

A Detailed Environment Settings

Tasks. We design a series of both simulated and real-world tasks featuring discrete and continuous action spaces to evaluate the effectiveness of MEREQ. These tasks are categorized into two experiment settings: 1) Learning from synthesized expert with heuristic-based intervention rules, and 2) human-in-the-loop (HITL) experiments.

A.1 Learning from Synthesized Expert with Heuristic-based Intervention

In order to directly evaluate the sub-optimality of the learned policy through MEREQ, we specify a residual reward function and train an expert policy using this residual reward function and the prior reward function. We then define a heuristic-based intervention rule to decide when the expert should intervene or disengage. In this experiment setting, we consider two simulation environments for the highway driving task and the robot manipulation task.

A.1.1 Highway-Sim

Overview. We adopt the *highway-env* [50] environment for this task. The ego vehicle must navigate traffic using discrete actions to control speed and change lanes. The expert policy prefers the ego vehicle to stay in the right-most lane of a three-lane highway. Expert intervention is based on KL divergence between the expert and learned policies: the expert steps in if there is a significant mismatch for several consecutive steps and disengages once the distributions align for a sufficient number of steps. Each episode lasts for 40 steps. The sample roll-out is shown in Fig. 5.

Rewards Design. In *Highway-Sim* there are 5 available discrete actions for controlling the ego vehicle: $\mathcal{A} = \{a_{\text{lane_left}}, a_{\text{idle}}, a_{\text{lane_right}}, a_{\text{faster}}, a_{\text{slower}}\}$. Rewards are based on 3 features: $\mathbf{f} = \{f_{\text{collision}}, f_{\text{high_speed}}, f_{\text{right_lane}}\}$, defined as follows:

- $f_{\text{collision}} \in \{0, 1\}$: 0 indicates no collision, 1 indicates a collision with a vehicle.
- $f_{\text{high_speed}} \in [0, 1]$: This feature is 1 when the ego vehicle’s speed exceeds 30 m/s, and linearly decreases to 0 for speeds down to 20 m/s.
- $f_{\text{right_lane}} \in \{0, 0.5, 1\}$: This feature is 1 for the right-most lane, 0.5 for the middle lane, and 0 for the left-most lane.

The reward is defined as a linear combination of the feature set with the weights θ . For the prior policy, we define the basic reward as

$$r = -0.5 * f_{\text{collision}} + 0.4 * f_{\text{high_speed}}. \quad (\text{A.1})$$

For the expert policy, we define its reward as the basic reward with an additional term on $f_{\text{right_lane}}$

$$r_{\text{expert}} = -0.5 * f_{\text{collision}} + 0.4 * f_{\text{high_speed}} + 0.5 * f_{\text{right_lane}}. \quad (\text{A.2})$$

Both prior and expert policy are trained using Deep Q-Network (DQN) [54] with the reward defined above in Gymnasium [55] environment. The hyperparameters are shown in Tab. 6.

Intervention Rule. The expert intervention is determined by the KL divergence between the expert policy π_e and the learner policy $\hat{\pi}$ given the same state observation \mathbf{s} , denoted as $D_{\text{KL}}(\hat{\pi}(\mathbf{a}|\mathbf{s}) \parallel$

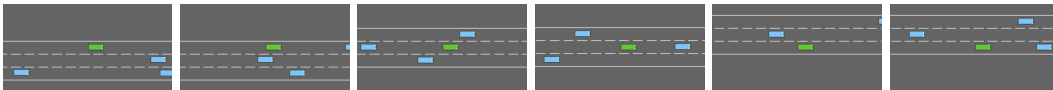


Figure 5: *Highway-Sim* Sample Roll-out. The green box is the ego vehicle, and the blue boxes are the surrounding vehicles. The bird-eye-view bounding box follows the ego vehicle.

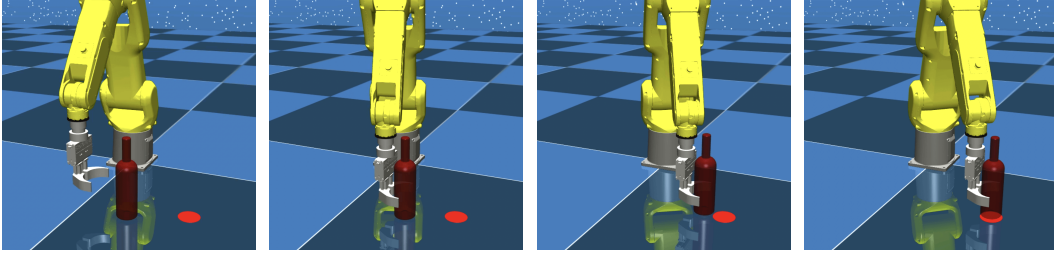


Figure 6: **Bottle-Pushing-Sim Sample Roll-out.** The location of the wine bottle and the goal are randomly initialized for each episode.

