Closed-loop Teaching via Demonstrations to Improve Policy Transparency (Supplementary Material)

Michael S. Lee, Reid Simmons, Henny Admoni
Robotics Institute, Carnegie Mellon University
{ml5, rsimmons, hadmoni}@andrew.cmu.edu

1 Particle Filter

We model a human's beliefs over an agent's reward function as a set of particles, defined by their positions and associated weights $\{\mathbf{x}_t, \check{\mathbf{w}}_t\}$. Each particle represents a possible reward function the human could believe the agent to have, and the associated particle weight captures the strength of that belief. Without loss of generality, assume that at each time step t, a demonstration is provided to the human or a test response is received from the human. Each demonstration or test response will produce one or more half-space constraints via Eq. 1 (see the main paper) that can be used to update the particle filter, where each constraint captures a characteristic of the reward function. For example, a demonstration or test response that takes two actions to detour around a mud patch rather than go through the mud patch will convey a constraint that going through mud must be at least twice as costly as an action.

1.1 Custom Distribution Parameter

We propose a custom probability distribution $p(x_t|y_t)$ for updating particle weights given a half-space constraint y_t. It is comprised of a uniform distribution that aligns with the consistent half-space of the constraint and a von Mises-Fisher distribution (a generalization of the Gaussian distribution on a sphere [Dhillon and Sra, 2003]) whose mean direction aligns with the inconsistent half-space. In addition to its mean direction, a von Mises-Fisher distribution is described by its concentration parameter κ, which, as the name implies, captures how concentrated the distribution is around its mean. In our experiments, we set κ to be 2, which we empirically observed as providing the right signal-to-noise ratio during the particle weight updates ($\kappa = 0$ corresponds to the uniform distribution and the distribution becomes more peaked around the mean, and less noisy, as κ increases).

1.2 Sampling Beliefs for Counterfactual Reasoning

We calculate the half-space constraints a demonstration could provide using counterfactual reasoning. Each constraint is calculated by comparing the agent's optimal behavior against alternative behaviors the human could counterfactually expect of the agent given their beliefs. We obtain these alternative behaviors by sampling a subset of the particles (which each correspond to a reward function) and rolling out the corresponding optimal behavior. As noted in the main paper, we rely on the 2-approximation algorithm [Hochbaum and Shmoys, 1985] to greedily select k distributed samples such that the maximum distance from any particle in the candidate set to one of the k samples is minimized (see Fig. 1).

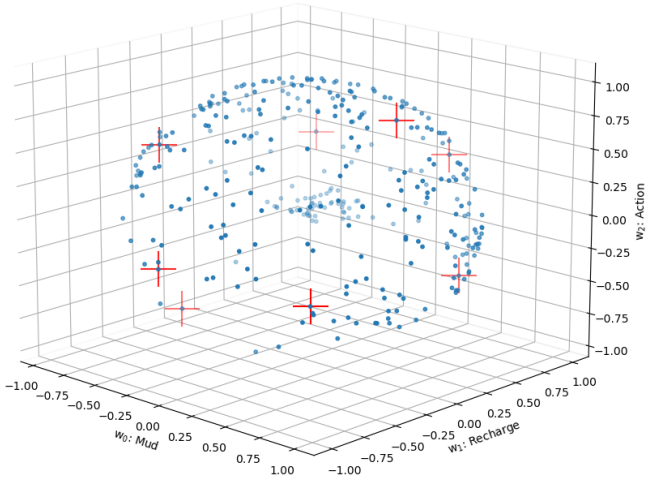

Figure 1: Human counterfactuals are generated by sampling beliefs from the particle filter model. As nearby particles are likely to generate similar counterfactuals, we rely on the 2-approximation algorithm for the k-center problem to sample k beliefs (marked by red crosses) that are spread out.

For our experiments, we set k to be 25. To support real-time counterfactual reasoning, we also sampled 2500 beliefs from the surface of the 2-sphere (the space of possible human beliefs regarding the agent's reward function in our domains) for which we pre-computed the optimal policies. Each particle in the particle filter was then mapped to the closest pre-computed belief during experiments toward efficient selection of additional demonstrations and tests.

