k^{(D)}$ which maximizes this probability function:

$$\begin{aligned} \hat{s}_k^{(D)}{}_{\text{iter}} &= \text{argmax}_{s_k^{(D)}} \frac{P(s_k^{(D)}, s_k^{(P)} | s_{k-1}, a_{k-1})}{P(s_k^{(P)} | s_{k-1}, a_{k-1})} \cdot I((a_{k-1} = \pi_h(s_{k-1}))) \\ &= \text{argmax}_{s_k^{(D)}} P(s_k^{(D)}, s_k^{(P)} | s_{k-1}, a_{k-1}) \cdot I((a_{k-1} = \pi_h(s_{k-1}))) \end{aligned} \tag{6}$$

noting that the second line follows from the fact that the probability of $s_k^{(P)}$ given the previous action and state is a constant with respect to $s_k^{(D)}$, and hence does not affect where the maximum is with respect to $s_k^{(D)}$, meaning we can drop it.

E.2.1. How to Compute the Iterative Estimator

Note that

$$\begin{aligned} \hat{s}_k^{(D)}{}_{\text{iter}} &= \text{argmax}_{s_k^{(D)}} P(s_k^{(D)}, s_k^{(P)} | s_{k-1}, a_{k-1}) \cdot I((a_{k-1} = \pi_h(s_{k-1}))) \\ &= \text{argmax}_{s_k^{(D)}} \mathcal{T}_h(s_{k-1}, a_{k-1}, (s_k^{(P)}, s_k^{(D)})) \cdot I(a_{k-1} = \pi_h(s_{k-1})) \end{aligned}$$

which is how we compute this estimator in practice.

Furthermore, we note that there are only three options for what $s_k^{(D)}$ can be: $s_k^{(D)} \in \{\max(s_{k-1}^{(D)} - 1, 1), s_{k-1}^{(D)}, \min(s_{k-1}^{(D)} + 1, l_d)\}$, noting that $s_k^{(D)}$ must be bounded between the start and end of the disengagement chain. Hence we must only compute the probability function over these three possible values of $s_k^{(D)}$, making this approach computationally very easy.

E.2.2. TIME COMPLEXITY OF THE ITERATIVE ESTIMATOR

The time complexity of the iterative approach is $O(n)$, as $n-1$ times, we compute Expression 6 (computing $\hat{s}_2^{(D)}{}_{\text{iter}}$ by plugging $s_1^{(D)} = 0$ into the expression, then plugging in $\hat{s}_2^{(D)}{}_{\text{iter}}$ to get $\hat{s}_3^{(D)}{}_{\text{iter}}$, and so forth, up to $\hat{s}_n^{(D)}{}_{\text{iter}}$). Hence we multiply $n-1$ terms together. This matches the empirical time complexity shown in Figure 11.

E.3. Comparing the Runtime of the Iterative and Sequential Estimators

Previously, we found the theoretical runtimes for the estimators. Here, we compare their empirical runtimes, which is shown in Figure 11.

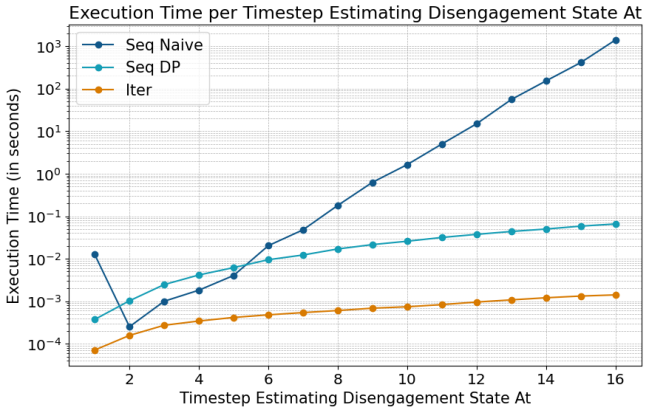


Figure 11. **DP approach for computing sequential estimator makes it nearly as fast as the iterative estimator.** Runtime for estimating disengagement state at each timestep under the sequential (seq) and iterative (iter) approaches, using both the naive and DP algorithms for computing the sequential estimator. Note that the y-axis uses a log-scale.

