We thank the reviewers for their thoughtful feedback and suggested improvements. We agree that our proposed e-stop framework is a simple and attractive mechanism for incorporating state-only observations from an expert demonstrator, or manual interventions from a supervisor. Here, we provide clarifications and additional results which address the reviewer concerns necessary to further increase their scores.

Additional empirical analysis The central feedback from all reviewers was the need for experiments in a continuous MDP, in addition to our previously presented tabular results. We present new results on an inverted pendulum environment with continuous states and actions which will be included in the manuscript. We trained an agent using deep deterministic policy gradient (DDPG), with an actor and critic using two and three hidden layers of size 64, respectively. We collected 500 demonstration trajectories from a near-optimal expert policy and constructed a support superset from a rectangular hull of these samples. Fig. 1 compares DDPG with and without an e-stop mechanism, and shows roughly an order of magnitude improvement in sample complexity on a hold-out set of random seeds.

We provide further fine-grained analysis by controlling for several other parameters in the grid-world environment. To address [R1], we present results in Fig. 2 over a range of expert demonstrations used to construct the support superset $\hat{\mathcal{S}}$. Even after decreasing the number of demonstrations from 1000 to 5, the cumulative reward of the agent trained via our e-stop mechanism decreased by less than 3%. Note that a bound relating the number of expert trajectories to the suboptimality of our method was proved in Theorem 5.1. To address [R3], in Fig. 3, we controlled for the quality of the demonstrator policy by injecting random noise into the optimal policy's Q-function. As predicted by Theorem 5.1, the learned e-stop policy has bounded regret with respect to the expert demonstrations, but in practice typically exceeded the performance of the expert.

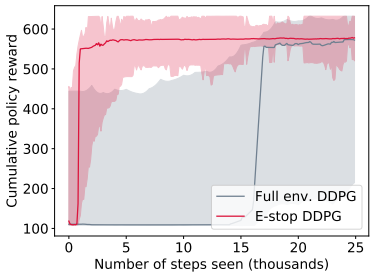

Figure 1: Pendulum environment.

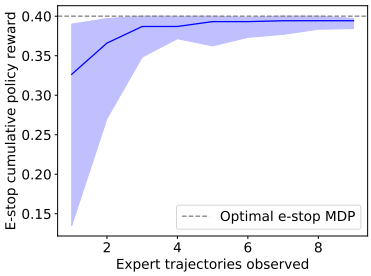

Figure 2: Number of observations.

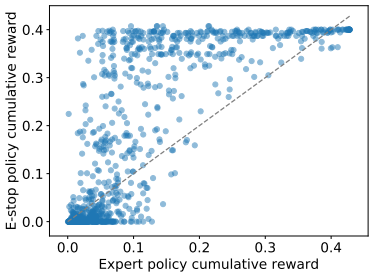

Figure 3: Expert quality.

Relationship to existing work We emphasize that our method applies to the reinforcement learning with expert observations (RLEO) setting, where only the states visited by the expert are observed. This precludes methods for the reinforcement learning with expert demonstrations (RLED) setting, where both the expert's states and actions are provided. To answer [R2], it is not immediately clear how DQfD (Hester et al.) could be extended to the RLEO setting, as their method requires action observations to perform Bellman updates. However, it is possible to extend our method to RLED by constructing a support superset $\hat{\mathcal{S}}$ based on state-action pairs. Thus, our method and many RLED methods are complimentary. For example, DQfD would allow pre-training the policy from the state-action demonstrations, whereas ours reduces exploration during the on-policy learning phase. Similarly, our work can be related to Fujimoto et al. (as suggested by [R1]) by using a state-action superset and off-policy training, where states outside of $\hat{\mathcal{S}}$ are never selected in the Bellman update rule due to the maximum terminal e-stop penalty.

To answer [R3], the method in Eisenbach et al. is also complimentary to our work, and does not permit a direct comparison. To elaborate, they consider the RL setting where a second "soft reset" policy is learned in addition to the standard hard reset. The soft reset policy prevents the agent from entering nearly non-reversible states and returns the agent to an initial state. Hard resets are required whenever the soft reset policy fails to terminate in a manually defined set of safe states $\mathcal{S}_{\text{reset}}$. Our method can be seen as learning $\mathcal{S}_{\text{reset}}$ from observation (in the absence of a soft reset policy, we also use $\mathcal{S}_{\text{reset}}$ during the forward policy roll-outs). Their method trades off hard resets for soft resets, whereas ours learns when to perform the hard resets.

"Shielding" (Alshiekh et al.), as suggested by [R2], uses a manually specified safety constraint and a course, conservative abstraction of the dynamics to prevent an agent from violating the safety constraint. Our expert observations can be seen as inducing a safe region (the support superset), although our e-stop mechanism is model-free and uses a terminal reset penalty for actions which would have exited the safe region.

Other minor improvements We will make all typographical and clarity changes suggested by the reviewers, including using a more informative title.

We believe these changes, and in particular the additional results in continuous MDPs, address all reviewer concerns.