1.3 Resampling and Resetting

We address common challenges to using particle filters in practice. Sample degeneracy occurs when successive updates to the weights of the particles cause only a few particles to have high weight and the particle filter fails to model regions of interest in the posterior with sufficient detail [Li *et al.*, 2014]. Furthermore, the number of particles (i.e. sample size) should adapt to the complexity of the distribution being mod-

eled. To address both concerns, we rely on KLD-resampling [Li *et al.*, 2013] to obtain the sample size that bounds the Kullback-Leibler (KL) divergence between the sample-based maximum likelihood estimate and the true posterior distribution, and simultaneously rely on systematic resampling to concentrate the sampling near regions of high probability. Finally, measures to combat sample degeneracy can actually cause sample impoverishment, where the particle filter is too concentrated and not amenable to future shifts in the posterior. Thus we resample only when the effective sample size (a measure of sample degeneracy) drops below a predefined threshold and also add Gaussian noise when resampling the particles [Li *et al.*, 2014]. This limited resampling balances the risk of running into sample degeneracy or sample impoverishment, which are at opposite extremes.

Furthermore, the particle filter may converge, then suddenly obtain new information that is heavily inconsistent with the current distribution. In this case, the filter will struggle to update, as none or very few of the particle weights would be increased to shift the distribution in a meaningful way. We thus implement particle filter resetting, taking inspiration from sensor resetting localization [Lenser and Veloso, 2000] that combats the kidnapped robot problem, where the robot has been moved without being told and must reinitialize its localization. A reset triggers when the weights of the particles, after accounting for $p(x_t|y_t)$ and before weight normalization, drop below a threshold. We uniformly distribute a set number of particles into the consistent half-space and again rely on KLD-resampling [Li *et al.*, 2013] to obtain the number of particles that will bound the KL divergence between the posterior distribution following the reset and its sample-based maximum likelihood estimate. We then sample that number of particles directly from the custom distribution corresponding to $p(x_t|y_t)$ and add it to the particle filter (see Fig. 2).

2 Closed-loop Teaching Framework

We characterize the proposed closed-loop teaching framework in pseudocode below (also described in the main paper in Section 4.2, Fig. 2, and Fig. 5).

Algorithm 1 Closed-loop Teaching Framework

```
Group related knowledge components (KC) into batches using counterfactual scaffolding [Lee et al., 2022]
for each batch of KCs (i.e. lesson) do
    Provide initial demonstrations and diagnostic tests
    Evaluate diagnostic test responses
    for each diagnostic test incorrect response do
        Provide corrective feedback, remedial demo, and a remedial test
        Evaluate remedial test response
        while remedial test response is incorrect do
            Provide corrective feedback and provide new remedial test
            Evaluate remedial test response
        end while
    end for
end for
```

3 User Study

We provide additional details on the user study that tests our four hypotheses on how the between-subjects variable of feedback loop (conditions: full closed-loop, partial closed-loop, open loop) affects correctness of test responses (H1) and ratings on focused attention and perceived usability (H2), improvement (H3), and understanding (H4). The within-subjects variable was domain (conditions: delivery, skateboard), and the order of the domains was counterbalanced.

H1: We considered analyzing test responses in two ways: binary scores measuring the optimality of human test responses, and regret measuring the degree of suboptimality of human test responses (i.e. the difference between rewards of human and optimal test responses). The former analysis was coarse and did not yield any significant results, so we opted for the latter which provides a finer resolution. And though one participant had only 11/12 test responses recorded, this does not significantly impact the results as responses were averaged for each participant and 2447 total responses were recorded.

H2: We ran a Cronbach's alpha to verify the reliability of the corresponding Likert scales for measuring focused attention and perceived usability. For focused attention, we observed that the value rose from $\alpha = 0.58$ to $\alpha = 0.65$ without the second item (which asked for a response to the question "The time I spent learning the game strategy passed by quickly." on a 5-point scale) and we remove this item from the analysis accordingly. For perceived usability, we keep all items for the analysis below as removing any of them did not increase the $\alpha = 0.86$ that was obtained using all items.

