MILE: Model-based Intervention Learning

Yigit Korkmaz, Erdem Bıyık

Abstract— **Imitation learning techniques have been shown to be highly effective in real-world control scenarios, such as robotics. However, these approaches not only suffer from compounding error issues but also require human experts to provide complete trajectories. Although there exist interactive methods where an expert oversees the robot and intervenes if needed, these extensions usually only utilize the data collected during intervention periods and ignore the feedback signal hidden in non-intervention timesteps. In this work, we create a model to formulate how the interventions occur in such cases, and show that it is possible to learn a policy with just a handful of expert interventions. Our key insight is that it is possible to get crucial information about the quality of the current state and the optimality of the chosen action from expert feedback, regardless of the presence or the absence of intervention. We evaluate our method on various discrete and continuous simulation environments, a real-world robotic manipulation task, as well as a human subject study. Videos and the code can be found at https://liralab.usc.edu/mile.**

I. Introduction

Imagine training a household robot to help users place the dishes in the dishwasher. One way to do this is to use reinforcement learning (RL) that has been proven successful in several areas ranging from gaming to dialogue systems to autonomous driving [1]–[4]. However, its need for lots of online interactions with the environment as well as a well-defined reward function make it unsuitable in a real-world situation like this. An alternative is to use imitation learning (IL) where an expert provides demonstrations of how to place the dishes. IL requires fewer interactions in the world than RL and does not require a reward function. A common drawback of this approach is the compounding distributional shift, which results from the accumulation of errors when deploying a learned policy [5]: the small inaccuracies in the robot's learned policy will move it to an out-of-distribution state where the policy may fail more significantly, which may result in the robot breaking the dishes.

Interactive learning techniques try to overcome this compounding errors problem by querying the expert with system states in an iterative fashion, and fine-tuning the policy based on the actions provided by the expert [6]–[9]. Most of these techniques do not let the human intervene at will but transfer the control to the human according to some criterion [8], [10]. In others, the human can take over the control at any timestep [7], [11]–[13].

Going back to the running example, assume the robot had a mediocre policy in the beginning, thanks to some initial training done in the factory setup. In this interactive scheme, we would control the robot only when we think it

Both authors are with Thomas Lord Department of Computer Science, University of Southern California. Emails: {ykorkmaz, biyik}@usc.edu

is doing or about to do something wrong. This is clearly more convenient than operating the robot for long periods to generate full demonstrations. But what about the intervals where we did *not* input any actions? Did we not provide any information to the robot? The answer is no: the fact that we *chose* to not intervene means the robot's actions were already good enough in those intervals. This is what the existing interactive learning techniques are missing: even though they try to improve the robot's policy based on the states where the expert intervenes, they do not utilize any structure of how or when those interventions occur, ignoring an important feedback signal that is leaking through the states where the expert does not intervene.

In order to efficiently use the information in both the states with and without human interventions, we argue that it is necessary to understand and utilize the structure behind how those interventions occur. Motivated by this, we propose the following contributions in this paper:

1) We propose a novel model that is fully differentiable to formulate how and when an expert intervenes.
2) We utilize this intervention model to fine-tune a weak policy and evaluate our method in various simulated and real-world environments to prove its effectiveness.
3) We compare our method against the state-of-the-art baselines which utilize interventions to show its higher sample-efficiency and performance.

II. Related Work

Interactive Imitation Learning. In imitation learning, expert data is used to train a policy in a supervised way [14]–[19]. Interactive imitation learning methods try to overcome its compounding errors problem [5], [6] by querying the expert on the learned policy's rollouts [6]–[8], [10], [14]. While the expert relabels either the whole trajectory of the learned policy with actions, or some parts of it that are automatically selected by estimating various quantities related to task performance [6], [10], there are also works where the expert has the freedom to intervene at will [7], [11], [12], [20]. Some of these methods consider the implicit feedback coming from the states that the expert chooses not to intervene. They attempt to incorporate the information leaking from non-intervention intervals either by enforcing those state-action pairs to be constrained in the action-value cost functions using some hyperparameters [13] or utilize weighted behavioral cloning (BC) to incorporate the signal from non-interventions, with different heuristics used for assigning weights to on-policy robot samples, and expert human interventions [11], [12], [20]. However, none of these algorithms uses a model to understand and learn from why
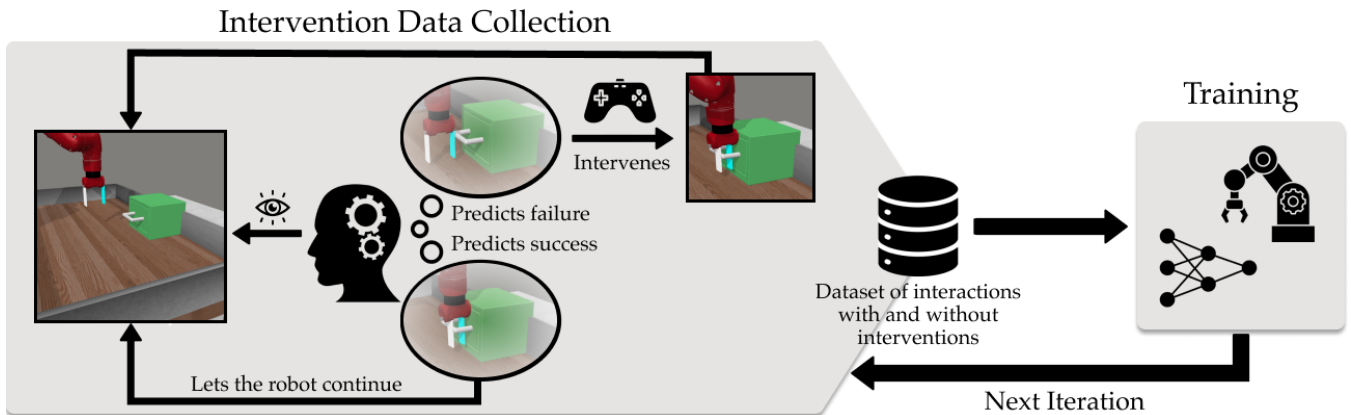
Fig. 1. **Overall MILE System.** A human operator oversees the robot during task execution and may decide to take over control at any timestep. The human makes the decision to intervene or not based on their prediction of the robot's potential failure, without observing the robot's action in that particular state. During the data collection phase, all interactions with the environment are recorded, both with and without interventions. The policy is then trained on this dataset using an iterative process that incorporates our novel intervention model.