E.4. Comparing the Goodness of the Iterative and Sequential Estimators

In this section, we compare the iterative and sequential estimators in terms of their estimated expected bias, mean absolute error (MAE) and estimated variance. We aimed to compare the empirical bias and variance of our two estimators via simulation. To do so, we sampled 600 trajectories under the optimal policy (found by value iteration) of a human MDP with parameters $r_g = 10, r_d = -10, f_1 = 0.9, f_2 = 0.2, b_1 = 0.1, b_2 = 0.8, f_s = 0.6, b_s = 0.6$, progress chain 1 of length 10, progress chain 2 of length 3, disengagement chain of length 20. Note that we purposely made the disengagement chain long in order to better examine how far off each estimator gets as the number of timesteps increases. At each timestep of each trajectory, we estimated the disengagement state that that timestep using both the iterative estimator and the sequential estimator. We used this to find the mean difference and mean absolute error (MAE) between the estimated disengagement value and actual disengagement value at each timestep, and computed the standard deviation over these estimates at each timestep.

We found that the sequential estimator has significantly lower empirical bias and MAE than the iterative estimator. This is demonstrated in Figure 12. We can see that the mean difference in the error between the disengagement estimate and true disengagement state remains around 0 at all timesteps–indicating empirically that our estimator is unbiased. Meanwhile, once the estimator started diverging from the true disnegagement state, as the number of timesteps increased, it diverged even more, getting to the point where it estimated on average that the disengagement state was 5 states ahead on the disengagement chain than it actually was.

Furthermore, we see that the sequential estimator's mean absolute error (MAE) remains low as the number of timesteps increases, remaining within 1 of the true disengagement state by 20 timesteps, and remaining within 2 after 50 timesteps. This demonstrates that it is a good estimator that can reliably capture the true disengagement state. Therefore, we chose to proceed with the sequential estimator in the main paper.

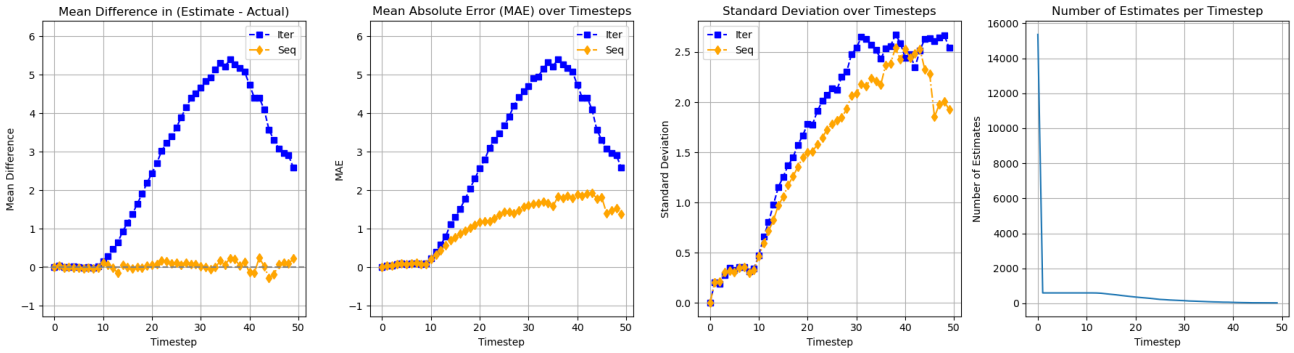


Figure 12. **Sequential estimator has significantly less error.** Disengagement estimators: iterative (labeled as Iter) and sequential (labeled as Seq). The x-axis denotes the k (timestep) for which we are estimating $s_k^{(D)}$. Note that by timestep 50, there are slightly above 30 estimates which we compute the mean difference (empirical bias), MAE, and standard deviation over. Parameters: $r_g = 10, r_d = -10, f_1 = 0.9, f_2 = 0.2, b_1 = 0.1, b_2 = 0.8, f_s = 0.6, b_s = 0.6$, progress chain 1 of length 10, progress chain 2 of length 3, disengagement chain of length 20).

E.5. Robustness of AI under Disengagement Estimation across all Policy Classes

Here, we show the robustness of the AI to both the sequential and the iterative estimator for all policy classes. Their parameters are as from C.1.