$\pi_e(\mathbf{a}|\mathbf{s})$). At each time step, the state observation is fed into both policies to obtain the expert action \mathbf{a}_e , the learner action $\hat{\mathbf{a}}$, and the expert action distribution $\pi_e(\mathbf{a}|\mathbf{s})$, defined as

$$\pi_e(\mathbf{a}|\mathbf{s}) = \frac{\exp(Q_e^*(\mathbf{s}, \mathbf{a}))}{\sum \exp(Q_e^*(\mathbf{s}, a_i))}, \quad (\text{A.3})$$

where Q_e^* is the soft Q -function. The learner’s policy distribution $\hat{\pi}(\mathbf{a}|\mathbf{s})$ is treated as a *delta distribution* of the learner action $\delta[\mathbf{a}_l]$.

We define heuristic thresholds $(D_{\text{KL},\text{upper}}, D_{\text{KL},\text{lower}}) = (1.62, 1.52)$. If the learner policy is in control and $D_{\text{KL}} \geq D_{\text{KL},\text{upper}}$ for 2 consecutive steps, the expert policy takes over; During expert control, if $D_{\text{KL}} \leq D_{\text{KL},\text{lower}}$ for 4 consecutive steps, the expert disengages. Each expert intervention must last at least 4 steps.

A.1.2 Bottle-Pushing-Sim

Overview. A 6-DoF robot arm is tasked with pushing a wine bottle to a random goal position. The expert policy prefers pushing from the bottom for safety. Expert intervention is based on state observation: the expert engages if the tooltip is too high, risking the bottle tilting for several consecutive steps, and disengages when the tooltip stays low enough for a sufficient number of steps. Each episode lasts for 100 steps. The sample roll-out is shown in Fig. 6.

Rewards Design. In *Bottle-Pushing-Sim*, the action space $\mathbf{a} \in \mathbb{R}^3$ is continuous, representing end-effector movements along the global x , y , and z axes. Each dimension ranges from -1 to 1 , with positive values indicating movement in the positive direction and negative values indicating movement in the negative direction along the respective axes. All values are in centimeter. The rewards are based on 4 features: $\mathbf{f} = \{\mathbf{f}_{\text{tip2bottle}}, \mathbf{f}_{\text{bottle2goal}}, \mathbf{f}_{\text{control_effort}}, \mathbf{f}_{\text{table_distance}}\}$, with:

- $\mathbf{f}_{\text{tip2bottle}} \in [0, 1]$: This feature is 1 when the distance between the end-effector tool tip and the wine bottle’s geometric center exceeds 30 cm, and decreases linearly to 0 as the distance approaches 0 cm.
- $\mathbf{f}_{\text{bottle2goal}} \in [0, 1]$: This feature is 1 when the distance between the wine bottle and the goal exceeds 30 cm, and decreases linearly to 0 as the distance approaches 0 cm.
- $\mathbf{f}_{\text{control_effort}} \in [0, 1]$: This feature is 1 when the end-effector acceleration exceeds $5 \times 10^{-3} \text{ m/s}^2$, and decreases linearly to 0 as the acceleration approaches 0.
- $\mathbf{f}_{\text{table_distance}} \in [0, 1]$: This feature is 1 when the distance between the end-effector tool tip and the table exceeds 10 cm, and decreases linearly to 0 as the distance approaches 0 cm.

The reward is defined as a linear combination of the feature set with the weights θ . For the prior policy, we define the basic reward as

$$r = -1.0 * \mathbf{f}_{\text{tip2bottle}} - 1.0 * \mathbf{f}_{\text{bottle2goal}} - 0.2 * \mathbf{f}_{\text{control_effort}}. \quad (\text{A.4})$$

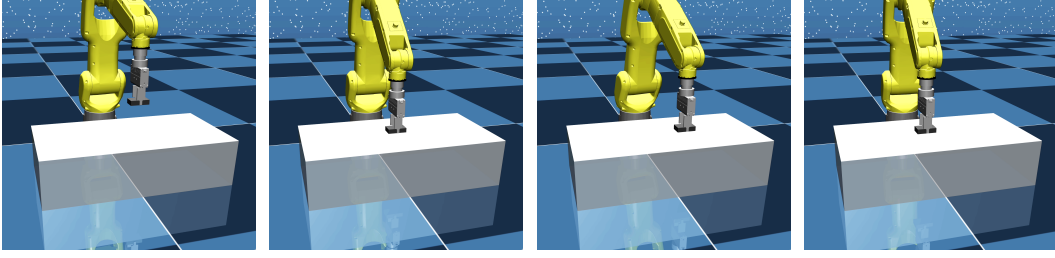


Figure 7: **Erasing-Sim Sample Roll-out.** The location of the whiteboard is fixed for each episode.