the expert chooses to not intervene. In our method, we propose a model that attempts to capture how the interventions occur, and how satisfied the user is with the current action in the case of not intervening. Moreover, we investigate how we can efficiently use the information coming from that model for training. To the best of our knowledge, this is the first work that models how human interventions happen in robotics while utilizing it for better policy learning.

Reinforcement Learning with Expert Data. Many studies focus on integrating expert demonstrations into the RL pipeline. This is often achieved by populating the replay buffer to warm-start the initial policy and guide its exploration in a favorable direction [21]–[24]. There are also approaches that incorporate interventions into RL. One approach introduces an extra cost term in the learning objective to minimize interventions [25]. Another recent method relabels the rewards based on the interventions, assigning negative rewards at intervention states, while giving a reward of zero to all other states [26]. However, most of these methods assume access to some type of reward information, either through explicit reward functions or through Q-functions of the task. Our method does not require any of these assumptions. Not to mention, these methods still require a lot of online interactions with the environment and RL algorithms that can handle both on-policy and off-policy data coming from the intervening agent.

Modeling Interventions. There have been efforts to provide formal definitions and qualitative analyses of interventions in the human-robot interaction research, along with developing metrics to evaluate operator workload [27], [28]. Within computational modeling, various studies have addressed the modeling of trust in human-robot collaborative settings, which include both binary and continuous measures [29]. There are also methods exploring the temporal dynamics of trust conditioned on the task performance [30], [31]. One such work models trust as a latent variable in a partially observable Markov decision process (POMDP), utilizing this model to maximize task performance [31]. To the best of our knowledge, no prior work has developed a similar computational model to explain the occurrence of

interventions in human-robot teams.

III. PROBLEM DEFINITION

We formulate the problem as a discrete time Markov decision process (MDP) with the standard $\langle S, A, \rho, f, R, T, \gamma \rangle$ notation. The robot does not know the reward function R or the transition dynamics f of the environment. It starts with an initial policy $\pi_\theta : S \to A$ parameterized by θ.

In addition, a potentially noisy expert human can intervene (take over the control) with any action $a_h \in A$ at any timestep. However, in contrast to some of the existing works [32], the human has to make the decision about intervening or not without observing the robot's action $a_r \in A$ in that particular state. This is a more viable setting in robotics where it is not realistic to expect the robot to check each and every action with the human.

We let the robot and the human interact in this setup, and we record (s, a_r, a_h, s') from every timestep, with the possibility that a_h is undefined if the human did not intervene in that timestep. Our objective is to find a policy π_{θ^*} that maximizes the expected cumulative discounted reward:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t) \right]$$

where $s_0 \sim \rho(\cdot)$, and (s_t, a_t) pairs are sampled according to the policy π_θ and transition function f.

We assume the human interacts with the robot for N iterations, each of which includes k episodes. After each iteration, we train the robot policy using the data collected so far. In the next section, we describe how we do this training.

IV. MODEL-BASED INTERVENTION LEARNING (MILE)

Our method to solve this problem is based on a computational model of when and how the human may decide to intervene the robot's operation. We will introduce this intervention model in the first subsection. Subsequently, we will leverage this model to create a framework where we update the robot's policy π_θ based on its interactions with the human and the environment.