We can see from Figure 13 that for all policy classes, estimating the disengagement state reduces the AI agent's ability to prevent the human from dropping out. This is true of both the iterative and sequential estimators. We can also see that in the cases where the human agent does not readily reach the goal state without assistance from the AI, estimating the disengagement state also slightly reduces the AI agent's ability to help the human agent reach the goal state. This reduction is most significant when the human agent struggles most to reach the goal state (complete the treatment plan), as seen in the easy only border easy prioritization policy class subfigure. Additionally, we also see that the iterative estimator results in a greater reduction in the AI's ability to help the human agent reach the goal state than the sequential estimator, but it also tends to lead to the human dropping out marginally less than under the sequential estimator (with the exception of the easy prioritization policy class).

F. Model-free Learning Fails to Perform Well if the Human is a Multi-progress Chainworld Additional Details

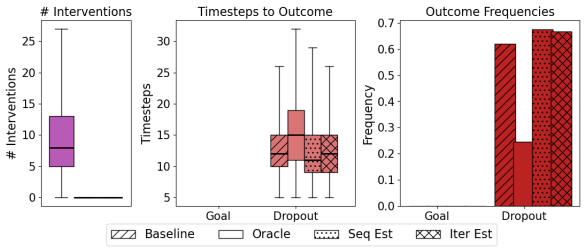
F.1. Set-up of Epsilon-Greedy Q-Learning

The AI agent uses the same parameter values as in Section 4, though it no longer has access to information about the human model; it can only observe the human agent's states and actions. Rather than solving for the optimal policy via value iteration, it now learns the policy via epsilon-greedy Q-learning. The learning rate $\alpha = 0.15$, which we obtained by fine-tuning the learning rate, as shown in F.2. The initial $\epsilon = 1$ and we use episodic ϵ-decay, multiplying it by $\kappa = 0.995$ each episode, down to a minimum of 0.1. All experiments are over 500 trials of 1000 episodes each, where each episode corresponds to a human trajectory of up to 80 timesteps.

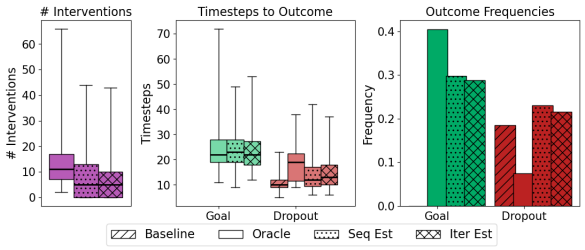
F.2. Fine-tuning the Learning Rate

We note that we have used a learning rate of $\alpha = 0.15$ for the main body of this paper since we found it to be the best learning rate for our setting. We fine-tuned it on the easy prioritization policy class (using the parameters as from C.1), and the results of doing so are in Figure 14.

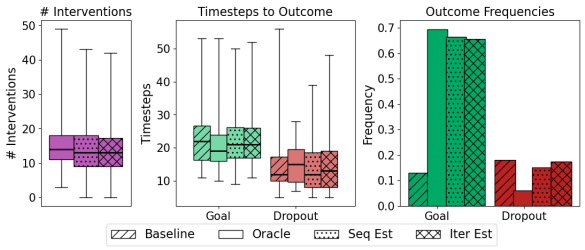
We did a log-based grid-search over learning rates in $(0, 1]$, and none of them resulted in quickly learning a policy which could perform on par with the oracle AI's policy. Recall that the oracle is the AI with full knowledge of the human model and can use value iteration to solve for its optimal intervention policy.



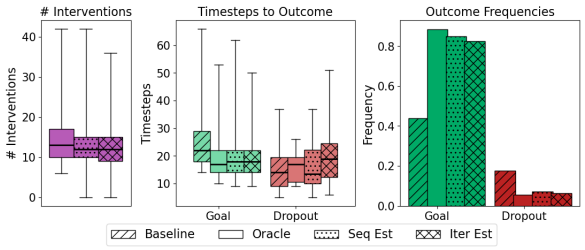
(a) Easy only policy class.