Questions from the User Engagement Scale short form [O'Brien *et al.*, 2018] were adapted to measure focused attention:

- "I was fully engaged with learning the game strategy."
- "The time I spent learning the game strategy passed by quickly."
- "I was absorbed in this experience."

and measure perceived usability:

- "I felt frustrated while learning the game strategy."
- "I found learning the game strategy confusing."
- "Learning the game strategy was taxing."

each answered with a 5-point Likert scale.

4 Limitations

In principle, the proposed methods in the paper can be used to teach RL policies whose reward functions take the form of a weighted linear combination of reward features.

Practically, demostrations and tests are selected from a set of possible grid world settings, which can be computationally expensive to enumerate depending on the size of the set. Furthermore, constraints on reward functions underlying demonstrations and test responses are generated using reward features counts μ, which extend naturally to continuous state and action spaces (see Eq. 1 of main paper). However, our method will still require discretization in continuous state and action spaces given an infinite set of possible demonstrations.

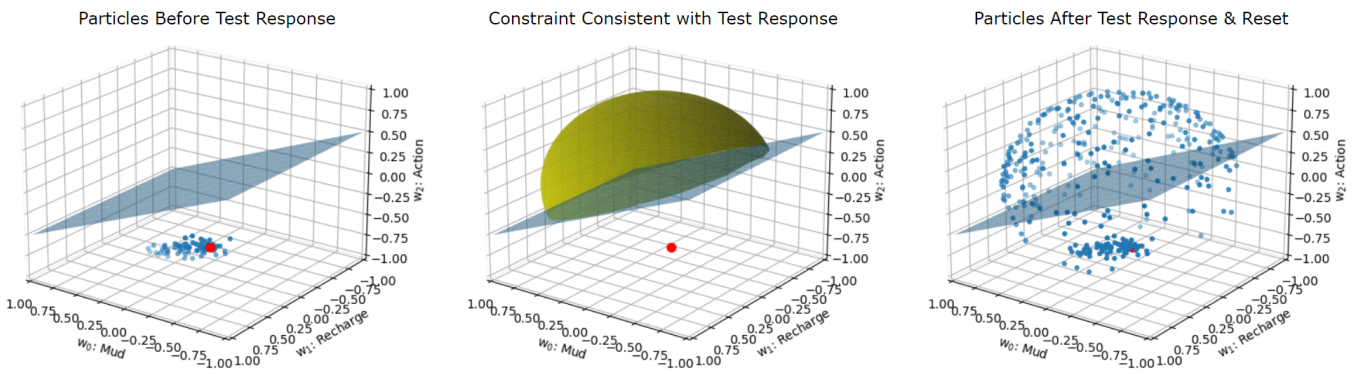

Figure 2: When a test response is heavily inconsistent with the current model of human beliefs, we perform a reset. The constraint consistent with the test response is shown in all panels, with the consistent side shown with the uniform distribution as a yellow dome in the center panel. The agent reward function is shown as a red dot.

References

[Dhillon and Sra, 2003] Inderjit S Dhillon and Suvrit Sra. Modeling data using directional distributions. Technical report, Citeseer, 2003.

[Hochbaum and Shmoys, 1985] Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 1985.

[Lee *et al.*, 2022] Michael S Lee, Henny Admoni, and Reid Simmons. Reasoning about counterfactuals to improve human inverse reinforcement learning. In *IROS*. IEEE, 2022.

[Lenser and Veloso, 2000] Scott Lenser and Manuela Veloso. Sensor resetting localization for poorly modelled mobile robots. In *International Conference on Robotics and Automation*, 2000.

[Li *et al.*, 2013] Tiancheng Li, Shudong Sun, and Tariq Pervez Sattar. Adapting sample size in particle filters through kld-resampling. *Electronics Letters*, 2013.

[Li *et al.*, 2014] Tiancheng Li, Shudong Sun, Tariq Pervez Sattar, and Juan Manuel Corchado. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with applications*, 2014.

[O'Brien *et al.*, 2018] Heather L O'Brien, Paul Cairns, and Mark Hall. A practical approach to measuring user engagement with the refined user engagement scale (ues) and new ues short form. *International Journal of Human-Computer Studies*, 2018.