Intervention Model. Consider our running example: would you intervene the robot's operation when it is quickly
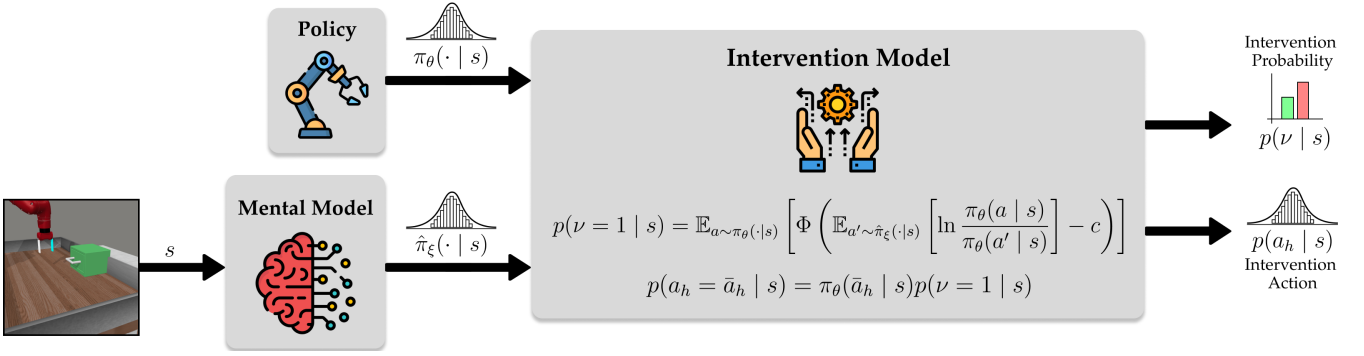
Fig. 2. **Framework for learning from interventions.** Starting from an inital policy π_θ, we jointly train the mental model, and the policy using our intervention model.

moving toward a glass jar that stands at the edge of the countertop? How about if the jar was made of plastic? This example highlights the important factors that affect when humans intervene the robot. Firstly, consider the case of glass jar. The human is likely to intervene because they predict the robot may hit and break the jar. So the human's belief about what the robot will do, i.e., their mental model of the robot, affects their decision to intervene, as this was previously shown in human-robot interaction research [31].

Next, let's compare the cases with glass and plastic jars. The human is less likely to intervene in the latter, because even if the robot takes bad actions, the outcome is not going to be as bad as in the case of glass jar. This points out that the human's interventions also depend on how good/bad the potential outcomes are.

Considering these factors, we propose an intervention model based on the probit model from discrete decision theory [33]. For the ease of the presentation, we will start with discrete action spaces and then extend our formulation to continuous domains.

Let ν be a binary random variable that indicates whether the human intervenes ($\nu = 1$) or not ($\nu = 0$), and $\bar{a}_h$ denote the nominal human action, i.e., the action human would take provided they decide to intervene. Mathematically, $a_h = \bar{a}_h$ if and only if $\nu = 1$. Otherwise, a_h is not defined in that state. Finally, let $\hat{\pi}$ denote the human's mental model of the robot, i.e., what the human believes the robot will do in a given state. This prediction is needed as the human has to intervene before seeing the robot's action. While some learning from intervention works assumes the human first sees the system's proposed action [32], ours is a more realistic setting, especially in robotics where conveying an action without taking it is both difficult and time-consuming.

We start with modeling the probability that the human will intervene at a given state s. We use σ for the softmax operation and Φ for the cdf of a standard normal distribution.

$$p(\nu = 1 \mid s) = \sum_{a \in A} p(\bar{a}_h = a, \nu = 1 \mid s)$$
$$= \sum_{a \in A} p(\bar{a}_h = a \mid s) p(\nu = 1 \mid \bar{a}_h = a, s) \quad (1)$$

Here, the first term inside the summation is just the probability that the human would take action a at state s. Since we assume the human is a (noisy) expert, we use a Boltzmann

policy under the true reward function of the environment to model this probability, similar to the prior work [34]–[38]:

$$p(\bar{a}_h = a \mid s) = \pi_h(a \mid s) := \sigma(Q(s,a)) = \frac{\exp(Q(s,a))}{\sum_{a' \in A} \exp(Q(s,a'))}$$

The second term in Eq. (1) denotes the probability that the human will intervene conditioned on the state and their nominal action. This is where we bring the probit model into play: the human will intervene only if their nominal action is considerably better than what they expect the robot to do:

$$p(\nu = 1 \mid \bar{a}_h = a, s) = \Phi\Big(Q(s,a) - \mathbb{E}_{a' \sim \hat{\pi}(\cdot \mid s)}[Q(s,a')] - c\Big),$$

where c is a scalar hyperparameter. In this probit-based model, the human is more likely to intervene if the value of their nominal action is much higher than the expected value of the robot's action. The scalar c depends on the effort the human needs to put to intervene the robot. If it is difficult to intervene, c is going to be high and the human will only intervene if the difference in action values is extremely high. On the other extreme, if interventions are free, then c will be low and the human may continually intervene.

Now that we modeled the probability of human interventions, we continue with *how* they intervene when they do:

$$p(a_h = \bar{a}_h \mid s) = p(a_h = \bar{a}_h \mid \nu = 0, s)p(\nu = 0 \mid s)$$
$$+ p(a_h = \bar{a}_h \mid \nu = 1, s)p(\nu = 1 \mid s)$$
$$= \sigma(Q(s, \bar{a}_h))p(\nu = 1 \mid s),$$

where $p(a_h = \bar{a}_h \mid \nu = 0, s) = 0$ by the definition of a_h.