For the expert policy, we define the expert reward as the basic reward with an additional term on $\mathbf{f}_{\text{table_distance}}$

$$r_{\text{expert}} = -1.0 * \mathbf{f}_{\text{tip2bottle}} - 1.0 * \mathbf{f}_{\text{bottle2goal}} - 0.2 * \mathbf{f}_{\text{control_effort}} - 0.8 * \mathbf{f}_{\text{table_distance}}. \quad (\text{A.5})$$

Both prior and expert policy are trained using Soft Actor-Critic (SAC) [47] with the rewards defined above in MuJoCo [51] environment. The hyperparameters are shown in Tab. 8.

Intervention Rule. During learner policy execution, the expert policy takes over if either of the following conditions is met for 5 consecutive steps:

1. After 20 time steps, the bottle is not close to the goal ($\mathbf{f}_{\text{bottle2goal}} \geq 3$ cm) and the distance between the end-effector and the table exceeds 3 cm ($\mathbf{f}_{\text{table_distance}} \geq 3$ cm).
2. After 40 time steps, the bottle is not close to the goal ($\mathbf{f}_{\text{bottle2goal}} \geq 3$ cm) and the bottle movement in the past time step is less than 0.1 cm.

During expert control, the expert disengages if either of the following conditions is met for 3 consecutive steps:

1. The distance between the end-effector and the table exceeds 3 cm ($\mathbf{f}_{\text{table_distance}} \leq 3$ cm) and the bottle movement in the past time step is greater than 0.1 cm.
2. The bottle is close to the goal ($\mathbf{f}_{\text{bottle2goal}} \leq 3$ cm).

A.1.3 Erasing-Sim

Overview. A 6-DoF robot arm is tasked with erasing marker on a whiteboard on the table with an eraser. The expert policy prefers applying a larger normal force to ensure the erasing performance. Expert intervention is based on the contact force of the end-effector: the expert engages if the normal force applied by the end-effector is too small for several consecutive steps, and disengages after a fixed number of steps. Each episode lasts for 100 steps. The sample roll-out is shown in Fig. 7.

Rewards Design. In *Erasing-Sim*, the action space $\mathbf{a} \in \mathbb{R}^3$ is continuous, representing end-effector movements along the global x , y , and z axes. Each dimension ranges from -1 to 1 , with positive values indicating movement in the positive direction and negative values indicating movement in the negative direction along the respective axes. All values are in centimeter. The rewards are based on 4 features: $\mathbf{f} = \{\mathbf{f}_{\text{tip_hor_move}}, \mathbf{f}_{\text{tip_ver_dist}}, \mathbf{f}_{\text{control_effort}}, \mathbf{f}_{\text{tip_force}}\}$, with:

- $\mathbf{f}_{\text{tip_hor_move}} \in [0, 1]$: This feature is 1 when the horizontal movement of the end-effector since last step exceeds 0.6 cm, and decreases linearly to 0 as the movement approaches 0.
- $\mathbf{f}_{\text{tip_ver_dist}} \in [0, 1]$: This feature is 1 when the distance between the eraser and the whiteboard exceeds 4 cm, and decreases linearly to 0 as the distance approaches 0 cm.
- $\mathbf{f}_{\text{control_effort}} \in [0, 1]$: This feature is 1 when the end-effector acceleration exceeds $5 \times 10^{-3} \text{ m/s}^2$, and decreases linearly to 0 as the acceleration approaches 0.

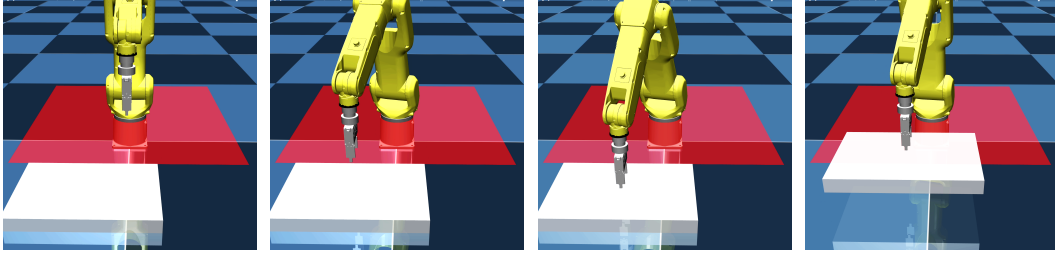


Figure 8: **Pillow-Grasping-Sim Sample Roll-out.** The expert prefers grasping from the center for improved success rate.

- $\mathbf{f}_{\text{tip_force}} \in [0, 1]$: This feature is 1 when the normal force applied by the eraser exceeds 4 N, and decreases linearly to 0 as the normal force approaches 0 N.