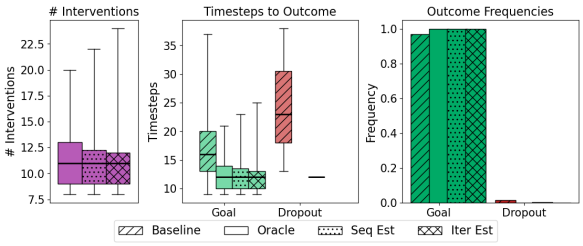
(b) Easy only border easy prioritization policy class.



(c) Easy prioritization policy class.



(d) Easy prioritization border hard policy class.



(e) Hard prioritization policy class.

Figure 13. **Impact of disengagement estimation is reducing AI's ability to prevent human agent dropping out.** AI with full knowledge (oracle) vs. disengagement estimation (seq est) by number of interventions (left), impact on how fast the human hits s_g and s_d as against its baseline (middle), and frequency of human dropping out vs. reaching goal (right).

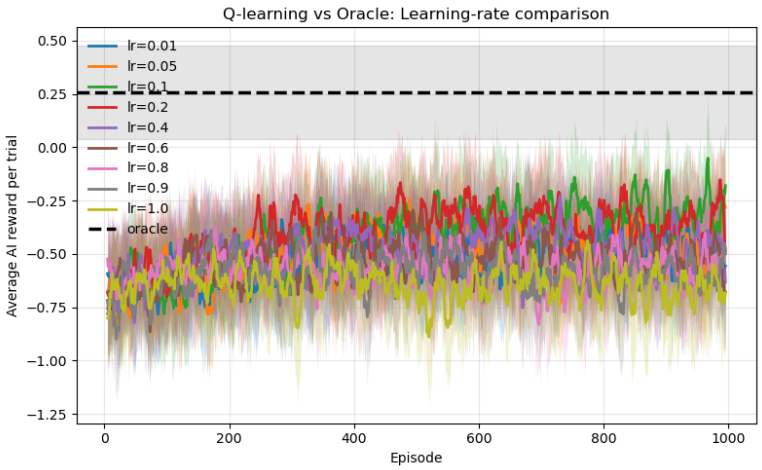
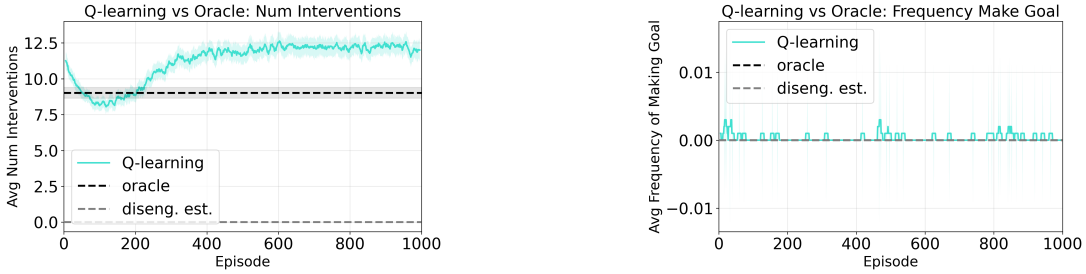


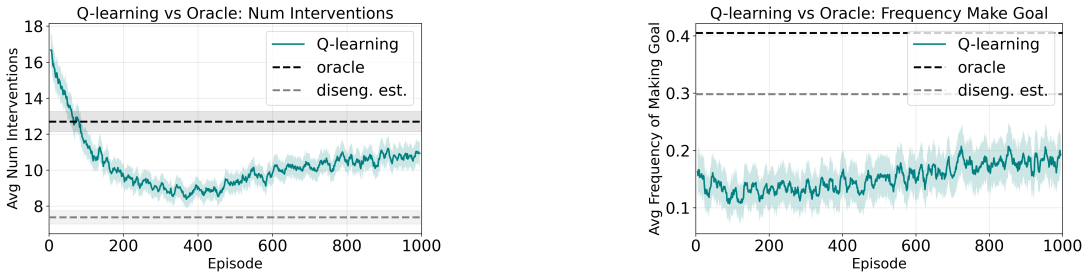
Figure 14. **Best learning rate is between 0.1 and 0.2.** Performance (in terms of average AI reward) of Q-learning as against the oracle AI across varying learning rates over 1000 episodes.