We note the following relation due to the Boltzmann policy formulation we use for the human:

$$\ln \pi_h(a \mid s) - \ln \pi_h(a' \mid s) = \ln\left(\frac{\exp Q(s,a)}{\sum_{a'' \in A} \exp Q(s,a'')}\right)$$
$$- \ln\left(\frac{\exp Q(s,a')}{\sum_{a'' \in A} \exp Q(s,a'')}\right)$$
$$= Q(s,a) - Q(s,a').$$

This allows us to rewrite $p(\nu = 1 \mid s)$ and $p(a_h = \bar{a}_h \mid s)$ only dependent on the policy and not the Q-function:

$$p(\nu = 1 \mid s) \qquad\qquad\qquad (2)$$
$$= \sum_{a \in A} \pi_h(a \mid s)\Phi\left(\mathbb{E}_{a' \sim \hat{\pi}(\cdot \mid s)}[\ln \pi_h(a \mid s) - \ln \pi_h(a' \mid s)] - c\right)$$
$$= \mathbb{E}_{a \sim \pi_h(\cdot \mid s)}\left[\Phi\left(\mathbb{E}_{a' \sim \hat{\pi}(\cdot \mid s)}[\ln \pi_h(a \mid s) - \ln \pi_h(a' \mid s)] - c\right)\right]$$
$$p(a_h = \bar{a}_h \mid s) = \pi_h(\bar{a}_h \mid s)p(\nu = 1 \mid s) \qquad (3)$$

This change of variables allows us to readily use this intervention model in continuous domains. The main difference is that, in the discrete case, a_h has a categorical distribution over the action space as well as the no-intervention ($\nu = 0$). In continuous case, ν acts like a gate controlling which continuous policy (π_h or π_θ) will be active. We will now present how we use this intervention model to learn from humans' intervention feedback.

Learning from Interventions. The robot misses information about two critical components in the intervention model: $\hat{\pi}$ that models what the human thinks the robot will do, and the human's policy π_h. In our learning algorithm, we model both of these policies with neural networks, $\hat{\pi}_\xi$ and π_θ, respectively (see "mental model" and "policy" in Figure 2). Since the intervention model is differentiable, we conveniently utilize the gradients coming from it to jointly train these networks using the dataset of (s, a_r, a_h, s') tuples. During inference time, we only employ the trained policy π_θ.

In discrete domains, we minimize the cross entropy loss between the ground truth actions (a_h if $\nu = 1$ and a_r otherwise) in the dataset and estimated final action probabilities, where no-intervention ($\nu = 0$) is an extra class in the action space. Mathematically, the loss function is:

$$J(\theta, \xi) = \text{CE}(\hat{\mathbf{P}}(\mathbf{S}; \theta, \xi), \mathbf{a_h}) = -\frac{1}{N} \sum_{i=1}^{N} \log(\hat{p}_{i,a_h^i}(\mathbf{s}_i; \theta, \xi)),$$

where $\mathbf{S}$ is the set of states in the dataset, $\hat{\mathbf{P}}(\mathbf{S}; \theta, \xi)$ is the matrix of probabilities with each row $\hat{\mathbf{p}}_i(\mathbf{s}_i; \theta, \xi)$ being the predicted probability vector for sample i, $\mathbf{a_h}$ is the vector of true intervention action labels including no-intervention, and $\hat{p}_{i,a_h^i}(\mathbf{s}_i; \theta, \xi)$ is the predicted probability for the true class of sample i.

In continuous domains, we use a combination of discrete and continuous loss functions as we want our model to learn both when and how the human intervenes. In order to accomplish the former, we use binary cross entropy loss between ground truth intervention signals ν and the estimated probability of intervention from Eq. (2), where π_θ replaces π_h and $\hat{\pi}_\xi$ replaces $\hat{\pi}$:

$$J_1(\theta, \xi) = \text{BCE}(\hat{\boldsymbol{\nu}}(\mathbf{S}; \theta, \xi), \boldsymbol{\nu})$$
$$= -\frac{1}{N} \sum_{i=1}^{N} [\nu_i \log(\hat{\nu}(\mathbf{s}_i; \theta, \xi)) + (1 - \nu_i) \log(1 - \hat{\nu}(\mathbf{s}_i; \theta, \xi))],$$

where $\hat{\boldsymbol{\nu}}(\mathbf{S}; \theta, \xi)$ is the predicted probability of whether an intervention will happen. For the latter, we minimize the negative log-likelihood of the human actions a_h (when $\nu = 1$) under the estimated action distribution in that state:

$$J_2(\theta) = \text{NLL}(\pi_\theta(\mathbf{a_h} \mid \mathbf{S})) = -\frac{1}{N} \sum_{i=1}^{N} \log(\pi_\theta(a_h^i \mid \mathbf{s}_i)) \quad (4)$$

J_1 (intervention loss) updates both the policy and the mental model, while J_2 (policy loss) updates only the policy π_θ, both terms contributing to policy learning:

$$J(\theta, \xi) = \lambda J_1(\theta, \xi) + (1 - \lambda) J_2(\theta) \quad (5)$$

In our experiments, we set $\lambda = 0.5$. We show the complete pipeline in Figure 2. The pseudocode for our framework can be found on the project website.

V. SIMULATION EXPERIMENTS

Experiment Setup. We tested our method on four different simulation tasks, one with a discrete and three with continuous action spaces. The discrete action space environment is Lunarlander from Gymnasium [39]. The remaining three tasks—Drawer-Open, Peg-Insertion, and Button-Press—are from the Metaworld suite [40] where the primary goal is to control a 6 DoF robot arm with a gripper to accomplish various tasks that require precise joint control and are prone to distributional shifts and local minima issues during RL optimization. For the Metaworld experiments, we use true world states as observations. To retain temporal information, we concatenate the states of the previous three timesteps with the current timestep. The action space is composed of 4 DoF: the Cartesian positions of the end effector, which always points down, and +1 dimension for the gripper. Since our model does not rely on any assumptions about the environment's reward structure, and our primary comparisons are with other interactive learning methods, we follow the prior work [41] to evaluate the methods using success rates. Lunarlander is the only exception where we used the environment rewards due to the simplicity of the task.

We use simulated humans to mimic human interventions in our simulation experiments. The goal of these experiments is to scale up the tasks and to show that it is possible to fine-tune an initial policy in a sample-efficient way using our framework. The data collection phase involves two distinct policies: a suboptimal agent for generating rollouts and an expert agent for interventions guided by our intervention model. In place of the simulated human's mental model of the robot (π_ζ), we train a BC policy on the rollouts of the suboptimal policy. During data collection, we rollout the suboptimal policy and intervene with the simulated human. In other words, the policy π_h of the human in the intervention model is replaced by π^* during the data collection using simulated humans. We see that these simulated interventions increase the success rate by comparing the dashed and the dotted lines in Fig. 3. During training, we jointly train the suboptimal agent and the mental model as we described in Section IV. At test time, we only use the trained policy and discard the mental model.

Baselines. We compare our methods against (i) BC intervention where the model estimates human's interventions with only a neural network, (ii) HG-DAgger [7], a state-of-the-art algorithm that iteratively finetunes the policy based on the interventions, (iii) RLIF [26], a recent work which models the interactive imitation learning as an RL problem and uses interventions as the reward signals to train the policy in an online fashion, (iv) IWR [12] and (v) Sirius [11], two methods utilizing weighted BC with different heuristics to learn from both intervention and non-intervention timesteps. We start with the same initial suboptimal policy for all methods, and assume there is no access to any expert dataset, apart from the interventions provided.

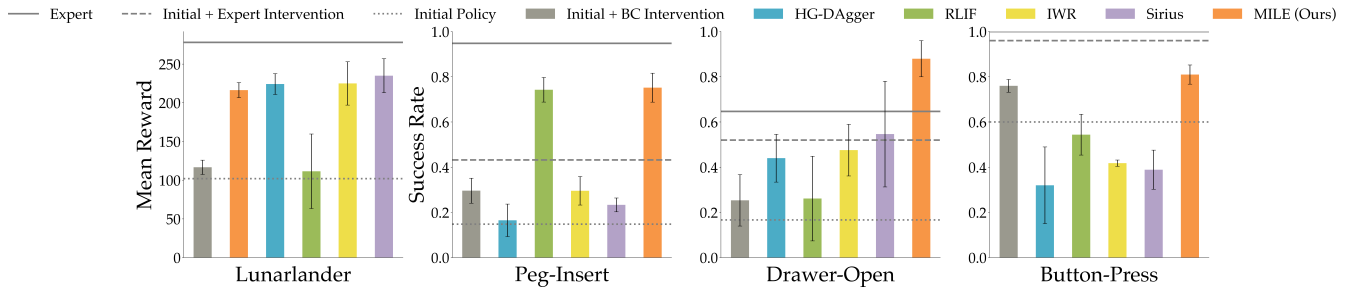To be consistent with the realistic settings, we keep the

Fig. 3. **Success rates for single iteration training (mean±std).** MILE was trained with $N = 1$ iteration and $k = 15$ episodes. To make it a fair comparison, we train the ohter baselines until they have the same number of interventions in their datasets as ours.

intervention ratio in the rollouts to be less than 30% by tuning the hyperparameter c. For our initial results, we train our method with $N = 1$ iteration and $k = 15$ episodes. For fairness, we train the other baselines until they have the same number of interventions in their datasets as ours.

Results. We present the results in Fig. 3. Our method achieves the best results across all environments (only tied with HG-DAgger, IWR and Sirius in Lunarlander, and with RLIF in Peg-Insert), proving its sample-efficiency. Due to the small amount of data which mostly consist of states with no interventions, IL methods suffer from overfitting as well as compounding errors. On the other hand, RLIF fails to produce reliable results, possibly due to the number of samples being too few to learn a successful policy.

To align more with the online nature of the baselines, we also perform iterative training with $N = 20$ iterations and $k = 1$ in Drawer-Open and Peg-Insert environments, as Button-Press was a relatively easier task. We run HG-DAgger, IWR, and Sirius in the same setting. For RLIF, we set the number of samples in its replay buffer approximately the same as MILE and other methods. The results are shown in Fig. 4. We observe that our method significantly outperforms the baselines. In just 10 iterations, it is able to achieve the success rate of the expert policy which provided the interventions, following with an even greater performance while the baselines mostly fall short.