The reward is defined as a linear combination of the feature set with the weights θ . For the prior policy, we define the basic reward as

$$r = 1.0 * \mathbf{f}_{\text{tip_hor_move}} - 1.0 * \mathbf{f}_{\text{tip_ver_dist}} - 0.2 * \mathbf{f}_{\text{control_effort}}. \quad (\text{A.6})$$

For the expert policy, we define the expert reward as the basic reward with an additional term on $\mathbf{f}_{\text{table_distance}}$

$$r_{\text{expert}} = 1.0 * \mathbf{f}_{\text{tip_hor_move}} - 1.0 * \mathbf{f}_{\text{tip_ver_dist}} - 0.2 * \mathbf{f}_{\text{control_effort}} + 2.0 * \mathbf{f}_{\text{tip_force}}. \quad (\text{A.7})$$

Both prior and expert policy are trained using Soft Actor-Critic (SAC) [47] with the rewards defined above in MuJoCo [51] environment. The hyperparameters are shown in Tab. 8.

Intervention Rule. During learner policy execution, the expert policy takes over if the normal force applied by the end-effector is smaller than 2 N ($\mathbf{f}_{\text{tip_force}} < 2$ N) for 5 consecutive steps. Expert control will last for 5 steps and automatically disengages.

A.1.4 Pillow-Grasping-Sim

Overview. A 6-DoF robot arm is tasked with grasping a pillow with a parallel two-finger gripper. The expert policy prefers grasping from the center. Expert intervention is based on the state observation: the expert engages if the gripper is not going lower and closer to the center, and disengages when the gripper is actively moving towards the center. Each episode lasts for 100 steps. The sample roll-out is shown in Fig. 7.

Rewards Design. In *Erasing-Sim*, the action space $\mathbf{a} \in \mathbb{R}^3$ is continuous, representing end-effector movements along the global x , y , and z axes. Each dimension ranges from -1 to 1 , with positive values indicating movement in the positive direction and negative values indicating movement in the negative direction along the respective axes. All values are in centimeter. The gripper will automatically close once the end-effector reach the pillow. The rewards are based on 4 features: $\mathbf{f} = \{\mathbf{f}_{\text{tip2pillow}}, \mathbf{f}_{\text{pillow_height}}, \mathbf{f}_{\text{control_effort}}, \mathbf{f}_{\text{tip2center}}\}$, with:

- $\mathbf{f}_{\text{tip2pillow}} \in [0, 1]$: This feature is 1 when the vertical movement of the end-effector towards the surface of the pillow since last step exceeds 0.5 cm, and decreases linearly to 0 as the end-effector moving away from the surface of the pillow for more than 0.5 cm.
- $\mathbf{f}_{\text{pillow_height}} \in [0, 1]$: This feature is 1 when the distance between the pillow and the table surface 5 cm, and decreases linearly to 0 as the distance approaches 0 cm.
- $\mathbf{f}_{\text{control_effort}} \in [0, 1]$: This feature is 1 when the end-effector acceleration exceeds $5 \times 10^{-3} \text{ m/s}^2$, and decreases linearly to 2 as the acceleration approaches 0.

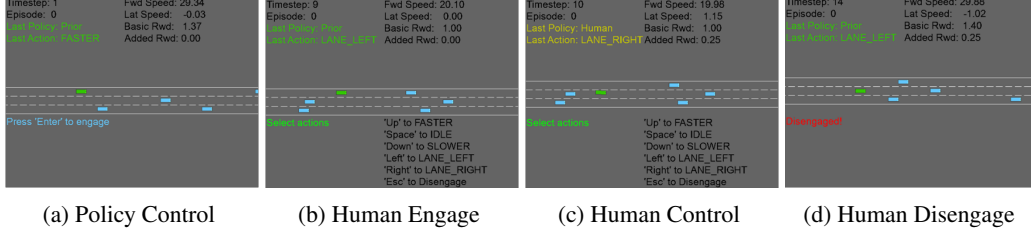


Figure 9: **Highway-Human Graphic User Interface.** There are four different scenarios during the sample collection process. When the human expert engages and takes over the control, additional information would show up for available actions.

- $f_{\text{tip2center}} \in [0, 1]$: This feature is 1 when the movement of the end-effector towards the center of the pillow since last step exceeds 0.5 cm, and decreases linearly to 0 as the end-effector moving away from the center of the pillow for more than 0.5 cm.

The reward is defined as a linear combination of the feature set with the weights θ . For the prior policy, we define the basic reward as

$$r = -0.5 * f_{\text{tip2pillow}} + 2.0 * f_{\text{pillow.height}} - 0.2 * f_{\text{control.effort}}. \quad (\text{A.8})$$

For the expert policy, we define the expert reward as the basic reward with an additional term on $f_{\text{table.distance}}$

$$r_{\text{expert}} = -0.5 * f_{\text{tip2pillow}} + 2.0 * f_{\text{pillow.height}} - 0.2 * f_{\text{control.effort}} - 0.8 * f_{\text{tip2center}}. \quad (\text{A.9})$$

Both prior and expert policy are trained using Soft Actor-Critic (SAC) [47] with the rewards defined above in MuJoCo [51] environment. The hyperparameters are shown in Tab. 8.