F.3. Results for all Policy Class Cases

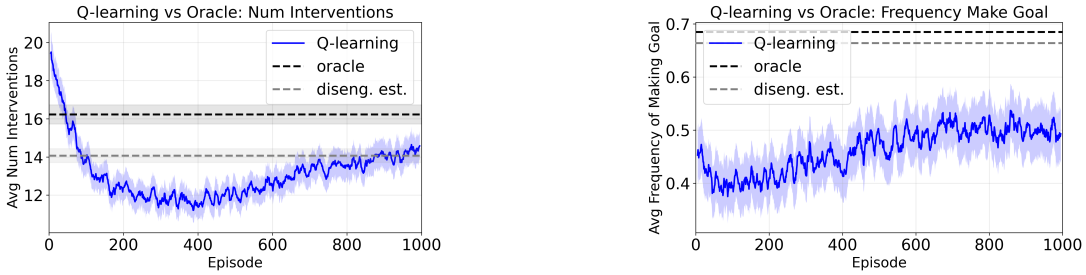
All human agents are instantiated with the default parameters from Section 3 and policy class-specific parameters from C.1. The oracle AI is instanstiated as from Section 6, noting that the AI under disengagement estimation is the same oracle AI which knows the human model, but which is provided estimates of the disengagement state provided by the sequential estimator rather than the true human agent's disengagement state. We show the results of Q-learning for all policy classes in Figure 15. We see that except for when the human agent can never reach the goal state (the easy only policy) or can very frequently reach it on its own (the hard prioritization policy), the AI whose policy is learned via Q-learning fails to be nearly as helpful in getting the human agent to the goal state. Even after 1000 episodes, not only is it worse than the oracle (as we would expect), but it is also worse than this oracle when it is estimating the disengagement state. It also appears to be struggling to learn the ideal number of interventions across all policy classes.



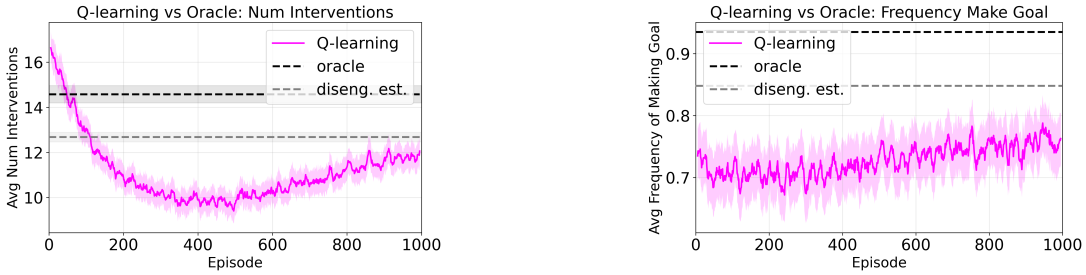
(a) Easy only policy class.



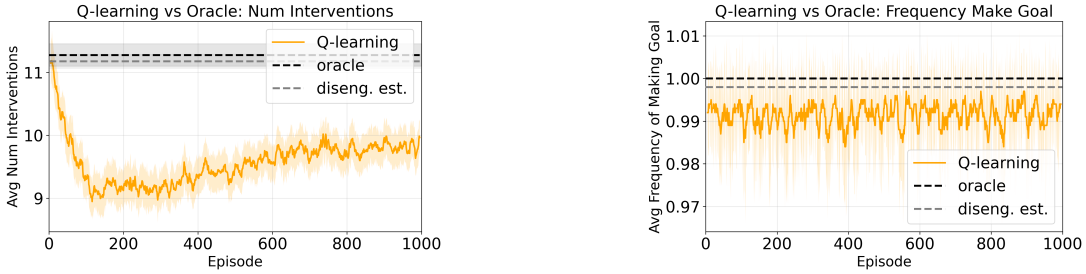
(b) Easy only border easy prioritization policy class.



(c) Easy prioritization policy class.



(d) Easy prioritization border hard prioritization policy class.



(e) Hard prioritization policy class.

Figure 15. **Q-learning fails to effectively personalize rapidly.** Average number of interventions (left) and frequency of human reaching goal state under intervention (right) using AI policy from Q-learning vs oracle and disengagement estimation for all policy classes.