Offline Demonstration Ablation. All of the baselines we compare against normally assume access to a set of offline demonstrations. Although MILE does not require such access, we run an ablation study in Drawer-Open, comparing
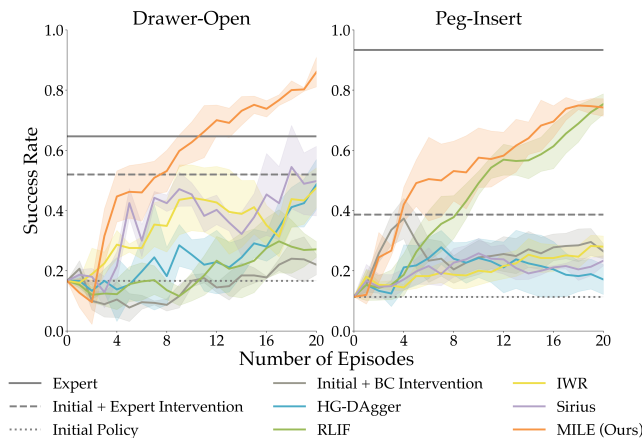

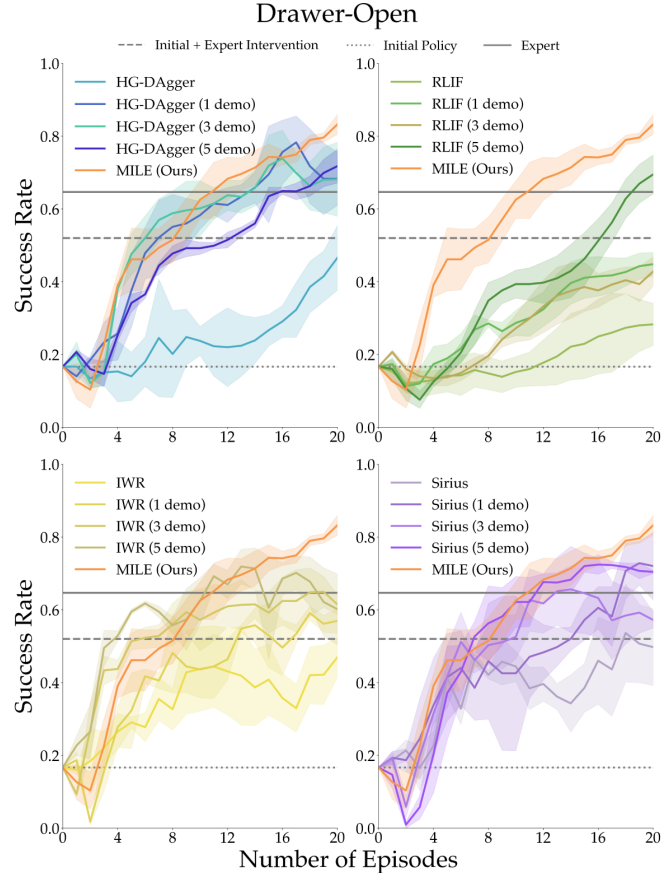Fig. 4. **Success rates for iterative training (mean±s.e.).**

Fig. 5. **Success rates for the demo ablation study (mean±s.e.).**

our method against the baselines when they have access to expert demonstrations. Results are shown in Fig. 5. Our method performs better than RLIF even when it has access to 5 expert demonstrations. IL-based methods' performance increases with the number of demonstrations in the dataset and becomes competitive with MILE but plateaus around the success rate of the expert agent. We believe this is due to overfitting and compounding errors. In both IWR and Sirius, robot actions are used as direct supervision signals in cases of non-intervention, albeit with different weights. In contrast, MILE uses the computational model for human interventions that we developed to construct probability distributions over the action space which are then used to train the models, thereby increasing the information bandwidth of the human feedback by computationally modeling them.

Overall, simulation results show our model for interventions improves learning, and increases adaptability, especially
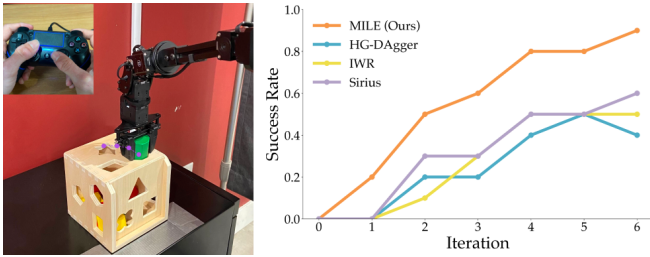
Fig. 6. **Real robot experiment setup and Success rates for MILE and other IL-based methods.** All methods were trained for 6 iterations with collecting 3 trajectories each iteration.

in low data regime where other interactive learning methods fail. With a handful of intervention trajectories, we are able to achieve near-optimal success levels in various tasks.

VI. REAL ROBOT EXPERIMENT

In order to show the effectiveness of our method in real world settings, we designed an experiment using a WidowX 6-DoF robot arm and with a real human. The task is to put the octagonal block into the wooden box through the correct hole as shown in Fig. 6. The sensitivity of this task is very high, as there exists a very small margin between the block and the hole, which makes it a suitable candidate when the small interventions provided by the humans can have substantial effect on the robot's success. We used image observations along with the robot's end effector position.