Intervention Rule. During learner policy execution, the expert policy takes over if:

1. The horizontal movement of the end-effector towards the center of the pillow during last step is less than a pre-defined threshold for 5 consecutive steps. The threshold varies depending on the current vertical distance between the end-effector and the center. For vertical distance larger than 5 cm, the threshold is 0.4 cm; for vertical distance between 3 cm and 5 cm, the threshold is 0.2 cm; for vertical distance smaller than 3 cm, the threshold is -0.5 cm (moving away from the center for more than 0.5 cm in the last step).
2. The vertical movement of the end-effector towards the surface of the pillow during last step is less than 0.15 cm for 10 consecutive steps.

During expert control, the expert disengages if the horizontal end-effector towards the center of the pillow during last step is greater than the pre-defined threshold for 3 consecutive steps.

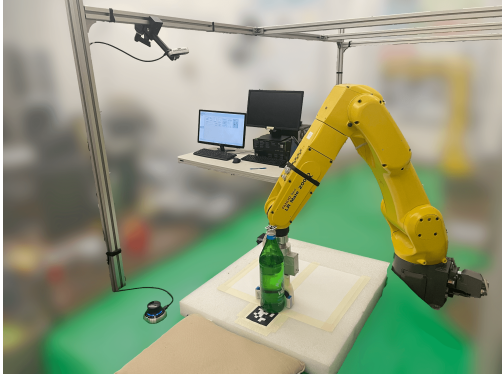
A.2 Human-in-the-loop Experiments

For the human-in-the-loop experiments, we substitute the synthesized experts in the corresponding experiments with human experts.

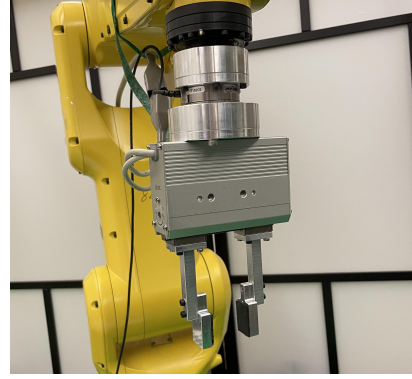
A.2.1 Highway-Human

Overview. We use the same highway-env environment with a customized Graphic User Interface (GUI) for human supervision. Human experts can intervene at will and control the ego vehicle using the keyboard. The sample GUI of 4 different scenarios are shown in Fig. 9.

Human Interface. We design a customized Graphic User Interface (GUI) for highway-env as shown in Fig. 9. The upper-left corner contains information about: 1) the step count in the current



(a) *Bottle-Pushing-Human* Hardware Setup



(b) *Pillow-Grasping-Human* Robot Gripper

Figure 10: **Hardware setups for robot experiments.** **(Left)** In the *Bottle-Pushing-Human* task, we use a Fanuc LR Mate 200iD/7L 6-DoF robot arm mounted on a tabletop, a fixed RealSense D435 depth camera for tracking AprilTags attached to the bottle and goal position, and a 3Dconnexion SpaceMouse for online human intervention. **(Right)** In the *Pillow-Grasping-Human* task, we use a two-finger parallel gripper mounted on the robot end-effector for grasping the pillow.

episode; 2) the total episode count; and 3) last executed action and last policy in control. The upper-right corner contains information about: 1) forward and lateral speed of the ego vehicle; and 2) basic and residual reward of the current state. The lower-left corner contains the user instruction on engaging and action selection. Whenever the human user is taking control, the lower-right corner shows the available actions and the corresponding keys.

A.2.2 Bottle-Pushing-Human

Overview. We use a Fanuc LR Mate 200iD/7L 6-DoF robot arm with a customized tooltip to push the bottle. Human experts can intervene at will and control the robot using a 3DConnexion SpaceMouse. Please refer to Fig. 4 (left) for a sample failure rollout where the robot knocks down the wine bottle before alignment, and a sample rollout where the robot successfully pushes the bottle to the goal position after alignment.

Human Interface. The hardware setup for the real-world experiment is shown in Fig. 10a. The robot arm is mounted on the tabletop. We use the RealSense d435 depth camera to track the AprilTags attached to the bottle and the goal position for the state feedback. The human expert uses the SpaceMouse to control the 3D position and orientation of the end-effector. The end-effector consists of a pair of tooltips specifically designed for the bottle-pushing task, which are 3D printed and attached to a parallel gripper with a fixed distance between the two fingers.

A.2.3 Pillow-Grasping-Human

We use the same robot arm with a standard two-finger parallel gripper (see Fig. 10b) to grasp the pillow. Human experts can intervene at will and control the robot using a 3DConnexion SpaceMouse. Please refer to Fig. 4 (right) for a sample failure roll-out where the robot fails to grasp the pillow by the center before alignment, and a sample roll-out where the robot successfully grasps the pillow by the center after alignment. The human interface is the same as *Bottle-Pushing-Human*.