The robot is initialized with a mediocre policy that is unable to complete the task. During the execution, the human can take over the controls of the robot at any time via a PS3 controller. Similarly, they can also give the controls back to the robot. We run the experiment with $N = 6$ iterations, with $k = 3$ intervention trajectories collected in each iteration.

We compare our method against IL-based methods, which have been somewhat more successful than RLIF in the simulation studies. All methods use the same data collection interface. Similar to the simulation experiments, we do not assume access to any offline expert demonstrations and only use the intervention trajectories. The results are shown in Fig. 6. Our method is able to achieve 80% success rate just after 4 iterations, while other baselines struggle to improve the policy. This result showcases that our method is applicable to real world settings as well.

VII. HUMAN SUBJECT STUDY ON REAL ROBOT

Finally, we conducted a user study to analyze how accurately our model estimates the humans' interventions, as the success of our method relies on the success of the intervention model capturing when and how humans intervene the robot. The study is approved by the IRB office of the University of Southern California. We used the same task setting and experiment setup as in the previous section.

Procedure. We recruited 10 participants (3 female, 7 male, ages 20-28). The study started with a training period where subjects had a chance to get familiar with the robot controls.

We had 2 experiment settings. At the beginning of each experiment, the subject was first shown 3 trajectories from the initial policy to make them aware of the robot's capabilities. In the first experiment, the user was asked to collect 5

intervention trajectories ($N = 1$, $k = 5$), where they can take over the initial policy at any time. These trajectories were then used to fine-tune the initial policy using our method, and the final policy was shown to the user. We asked the users to rate the initial and the final policies.

The second experiment differed from the first by only using $N = 5$ and $k = 2$. The users could intervene and take control at any time during data collection: we did not provide any instructions on how often to intervene. We showed the finetuned policy at the end of each iteration and asked the user to rate its performance. At the end of the study, we did a post-study survey to collect participants' feedback on the method's effectiveness, improvement and the their satisfaction on the final result.

Results. We compared our intervention model against two other methods: one that always predicts no intervention (Majority Estimator) and a neural network classifier trained on data from previous iterations to estimate interventions in the current iteration (NN-Based Estimator). As shown in Fig. 7, our model outperforms these methods in predicting when an intervention will occur. Additionally, most users rated our model's effectiveness and adaptation speed as adequate, demonstrating that it successfully mirrors real human interventions.
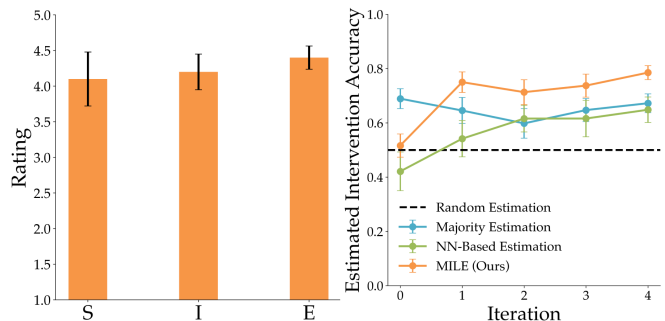

Fig. 7. **User Study Results (mean±s.e.).** Legend: S=Satisfaction, I=Improvement, E=Effectiveness

VIII. CONCLUSION

Summary. In this work, we presented MILE, a novel way of integrating interventions in policy learning. We developed a fully-differentiable intervention model to estimate when and how the interventions occur, and a policy training framework that uses the intervention model.

Limitations and Future Work. In its current form, our intervention model considers only the immediate state for estimating interventions. This does not fully capture the temporal analysis humans make before intervening, i.e., waiting for robot to fail in a recoverable way before taking control. Future work should explore ways of integrating this temporal aspect of intervention estimation, possibly with maintaining a belief over robot actions and using Bayesian updates. Some recent works show that the interventions can be used for data generation and allow better sim-to-real generalization [42], [43]. Another possible direction can be testing our method's capabilities in these domains, exploring whether using a structure behind interventions enables even further improvement in generalization or not.

REFERENCES

[1] D. Kalashnikov, A. Irpan, P. Pastor, *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*, PMLR, 2018, pp. 651–673.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[3] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[4] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[5] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, Eds., ser. Proceedings of Machine Learning Research, vol. 9, PMLR, 2010, pp. 661–668. [Online]. Available: https://proceedings.mlr.press/v9/ross10a.html.

[6] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudík, Eds., ser. Proceedings of Machine Learning Research, vol. 15, PMLR, 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html.

[7] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8077–8083. DOI: 10.1109/ICRA.2019.8793698.

[8] K. Menda, K. Driggs-Campbell, and M. J. Kochenderfer, "Ensembledagger: A bayesian approach to safe imitation learning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5041–5048. DOI: 10.1109/IROS40897.2019.8968287.

[9] R. Hoque, L. Y. Chen, S. Sharma, *et al.*, "Fleet-dagger: Interactive robot fleet learning with scalable human supervision," in *Conference on Robot Learning*, PMLR, 2023, pp. 368–380.

[10] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg, "Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning," in *Conference on Robot Learning*, PMLR, 2022, pp. 598–608.

[11] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu, "Robot learning on the job: Human-in-the-loop autonomy and learning during deployment," in *Robotics: Science and Systems (RSS)*, 2023.

[12] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese, *Human-in-the-loop imitation learning using remote teleoperation*, 2020. arXiv: 2012.06733 [cs.RO].

[13] J. Spencer, S. Choudhury, M. Barnes, *et al.*, "Learning from interventions: Human-robot interaction as both explicit and implicit feedback," in *16th Robotics: Science and Systems, RSS 2020*, MIT Press Journals, 2020.

[14] C. Celemin, R. Pérez-Dattari, E. Chisari, *et al.*, "Interactive imitation learning in robotics: A survey," *Foundations and Trends® in Robotics*, vol. 10, no. 1-2, pp. 1–197, 2022.

[15] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2008.10.024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008001772.

[16] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf.

[17] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg, "Dart: Noise injection for robust imitation learning," in *Conference on robot learning*, PMLR, 2017, pp. 143–156.

[18] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010. DOI: 10.1177/0278364910371999. [Online]. Available: https://doi.org/10.1177/0278364910371999.

[19] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1, Morgan-Kaufmann, 1988. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf.

[20] E. Chisari, T. Welschehold, J. Boedecker, W. Burgard, and A. Valada, "Correct me if i am wrong: Interactive learning for robotic manipulation," *arXiv preprint arXiv:2110.03316*, 2021.

[21] T. Hester, M. Vecerik, O. Pietquin, *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[22] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine, "Efficient online reinforcement learning with offline data," in *International Conference on Machine Learning*, PMLR, 2023, pp. 1577–1594.

[23] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 6292–6299.

[24] T. Xie, N. Jiang, H. Wang, C. Xiong, and Y. Bai, "Policy finetuning: Bridging sample-efficient offline and online reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 27 395–27 407, 2021.

[25] Q. Li, Z. Peng, and B. Zhou, "Efficient learning of safe driving policy via human-ai copilot optimization," *arXiv preprint arXiv:2202.10341*, 2022.

[26] J. Luo, P. Dong, Y. Zhai, Y. Ma, and S. Levine, "Rlif: Interactive imitation learning as reinforcement learning," in *The Twelfth International Conference on Learning Representations*, 2023.

[27] J. Scholtz, "Theory and evaluation of human robot interactions," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, IEEE, 2003, 10–pp.

[28] A. Steinfeld, T. Fong, D. Kaber, *et al.*, "Common metrics for human-robot interaction," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, 2006, pp. 33–40.

[29] L. Rodriguez Rodriguez, C. E. Bustamante Orellana, E. K. Chiou, L. Huang, N. Cooke, and Y. Kang, "A review of mathematical models of human trust in automation," *Frontiers in Neuroergonomics*, vol. 4, 2023, ISSN: 2673-6195. DOI: 10.3389/fnrgo.2023.1171403.

[30] Y. Guo and X. J. Yang, "Modeling and predicting trust dynamics in human–robot teaming: A bayesian inference approach," *International Journal of Social Robotics*, vol. 13, no. 8, pp. 1899–1909, 2021.

[31] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction*, 2018, pp. 307–315.

[32] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, "Trial without error: Towards safe reinforcement learning via human intervention," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 2067–2069.

[33] C. I. Bliss, "The method of probits," *Science*, vol. 79, no. 2037, pp. 38–39, 1934. DOI: 10.1126/science.79.2037.38. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.79.2037.38.

[34] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013. DOI: 10.1177/0278364913490324.

[35] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *Robotics science and systems: online proceedings*, vol. 2015, 2015.

[36] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh, "When humans aren't optimal: Robots that collaborate with risk-aware humans," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 43–52. DOI: 10.1145/3319502.3374832.

[37] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, 2008, pp. 1433–1438, ISBN: 9781577353683.

[38] C. Laidlaw and A. Dragan, "The boltzmann policy distribution: Accounting for systematic suboptimality in human models," in *ICLR*, 2022.

[39] M. Towers, J. K. Terry, A. Kwiatkowski, *et al.*, *Gymnasium*, Mar. 2023. DOI: `10.5281/zenodo.8127026`. [Online]. Available: `https://zenodo.org/record/8127025` (visited on 07/08/2023).

[40] T. Yu, D. Quillen, Z. He, *et al.*, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019.

[41] S. A. Sontakke, J. Zhang, S. M. R. Arnold, *et al.*, "Roboclip: One demonstration is enough to learn robot policies," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[42] Y. Jiang, C. Wang, R. Zhang, J. Wu, and L. Fei-Fei, *Transic: Sim-to-real policy transfer by learning from online correction*, 2024. arXiv: `2405.10315 [cs.RO]`.

[43] R. Hoque, A. Mandlekar, C. Garrett, K. Goldberg, and D. Fox, *Intervengen: Interventional data generation for robust and data-efficient robot imitation learning*, 2024. arXiv: `2405.01472 [cs.RO]`.