B Additional Results

Sample Efficiency. Tab. 1 and Tab. 2 present the detailed numerical results corresponding to the plots shown in Fig. 2 and Fig. 3, respectively. Both tables report the mean values and 95% confidence intervals of the number of expert samples required by each algorithm. The results clearly demonstrate the advantage of MEREQ over the baseline methods with respect to sample efficiency.

Table 1: **MEReQ** and its variation **MEReQ-NP** require fewer total expert samples to achieve comparable policy performance compared to the max-ent IRL baselines **MaxEnt** and **MaxEnt-FT**, and interactive imitation learning baselines **HG-Dagger-FT** and **IWR-FT** under varying criteria strengths in different task and environment. Results are reported in mean (95%ci).

| <i>Environment</i> | δ | MEReQ | MEReQ-NP | MaxEnt | MaxEnt-FT | HG-Dagger-FT | IWR-FT |
|----------------------------|----------|-------------------|-----------------|---------------|------------------|---------------------|-------------------|
| Highway-Sim | 0.05 | 1819 (456) | 1990 (687) | 4363 (1266) | 4330 (1255) | 1871 (183) | 2284 (1039) |
| | 0.1 | 1208 (254) | 1043 (154) | 2871 (1357) | 1612 (673) | 1754 (160) | 1856 (1214) |
| | 0.15 | 965 (100) | 965 (37) | 2005 (840) | 1336 (468) | 1458 (194) | 1527 (930) |
| Bottle-Pushing-Sim | 0.05 | 1707 (261) | 3338 (1059) | 5298 (2000) | 2976 (933) | 2519 (1459) | 3554 (1118) |
| | 0.1 | 1613 (141) | 2621 (739) | 4536 (1330) | 2636 (468) | 1706 (785) | 2280 (1273) |
| | 0.15 | 1604 (134) | 2159 (717) | 4419 (1306) | 2618 (436) | 1692 (787) | 1290 (516) |
| Erasing-Sim | 0.05 | 925 (51) | 989 (228) | 8627 (3019) | 1899 (2796) | 1268 (827) | 4236 (1670) |
| | 0.1 | 923 (45) | 989 (228) | 7965 (3610) | 1899 (2796) | 1258 (842) | 3643 (2231) |
| | 0.15 | 923 (45) | 989 (228) | 7965 (3610) | 1899 (2796) | 1258 (842) | 2968 (1934) |
| Pillow-Grasping-Sim | 0.05 | 2848 (699) | 3086 (672) | 4992 (2375) | 3188 (1360) | 7699 (624) | 9645 (1034) |
| | 0.1 | 2398 (470) | 2807 (558) | 4127 (2737) | 2808 (1135) | 6490 (1696) | 9645 (1034) |
| | 0.15 | 2284 (332) | 2564 (633) | 3993 (2681) | 2715 (913) | 5427 (2170) | 8879 (1960) |

Table 2: **MEReQ** require fewer total human samples to align the prior policy with human preference. Results are reported in mean (95%ci).

| <i>Environment</i> | MEReQ | MaxEnt | MaxEnt-FT | HG-Dagger-FT | IWR-FT |
|------------------------------|------------------|---------------|------------------|---------------------|---------------|
| Highway-Human | 654 (174) | 2482 (390) | 1270 (440) | 864 (194) | 927 (237) |
| Bottle-Pushing-Human | 423 (107) | 879 (56) | 564 (35) | 450 (105) | 524 (130) |
| Pillow-Grasping-Human | 149 (20) | 376 (123) | 234 (141) | 456 (126) | 497 (301) |

Behavior Alignment. As discussed in Sec.6.1, when using a synthesized expert, we can directly measure the alignment between the behaviors of the learned and expert policies, since both the expert policy distribution and the ground-truth expert reward are available. Specifically, for the *Bottle-Pushing-Sim* task, we collect sample rollouts from both policies, estimate their feature distributions, and compute the Jensen–Shannon divergence [56] between these distributions as a quantitative measure of behavior alignment. The feature distributions and their corresponding Jensen–Shannon divergences relative to the expert policy are shown in Fig. 11 and Tab. 3. We also visualize the reward distributions for all policies in Fig. 11 and report their means and standard deviations in Tab. 4. These results show that the MEReQ policy more closely matches the synthesized expert in terms of both feature and reward distributions compared to the baseline methods.

Performance under Noisy Intervention. Our work focuses on learning from interventions provided by a single, consistent human expert—*i.e.*, assuming a single trainer whose behavior preference does not shift. There could still be some noise, and indeed that was not controlled for in the experiments. However, handling noisy or inconsistent interventions is beyond our scope and remains a valuable direction for future work. As a preliminary exploration, we introduced Gaussian noise (mean 0, standard deviation 0.1) to the normalized actions $[-1, 1]$ of synthesized expert interventions (see Tab. 5, results are reported in mean (95%ci) with $\delta = 0.1$) in the *Bottle-Pushing-Sim* environment. While MEReQ’s performance degrades under injected noise, it still outperforms the baselines.

C Implementation Details

In this section, we provide the hyperparameters for the prior policy training (see Tab. 6 and Tab. 8) and the Residual Q-Learning training (see Tab. 7 and Tab. 9).

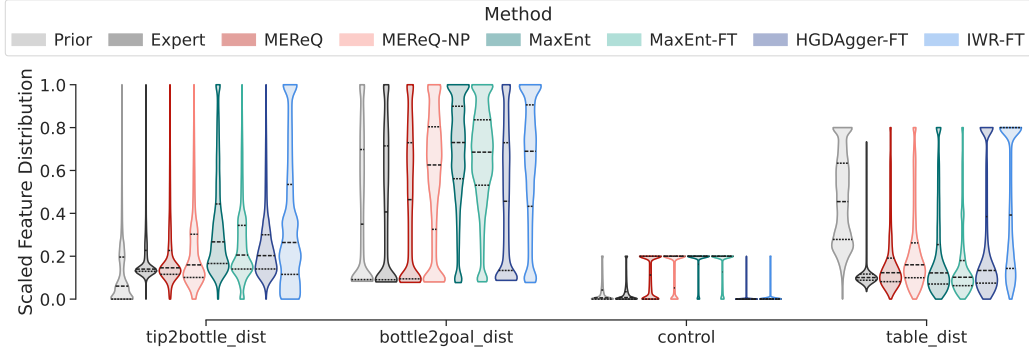


Figure 11: **Behavior Alignment.** We evaluate the policy distribution of all methods with a convergence threshold of 0.1 for each feature in the *Bottle-Pushing-Sim* environment. All methods align well with the **Expert** in the feature `table_dist` except for **IWR-FT**. Additionally, **MEREQ** aligns better with the **Expert** across the other three features compared to other baselines.

Table 3: The Jensen-Shannon Divergence of the feature distribution between each method and the synthesized expert in the *Bottle-Pushing-Sim* environment. Results are reported in mean (95%ci). The intervention rate threshold is set to 0.1.

| Features | MEREQ | MEREQ-NP | MaxEnt | MaxEnt-FT | HG-DAGger-FT | IWR-FT |
|-----------------------|----------------------|-----------------|---------------|------------------|---------------------|---------------|
| scaled_tip2wine | 0.237 (0.032) | 0.265 (0.023) | 0.245 (0.022) | 0.250 (0.038) | 0.240 (0.017) | 0.302 (0.058) |
| scaled_wine2goal | 0.139 (0.005) | 0.194 (0.044) | 0.247 (0.046) | 0.238 (0.039) | 0.167 (0.033) | 0.236 (0.040) |
| scaled_eef_acc_sqrsum | 0.460 (0.018) | 0.479 (0.022) | 0.500 (0.026) | 0.505 (0.016) | 0.707 (0.006) | 0.654 (0.022) |
| scaled_table_dist | 0.177 (0.021) | 0.219 (0.025) | 0.236 (0.029) | 0.210 (0.049) | 0.284 (0.080) | 0.308 (0.051) |

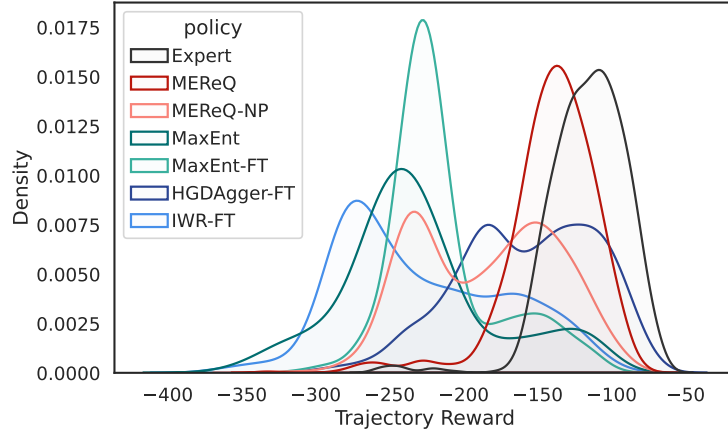


Figure 12: **Reward Alignment.** We visualize the reward distributions of all methods with a convergence threshold of 0.1 for each feature in the *Bottle-Pushing-Sim* environment. **MEREQ** aligns best with the **Expert** compared to other baselines.

Table 4: The mean and standard deviation of the reward distribution of each method.

| Expert | MEREQ | MEREQ-NP | MaxEnt | MaxEnt-FT | HG-DAGger-FT | IWR-FT |
|---------------|----------------------|-----------------|---------------|------------------|---------------------|---------------|
| -115.9 (25.9) | -140.5 (30.8) | -184.7 (46.9) | -231.1 (52.9) | -214.1 (36.7) | -157.5 (46.1) | -228.1 (56.1) |

Table 5: Number of total expert samples with noisy intervention.

| | MEREQ | MaxEnt-FT | HG-DAGger-FT | IWR-FT |
|-----------|-------------------|------------|--------------|--------------|
| No Noise | 1613 (141) | 2636 (468) | 1706 (785) | 2280 (1273) |
| 10% Noise | 1043 (420) | 2228 (182) | 3987 (1831) | 11921 (1749) |
| 50% Noise | 1011 (315) | 2612 (252) | 3612 (1529) | 11487 (3966) |

Table 6: Hyperparameters of DQN Policies.

| <i>Hyperparameter</i> | <i>Highway-Sim</i> | <i>Highway-Human</i> |
|------------------------|--------------------|----------------------|
| n_timesteps | 5×10^5 | 5×10^5 |
| learning_rate | 10^{-4} | 10^{-4} |
| batch_size | 32 | 32 |
| buffer_size | 1.5×10^4 | 1.5×10^4 |
| learning_starts | 200 | 200 |
| gamma | 0.8 | 0.8 |
| target_update_interval | 50 | 50 |
| train_freq | 1 | 1 |
| gradient_steps | 1 | 1 |
| exploration_fraction | 0.7 | 0.7 |
| net_arch | [256, 256] | [256, 256] |

Table 7: Hyperparameters of Residual DQN Policies.

| <i>Hyperparameter</i> | <i>Highway-Sim</i> | <i>Highway-Human</i> |
|------------------------|--------------------|----------------------|
| n_timesteps | 4×10^4 | 4×10^4 |
| batch_size | 32 | 32 |
| buffer_size | 2000 | 2000 |
| learning_starts | 2000 | 2000 |
| learning_rate | 10^{-4} | 10^{-4} |
| gamma | 0.8 | 0.8 |
| target_update_interval | 50 | 50 |
| train_freq | 1 | 1 |
| gradient_steps | 1 | 1 |
| exploration_fraction | 0.7 | 0.7 |
| net_arch | [256, 256] | [256, 256] |
| env_update_freq | 1000 | 1000 |
| sample_length | 1000 | 1000 |
| epsilon | 0.03 | 0.03 |
| eta | 0.2 | 0.2 |

Table 8: Hyperparameters of SAC Policies.

| <i>Hyperparameter</i> | <i>Bottle-Pushing-Sim</i> | <i>Bottle-Pushing-Human</i> | <i>Erasing-Sim</i> | <i>Pillow-Grasping-Sim</i> | <i>Pillow-Grasping-Human</i> |
|-----------------------|---------------------------|-----------------------------|--------------------|----------------------------|------------------------------|
| n_timesteps | 5×10^4 | 5×10^4 | 5×10^4 | 5×10^4 | 5×10^4 |
| learning_rate | 5×10^{-3} | 5×10^{-3} | 5×10^{-3} | 5×10^{-3} | 5×10^{-3} |
| batch_size | 512 | 512 | 512 | 512 | 512 |
| buffer_size | 10^6 | 10^6 | 10^6 | 10^6 | 10^6 |
| learning_starts | 5000 | 5000 | 5000 | 5000 | 5000 |
| ent_coef | auto | auto | auto | auto | auto |
| gamma | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| tau | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| train_freq | 1 | 1 | 1 | 1 | 1 |
| gradient_steps | 1 | 1 | 1 | 1 | 1 |
| net_arch | [400, 300] | [400, 300] | [400, 300] | [400, 300] | [400, 300] |

Table 9: Hyperparameters of Residual SAC Policies.

| <i>Hyperparameter</i> | <i>Bottle-Pushing-Sim</i> | <i>Bottle-Pushing-Human</i> | <i>Erasing-Sim</i> | <i>Pillow-Grasping-Sim</i> | <i>Pillow-Grasping-Human</i> |
|-----------------------|---------------------------|-----------------------------|--------------------|----------------------------|------------------------------|
| n.timesteps | 2×10^4 | 2×10^4 | 2×10^4 | 2×10^4 | 2×10^4 |
| batch_size | 512 | 512 | 512 | 512 | 512 |
| buffer_size | 10^6 | 10^6 | 10^6 | 10^6 | 10^6 |
| learning_starts | 5000 | 5000 | 5000 | 5000 | 5000 |
| learning_rate | 5×10^{-3} | 5×10^{-3} | 5×10^{-3} | 5×10^{-3} | 5×10^{-3} |
| ent_coef | auto | auto | auto | auto | auto |
| ent_coef_prior | 0.035 | 0.035 | 0.035 | 0.035 | 0.035 |
| gamma | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| tau | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| train_freq | 1 | 1 | 1 | 1 | 1 |
| gradient_steps | 1 | 1 | 1 | 1 | 1 |
| net_arch | [400, 300] | [400, 300] | [400, 300] | [400, 300] | [400, 300] |
| env_update_freq | 1000 | 1000 | 1000 | 1000 | 1000 |
| sample_length | 1000 | 1000 | 2000 | 2000 | 1000 |
| epsilon | 0.2 | 0.2 | 0.1 | 0.1 | 0.4 |
| eta | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |