Q-OIL: Optimistic Exploration in Learning from Human Interventions

Pranav Teegavarapu[1,*] **Sriyash Poddar**[1,*] **Rico Qi**[1]
Cleah Winston[1] **Krisha Khandelwal**[1] **Aykut Önol**[2]
Natasha Jaques[1,‡] **Abhishek Gupta**[1,‡]

[1]University of Washington [2]Toyota Research Institute
[*]Equal contribution [‡]Equal advising

Project website: https://weirdlabuw.github.io/qoil

Abstract: As robots are deployed in the real world, they will inevitably make mistakes. Human interventions provide valuable feedback for improving their behavior, enabling them to adapt online to new scenarios. However, studying human-in-the-loop robot learning is difficult: real-world experiments are slow and expensive, so most algorithm development relies on simulated human models. We show that common human modeling assumptions do not match how real people intervene in practice. Through analyzing intervention data from human participants, we find that humans tend to intervene when agent progress stalls, and that their corrective actions are often suboptimal. Motivated by these findings, we introduce a progress-based model of human intervention that achieves higher rank correlation with real-user algorithm rankings, enabling realistic algorithm benchmarking and development in simulation. Building on these insights, we propose Q-OIL, a method that adds optimistic bonuses to intervention transitions in reinforcement learning and softly guides exploration toward human corrections. We find that decoupled critics are critical for stability because they prevent the bonuses from backing up to pre-intervention states. As a result, Q-OIL learns more efficiently than competitive baselines, achieving 10–50% higher success in simulation and on real robots, and improving pretrained VLAs on two bimanual manipulation tasks. As we ground both evaluation and algorithms in realistic human behavior, this work provides a practical path toward scalable robot learning from interventions.

Keywords: Human-in-the-loop, Robot Learning, Reinforcement Learning

1 Introduction

Robot policies trained offline on large-scale datasets are brittle [1], often struggling to generalize beyond their training distributions [2] and failing across diverse testing conditions. For robust deployment, robots must adapt online from feedback [3, 4, 5, 6]. As robots fail around human observers, we can rely on human feedback through corrections [7, 8, 9], where humans intervene when the robot gets stuck or makes mistakes. Such signals arise organically in the real world; for example, users may physically steer a manipulator toward the goal [10, 11, 6] or take over in self-driving systems [12]. Interventions can be viewed as more than resets or recovery actions [12]. Corrective actions convey implicit preference signals, i.e., information about failures in the robot's behavior and potential ways to improve [13, 14, 15, 16, 17, 18, 19, 20].

Despite recent progress in robot learning from interventions, algorithm development remains difficult because real-world interaction is slow and expensive. Simulation offers a scalable alternative [21]. For intervention learning, we also require synthetic human models to determine when and how to intervene [16]. These enable repeatable evaluation and systematically guide algorithmic components (e.g., how to incorporate corrections, the exploration-safety trade-off, robustness to

[*]Corresponding authors: {pranavnt,sriyash}@cs.washington.edu

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

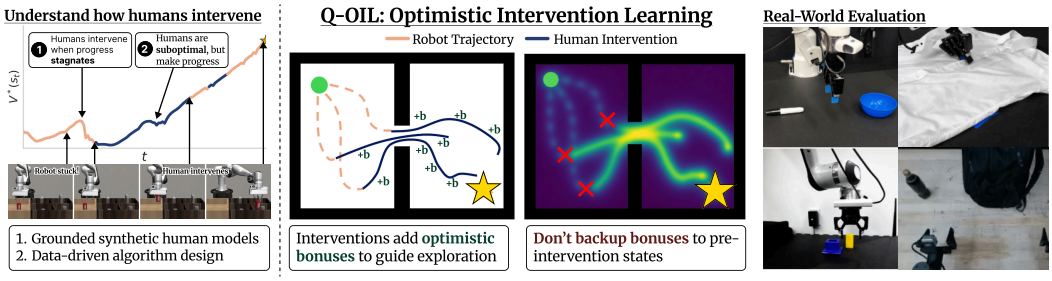


Figure 1: Overview of Q-OIL. Real-user data shows that humans intervene when robot progress stalls and provide suboptimal corrections. Q-OIL treats interventions as optimistic exploration bonuses while preventing bonuses from propagating to non-intervention states for faster real-world learning.

noise) before deployment. However, common synthetic human models and learning algorithms are insufficiently grounded in real-world behavior. They make assumptions that we show are *not reflective of real human intervention data*. This creates a gap between methods trained on synthetic human models and their performance when exposed to real users. We need a principled understanding of *when* and *how* humans intervene, both to guide our algorithmic design choices and to evaluate these methods. We first analyze real human interventions in robot manipulation. Our analysis suggests both how to better model human interventions and how to learn from them.

First, we find that rather than reacting to instantaneous suboptimality, humans intervene when the robot's progress stagnates over a horizon. This pattern aligns with cognitive models of caregiving: humans act on the agent's ability over time rather than on isolated states [22]. Given this finding, we propose a progress-based synthetic intervention model that accounts for value improvement over a horizon, better matches observed humans, and yields algorithm rankings more consistent with real human evaluation (Fig. 4). Second, this analysis shows that real corrections are suboptimal and can diverge from the learner policy during training, which suggests that algorithms for learning from interventions must be robust to noisy, divergent interventions, i.e., use feedback to adapt robot behavior without compromising asymptotic performance. This motivates our key algorithmic insight: use interventions as optimistic guidance for exploration, rather than optimal supervision. This speeds up training without compromising asymptotic performance.

Building on these insights, we introduce Q-OIL (**Q**-**O**ptimistic **I**ntervention **L**earning), a simple online RL framework that learns from real human corrections and efficiently handles diverse users and suboptimal interventions. We start by providing an additional positive reward, i.e., an optimism bonus, to intervention transitions. Intuitively, the optimism bonus uses interventions to guide exploration toward potentially successful behaviors. The influence of optimism naturally decays as interventions become less frequent, shifting learning back to the original task reward. To make this stable, we introduce a critical decoupling of optimism from TD learning [23]: the optimistic critic drives exploration and policy updates, while a task-only critic provides TD targets, preventing catastrophic bonus propagation to non-intervention states. Importantly, Q-OIL is agnostic to the underlying off-policy RL algorithm, allowing it to be incorporated into different policy learning frameworks. We demonstrate this across simulated and real-world robot manipulation tasks with human participants, and further scale Q-OIL to improve pretrained vision-language-action (VLA) policies on challenging manipulation tasks.

In summary, we make the following contributions:

1. We analyze real data to ground synthetic human models for realistic simulation evaluation and informed algorithm design. Our findings show that interventions occur when robot progress stagnates, motivating a progress-based model that promotes improved experiment design.
2. Based on human analysis, we propose Q-OIL, a simple algorithm that treats interventions as optimistic guidance for exploration rather than optimal supervision targets.
3. Across diverse real and simulated robot manipulation tasks with human participants, Q-OIL learns faster, reduces human effort, and remains robust to suboptimal and diverse interventions. We further scale it to improve pretrained VLAs on two bimanual manipulation tasks.

2 Related Work

Learning from Human Demonstrations. Behavior cloning from offline demonstrations [24, 25, 26] is a standard approach for training robot policies, but suffers from compounding errors under distribution shift at deployment [4, 19]. Reinforcement learning can learn robust behaviors from task rewards [27, 28], but is often sample-inefficient due to undirected exploration. Recent works [29, 30, 31, 5, 32] warm-start RL with offline datasets, and particularly, Lu et al. [33] combine behavior cloning with off-policy actor-critic updates. However, collecting task demonstrations is costly. In contrast, we leverage easy-to-provide online interventions [7] to bias policy behavior during training.

Interactive Learning. Interactive learning methods [4, 7, 8, 34] address the drawbacks of offline approaches by collecting online feedback, reducing compounding error from quadratic to linear regret [4]. A natural class of methods allows humans to override policy behavior with corrections [7, 11, 15, 34], which prior work uses for behavior-cloning [7, 11], as rewards [10], or for reinforcement learning [34]. Notably, Luo et al. [34] incorporate interventions into the replay buffer, making minimal assumptions about human intervention behavior, and achieve strong real-world results. Beyond corrective actions, interventions also provide implicit feedback: RLIF [16] uses them as implicit negative feedback while Spencer et al. [12] uses interventions to constrain the learner's value function. Our work investigates how corrections can guide exploration. Inspired by intrinsic-reward-driven exploration [35, 36], `Q-OIL` adds optimistic bonuses to critic targets for intervention transitions. Prior exploration methods use self-supervised prediction errors [36] or explicit human feedback [37], while we use naturally occurring corrective interventions to guide exploration.

Modeling Human Behavior. Learning from interventions requires accurate models of when and how humans intervene. Because real-world training is costly [5], prior work relies on approximate human models: TAMER [14] learns a scalar reward predictor, while others use Boltzmann-rational models [10, 38], probit triggers [17], or threshold-based rules tied to instantaneous suboptimality [16]. A common assumption is that human corrections are Markovian, i.e., merely reactions to instantaneous suboptimality, which our analysis contradicts. Consistent with cognitive theories of utility trade-offs [39] and computational models of caregiving [22], our proposed model accounts for agent progress over a short horizon when triggering interventions, grounded in real human data.

3 Problem Statement

We build on the framework of interactive imitation learning, focusing on learning from *human interventions* [7]. Consider an interactive control setting over an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $\mathcal{T}, \mathcal{R}, \rho_0, \gamma$ are the transition kernel, (sparse) reward function, initial state distribution, and discount, respectively. The objective is to learn a policy $\pi_\theta(a|s)$ that maximizes expected discounted return, i.e., $\pi_\theta = \arg\max_\theta \mathbb{E}_{\pi_\theta}\big[\sum_{t\geq 0} \gamma^t r(s_t)\big]$. During deployment, a human observes the agent and may intervene. We model the human as a decision function $\mathcal{H} = (g, \pi_h)$ with two components: a *gating function* g that decides *when* to intervene, and a *human policy* π_h that decides *how* to intervene. To allow temporal context, both may depend on a recent history $\tau_{t-L:t} = (s_{t-L}, a_{t-L}, \ldots, s_t)$, so, $g : \mathcal{S}^{L+1} \times \mathcal{A}^L \to [0, 1]$ outputs the probability of intervening at time t, and $\pi_h : \mathcal{S}^{L+1} \times \mathcal{A}^L \to \mathcal{A}$ returns a distribution over actions. The resulting rollout policy is the mixture $\pi'(a \mid s_t, \tau_{t-L:t}) = g(\tau_{t-L:t})\, \pi_h(a \mid \tau_{t-L:t}) + \big(1 - g(\tau_{t-L:t})\big)\, \pi_\theta(a \mid s_t)$. Thus, modeling human behavior requires simulating g and π_h. In the following sections, we (i) collect and analyze real human interventions to characterize when and how people intervene (Section 4), (ii) propose a non-Markovian progress-based model for g validated against this data (Section 4.2), and (iii) leverage this analysis to develop a new algorithm addressing key failure modes of prior approaches in learning from interventions (Section 5).

4 Rethinking Simulated Interventions

Developing methods that learn from interventions requires extensive experiments, but real-user studies are slow to reproduce, making it impractical to explore the full design space in deployment. Thus,

researchers rely on simulated human models for scalable training and evaluation to build better learning methods. However, if the human model is misaligned with reality, methods optimized for these models and algorithmic choices inferred from such experiments may not transfer to learning from real users. This motivates a central question: *when and how do humans actually intervene?*

Prior Approaches to Simulating Interventions. Prior works build simulated models of both intervention timing and corrective actions, with different assumptions. Korkmaz and Biyik [17] model intervention decisions using a probit choice model that accounts for a nominal human action and the human's model of the robot. Luo et al. [16] uses a Markovian model of intervention decisions based on instantaneous action suboptimality relative to a reference policy. Beyond deciding when to intervene, simulated interventions require a corrective policy π_h, which may be instantiated by a reference policy π^* or a weaker policy to model imperfect corrections. Below, we analyze human interventions and find that humans intervene when the robot's perceived progress is slow and that human corrections are suboptimal. These findings contradict prior assumptions, and thus, conclusions based on such models can diverge from experiments with real humans.

Collecting Real Human Intervention Data. We ran a pilot study with 8 participants supervising a simulated robot across 3 manipulation tasks, yielding 24 runs and $\sim$ 200k transitions. Participants provided interventions during training (Section 6), following Luo et al. [34]. We recorded when takeovers occurred and how users corrected the robot across learning stages. The study received IRB approval (Appendix D.1). For analysis, we trained task-specific reference policies and value functions, π^*, $Q^*(s,a)$, and $V^*(s)$, using RLPD [32] with sparse rewards, allowing us to quantify interventions in terms of optimality and convergence. We include participant scores in Appendix D.

4.1 Analyzing Real Human Intervention Data

Finding 1: Human intervention gating is non-Markovian.

Typically, the gating function is modeled as *Markovian* and reactive to the suboptimality of the action, implying that intervention timing depends only on the current state and action. Figure 2a contradicts this: $Q^*(s,a_\pi)/Q^*(s,a^*)$ is nearly identical for intervention vs. non-intervention states, indicating similar instantaneous values. But, when we measure k step progress as $V^*(s_t) - V^*(s_{t-k})$, the three separate value trajectories (Fig. 2b) show that takeovers follow plateaus or drops in $V^*(s_t)$, suggesting stagnation rather than single-step failure. This matches evidence that corrections can lag agent behavior [14]. Thus, Markovian models insufficiently represent humans, leading to experimentally invalid conclusions.

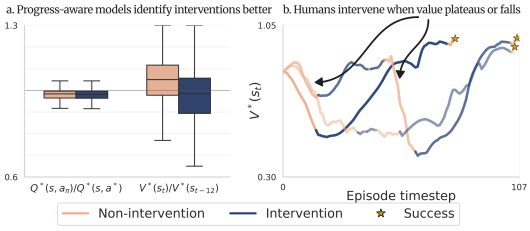


Figure 2: (a) Progress separates interventions better than instantaneous values. (b) Qualitatively, humans intervene when value plateaus or declines.

Finding 2: Human interventions are suboptimal demonstrations of the target behavior.

Several methods [7, 11] treat human corrections as demonstrations, assuming π_h is near-optimal. To quantify takeover quality, we compare the value of the intervention action to that of the expert action at the same state, and compute a suboptimality score defined in Appendix B. Under this metric, interventions are more suboptimal than π^* and human demonstrations. Figure 2 shows that interventions do not consistently increase $V^*(s_t)$. These results motivate methods that are robust to noisy corrections.

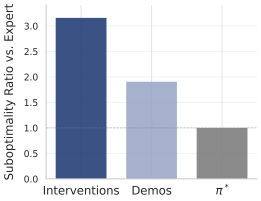
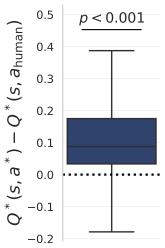


Figure 3: Interventions are suboptimal to demos and π^*.

Finding 3: Human corrections diverge from the learned policy over training.



Across the intervention dataset ($\sim$ 40k intervention transitions), we find a systematic mismatch between human corrections and the learner's converged behavior. Using Q^*, we compare values between the converged policy (a^{final}) and human actions and observe $Q^*(s, a^{\text{final}}) \geq Q^*(s, a_{\text{human}})$, implying that at the same states, the final policy prefers different actions with higher values. As both have high success ($\sim$70-80%), the gap is better explained as divergence in solution modes, which is a result of sparse rewards admitting multiple solutions (e.g., left- vs. right-side grasps). Human supervisors do not observe the full policy, so may provide corrections with an alternative strategy. Therefore, treating corrections as strict supervision can induce undesirable multi-modality, so instead we motivate using interventions as soft guidance for exploration to speed up training.

4.2 A Non-Markovian Gating Model for Human Interventions

Motivated by these findings, we define g as a progress-aware model (PAM) of when humans intervene over recent history $\tau_{t-k:t}$:

$$g(\tau_{t-k:t}) = \Pr(\nu_t = 1 \mid \tau_{t-k:t}) = \begin{cases} \alpha, & \text{if } V^*(s_t) - V^*(s_{t-k}) < \delta, \\ \beta, & \text{otherwise.} \end{cases} \quad (1)$$

using $V^*(s_t) - V^*(s_{t-k})$ as k-step progress, and where k is the horizon, δ the progress threshold, and α, β capture intervention stochasticity. Thus, interventions are likely when recent value progress stagnates (including failures where $V^*(s_t) - V^*(s_{t-k}) < 0$). We tune k, δ by grid search (Appendix F) on real human data. To model how humans intervene, i.e., π_h, we use an intermediate RL checkpoint ($\sim$50–70% success), reflecting that corrections are imperfect.

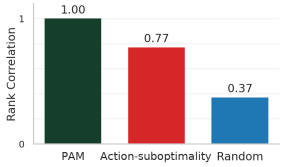


Figure 4: ρ between approximate models and real human algorithm rankings.

Validation Against Real-Human Rankings. A useful intervention model should predict which algorithms succeed with real users. We validate g by held-out intervention prediction and ranking agreement with real-human evaluations. Across all runs, we rank six learning algorithms (Section 6) by final success under three models: our PAM model, action suboptimality [16], and a random model calibrated to the true intervention rate, and compare these rankings to human evaluations using Spearman correlation.

Our progress-based gating approach is a stronger approximate human model for evaluating robot intervention learning. Figure 4 shows stronger agreement with real-human rankings ($\rho = 1.0$) than alternative gating models, providing a more reliable proxy for benchmarking algorithms with real users. We provide further discussion in Appendix F.

5 How Should We Use Human Interventions for Robot Policy Learning?

As outlined in Section 4.1, prior methods for intervention learning rely on assumptions unsupported by real data. Given this, we ask: *how should we use imperfect human interventions for robots?*

Human interventions provide optimistic signals to guide exploration during RL.

To this end, grounded in our findings about human behavior, we present an approach based on off-policy model-free RL and optimistic exploration [35]: `Q-OIL`, which adds a positive reward bonus at intervention transitions, encouraging exploration informed by corrections without forcing strong alignment to potentially noisy and divergent corrections (Findings 2 and 3). Standard off-policy RL learns a critic by using the given rewards and bootstrapping the value of future states. Training a single critic

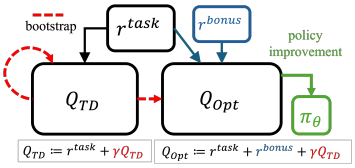


Figure 5: Overview of `Q-OIL`

with the optimism bonus and bootstrapped updates would propagate this bonus backward, reinforcing the actions that triggered the intervention. We address this catastrophic over-optimism via decoupling the critic into an optimistic and task critic, as illustrated in Fig. 5. This is critical because an optimistic critic (trained on task reward + bonus) bootstraps from the task critic (trained only on task reward), thereby confining the bonus to intervention transitions. The optimistic critic improves the policy, while the task critic is an unbiased estimate of the true value function. We discuss this further in Appendix A. Finally, we show that this human-informed method improves performance with real users and the better-aligned synthetic human model.

5.1 Q-Optimistic Intervention Learning

Practically, we learn a policy π_θ, task critic Q_{TD}, and optimistic critic Q_{Opt} using an online off-policy actor–critic algorithm [27, 32], though in principle it may improve any RL method. In Algorithm 1, π_θ is rolled out to interact with the environment, while a human or an approximate model supervisor intervenes with corrective actions. D_π stores transitions $(s_t, a_t, r_t^{\text{task}}, s_{t+1})$ and bonus reward $r_t^{\text{bonus}} = b * \mathbb{1}(\text{intervene})$, while D_{int} stores intervention state-action pairs (s_t, a_t). Optionally, we can initialize D_π and D_{int} with a few offline demonstrations to warm-start training [32]. The backup critic Q_{TD} is updated on D_π by minimizing the standard Bellman regression loss, $\mathcal{L}_{\text{TD}} = \mathbb{E}_{(s,a,r_t^{\text{task}},s')\sim D_\pi}\left[Q_{\text{TD}}(s,a) - \left(r_t^{\text{task}} + \gamma\, \mathbb{E}_{a'\sim\pi_\theta(\cdot|s')}\, Q_{\text{TD}}(s',a')\right)\right]^2$. For the optimistic critic Q_{Opt}, we add the optimistic reward r^{bonus} but use TD targets from Q_{TD}, while the actor π_θ objective combines (i) RL policy improvement using Q_{Opt} and (ii) maximum likelihood on D_{int}:

Algorithm 1

Require: π_θ, π^{int}, D_π, D_{int}, bonus $b > 0$
for trial $i = 1$ to N **do**
  `// Training decoupled critics`
  Train Q_{TD} on D_π via $\mathcal{L}_{\text{TD}}$
  Train Q_{Opt} on D_π via $\mathcal{L}_{\text{Opt}}$
  `// Policy improvement`
  Update π_θ on D_π and D_{int} via $\mathcal{L}_\theta$
  for timestep $t = 1$ to T **do**
    // `rollout` π_θ
    if π^{int} intervened at $t-1$ **then**
      $r_{t-1}^{\text{bonus}} = b$
      append $(s_{t-1}, a_{t-1}^{\text{int}})$ to D_{int}
    else
      $r_{t-1}^{\text{bonus}} = 0$
    end if
    add $(s_{t-1}, a_{t-1}, r_{t-1}^{\text{task}}, r_{t-1}^{\text{bonus}}, s_t)$ to D_π
  end for
end for

$$\mathcal{L}_{\text{Opt}} = \mathop{\mathbb{E}}_{(s,a,r_t^{\text{task}},r^{\text{bonus}},s')\sim D_\pi}\left[Q_{\text{Opt}}(s,a) - \left(\underbrace{r_t^{\text{task}} + r^{\text{bonus}}}_{\text{adding optimism}} + \gamma \underbrace{\mathop{\mathbb{E}}_{a'\sim\pi_\theta(\cdot|s')} Q_{\text{TD}}(s',a')}_{\text{using TD-backup}}\right)\right]^2$$

$$\mathcal{L}_\theta = -\underbrace{\mathop{\mathbb{E}}_{s\sim D_\pi, a\sim\pi_\theta(\cdot|s)}\left[Q_{\text{Opt}}(s,a)\right]}_{\text{optimistic policy improvement}} - \underbrace{\mathop{\mathbb{E}}_{(s,a)\sim D_{\text{int}}}\left[\lambda \log \pi_\theta(a|s)\right]}_{\text{BC-regularization}}$$

In `Q-OIL`, we apply the bonus to interventions, and during training we sample the replay uniformly. As a result, the probability of sampling bonus transitions is proportional to the cumulative intervention rate ρ_t. At convergence $\rho_t \to 0$, so $Q_{\text{Opt}} \to Q_{\text{TD}}$, i.e., the bonus anneals implicitly, leading to no asymptotic bias from the optimistic critics. Thus, `Q-OIL` speeds up training while remaining robust to potential bias from human intervention. Additionally, `Q-OIL` is a simple extension on top of any RL fine-tuning algorithm, and we extend this method to fine-tuning pretrained vision-language-action models (VLAs) in Algorithm 2.

BC regularization. Interventions are suboptimal (Section 4.1) but provide a useful early directional signal compared to random exploration (see value improvement over corrections in Figure 2). We therefore regularize the actor with a BC term on intervention actions in D_{int} with a fixed $\lambda \approx 0.1$.

6 Experiments

Our experiments aim to answer whether `Q-OIL` improves the ability to learn from interventions:

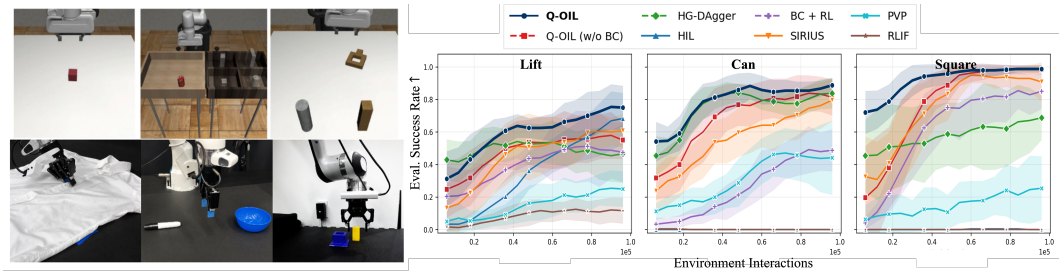


Figure 6: **Left:** Evaluation tasks in simulation and on the Franka robot. **Right:** On Robomimic with the progress-aware human model (PAM), `Q-OIL` improves success rate and learning efficiency over baselines.

1. in simulated robot experiments against the proposed model of human behavior;
2. with real humans intervening on simulated robots;
3. with real human interventions on a robot; and
4. when fine-tuning pretrained VLAs on bimanual tasks.

Finally, ablations investigate how components such as the decoupled critics and BC-regularization affect performance.

6.1 Experimental Setup

Tasks. We evaluate our method on real and simulated manipulation tasks using both simulated human models and real participants: (i) **Pen-in-Bowl:** grasp a pen and place it in a bowl; (ii) **Unbutton-Shirt:** open a shirt button, requiring fine-grained deformable-object manipulation; (iii) **Peg-Insertion:** pick and precisely insert a randomly placed peg; (iv) **Bag Assembly:** use two arms to pick up a bottle and insert it into a backpack's side pocket; (v) **Coffee Pouring:** mimicking a wet-lab task, pick a test tube filled with coffee beans from a rack and pour the beans into a beaker, requiring precise grasping and pouring; and (vi) **Simulated Robomimic:** three simulated pick-and-place/assembly tasks [40].

Baselines. We compare against methods spanning the three categories in Section 2: (i) **HIL** [34], which accelerates off-policy RL with online interventions in the replay buffer; (ii) **HG-DAgger** [7], a DAgger [4] variant that treats interventions as BC targets; (iii) **SIRIUS** [41], which applies weighted BC to interventions and on-policy data; (iv) **PVP** [42], which modifies the critic loss to prefer human actions; (v) **RLIF** [16], which assigns negative reward to pre-intervention states during off-policy RL; and (vi) **BC+RL** [43], which combines BC on intervention actions with off-policy RL.

Training Details. We use a JAX-based implementation built on RLPD [32]. In simulation, we use our synthetic model PAM for reliable benchmarking and report success rates over 100k environment interactions. We then test algorithms with four real participants on two state-based simulated tasks. We narrow the reset distribution when real humans intervene in simulation to reduce training time. For real-robot tasks, we train RGB policies with wrist-mounted and third-person cameras on the Franka Panda robot and run training with a single human across three tasks. We warm-start experiments with 10-20 demonstrations, and evaluate each checkpoint for 10–20 episodes (100 for simulated tasks). For all our Franka experiments, we use RLPD [32] as our base algorithm. For our bimanual experiments, we fine-tune a pretrained VLA with EXPO-FT [44], which effectively learns a residual policy via interventions (See Appendix E.1). We include further details about the robot setup, tasks, and hyperparameters in Appendices E and I.

6.2 Results

Does `Q-OIL` improve success rate and learning efficiency in simulated tasks?

In Figure 6 (right), we evaluate `Q-OIL` on Robomimic using our progress-aware human model (PAM), which we show (Section 4.2) better aligns with real human interventions and evaluations. Across tasks, `Q-OIL` consistently achieves higher success rates with fewer interactions than base-

lines. This highlights that optimistic exploration with noisy corrections is more effective than directly imitating, penalizing, or naively adding interventions to the replay buffer. We also compare all methods under existing human models and ablate expert quality in Appendices F and H.

Does Q-OIL improve task performance and reduce real human effort when participants interact with simulated robots?

To evaluate performance under real human interventions, we recruited four participants to intervene on two Robomimic tasks across six methods, yielding 48 runs; see Appendix D.2 for details. In Figure 7 (top), we observe that Q-OIL outperforms all baselines with real humans, consistent with our simulated intervention results. HG-DAgger [7] has non-trivial performance but over-relies on imitating the interventions (which are suboptimal), and therefore underperforms. HIL/RLIF [34, 16] fail to learn within the interaction budget. In contrast, Q-OIL uses optimism to explore the human-guided regions and rapidly learns a policy with a high success rate. We also find that BC regularization enables faster and lower-variance learning in the longer-horizon ($\sim$ 400 steps) task. We observe similar trends between real-human interventions and our synthetic intervention model, supporting the fidelity of our data-driven evaluation. Figure 7 (bottom) shows that Q-OIL requires fewer interventions than baselines, while HIL [34] requires substantially more effort.

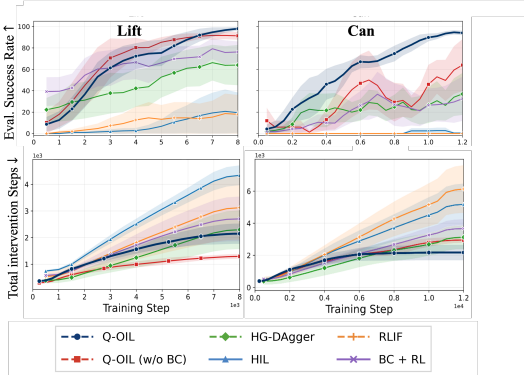


Figure 7: Q-OIL outperforms baselines when human users interact with simulated robots across two tasks.

Does Q-OIL improve efficiency and performance on real robot tasks?

We validate Q-OIL on real robot hardware using a Franka Panda (Appendix E), across three different tasks and RL baselines. Our analysis in Section 4.1 shows that human interventions can be noisy and divergent, and Q-OIL robustly utilizes interventions to learn a policy with 10–50% higher success rates and almost 30% lower effort (Figure 8). By biasing exploration toward human-corrected regions, Q-OIL collects more task-relevant experience within the same interaction budget and learns policies with an average 90% higher execution efficiency, as reflected in higher throughput, measured as successes per 1k steps. We include complete videos of our training runs, and some qualitative discussion on our website [2]. Finally, in Figure 8, we observe that Q-OIL is the most efficient in terms of interventions across tasks, i.e., it makes maximal use of each human correction to adapt its behavior. Together with our simulation results under synthetic and real corrections, these results show that human-grounded evaluation leads to better algorithmic design that transfers to real users and robots. HIL fails to learn within a smaller budget, so, for a task we run baselines for additional steps to get non-trivial task successes.

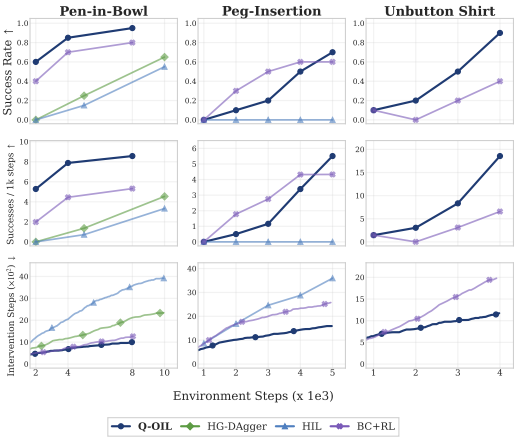


Figure 8: Across three tasks, Q-OIL achieves higher success and throughput given a fixed interaction budget, with lower human effort.

Does Q-OIL scale to improve pretrained VLAs on bimanual tasks?

We evaluate whether Q-OIL can improve a pretrained VLA on two challenging bimanual tasks: bag assembly and coffee pouring (Figure 9). We compare residual Q-OIL against the base VLA and the

[2] https://weirdlabuw.github.io/qoil

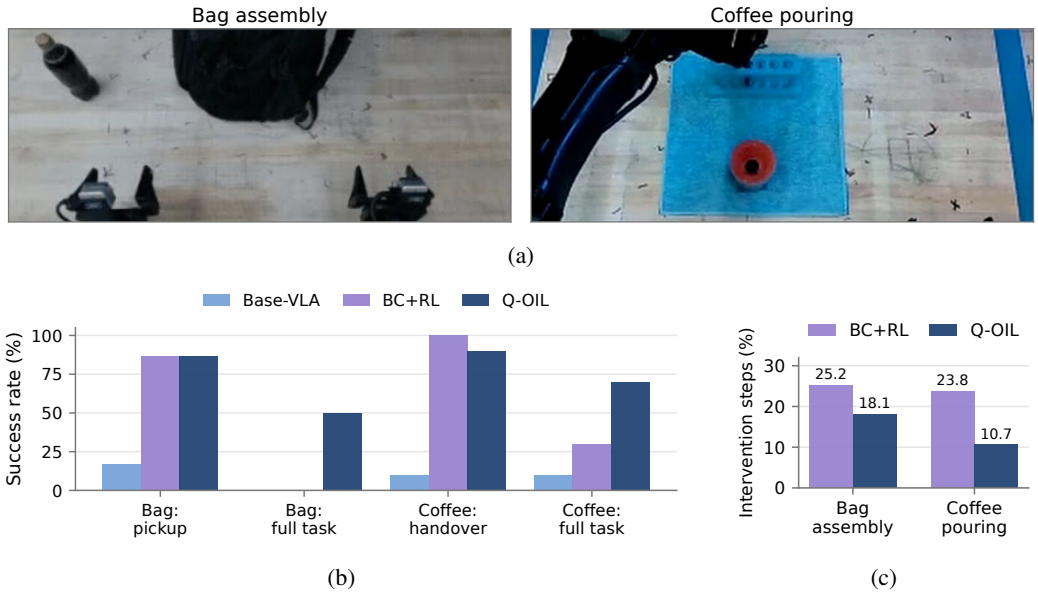


Figure 9: **Bimanual VLA Fine-Tuning.** (a) Task setups: bag assembly and coffee pouring. (b) Success rates of the base VLA, BC+RL, and Q-OIL. Bag assembly reports bottle pickup and full task completion over 30 evaluation episodes; coffee pouring reports handover and full task completion over 10 episodes. (c) Fraction of training steps under human intervention.

best-performing baseline BC+RL, using EXPO-FT [44] for residual policy learning. On both tasks, Q-OIL and BC+RL improve substantially over the base VLA on the first subtask, i.e., picking up the bottle or handing over the test tube. However, Q-OIL completes the full task far more often (50% versus 0% on bag assembly and 70% versus 30% on coffee pouring) while requiring fewer human interventions. We hypothesize that Q-OIL guides local exploration to correct the most challenging subtask, i.e., stretching the side pocket to insert the bottle or pouring the beans into the beaker, leading to higher success rates with a comparable training interaction budget.

Are the decoupled critics essential for Q-OIL stability?

In Figure 10, we ablate a key design choice: removing the decoupling between the task reward and optimism bonus. We present the success rate averaged over Robomimic tasks with our synthetic human model. We sweep the optimism bonus $b \in [0.01, 0.5]$ for Q-OIL with/without BC and decoupling, and show that Q-OIL's success is stable across this $50\times$ range. At b=0, Q-OIL reduces exactly to BC+RL. No-decoupling fails at high b. In Appendix A, we show a toy example in which removing decoupling leads to catastrophic propagation of positive bonuses to non-intervention states, causing over-optimism and higher values for suboptimal states. This severely harms performance, as shown in Figure 10, where performance decreases as we increase the bonus value for optimism.

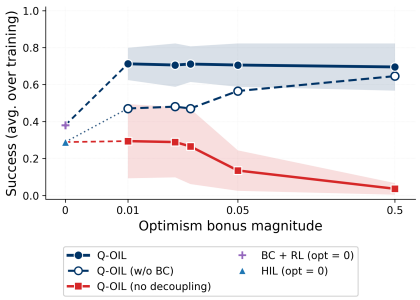


Figure 10: Q-OIL w/o decoupling degrades with higher bonus due to over-optimism.

7 Conclusion

In this work, we study real-world human interventions, revealing a significant gap between existing assumptions and actual behavior. Analyzing real intervention data, we find that humans intervene based on non-Markovian progress rather than local suboptimality, and that their corrections are suboptimal and diverge from the converged policy. These findings motivate two contributions: (1) a progress-aware intervention model (PAM) that better supports algorithm development and evaluation by matching real-user outcomes, unlike prior synthetic human models, and (2) Q-OIL, an algorithm that uses interventions to guide exploration by adding an optimism bonus at intervention

transitions, while decoupling the critics for backups and policy improvement. Across simulated and real robot experiments, Q-OIL improves task success by approximately 10–50% with 30% less human effort than baselines, providing a practical path toward scalable human-in-the-loop learning.

Limitations: Our human study uses 8 users in simulated tasks across 24 experimental runs; a larger and more diverse participant pool in the real world would further strengthen our findings about interventions. We evaluate Q-OIL on tabletop manipulation tasks; future work should study robots deployed in unstructured settings, where humans intervene across diverse, long-horizon tasks with richer intervention patterns. During training, our proposed method upweights intervention transitions, which updates the policy to explore these regions. Function approximation propagates it to similar transitions, shaping exploration beyond intervention states. An interesting future direction is to leverage foundation models to improve bonus propagation beyond local exploration to a semantically richer space such as language-based interventions.

References

[1] X. Zhou, Y. Xu, G. Tie, Y. Chen, G. Zhang, D. Chu, P. Zhou, and L. Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

[2] Y. Hu, F. Lin, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[4] S. Ross, G. J. Gordon, and J. A. Bagnell. Reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pages 627–635, 2011. URL http://www.aaai.org/ocs/index.php/AISTATS/AISTATS11/paper/view/2156.

[5] P. Yin, T. Westenbroek, S. Bagaria, K. Huang, C.-A. Cheng, A. Kolobov, and A. Gupta. Rapidly adapting policies to the real-world via simulation-guided fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2025.

[6] P. Intelligence. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025. URL https://arxiv.org/abs/2511.14759.

[7] K. Michael, 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)*, pages 5053–5059. IEEE, 2019. URL https://arxiv.org/abs/1810.02890.

[8] Y. Jiang, C. Wang, R. Zhang, J. Wu, and L. Fei-Fei. Transic: Sim-to-real policy transfer by learning from online correction. In *Conference on Robot Learning*, 2024.

[9] J. Liang, F. Xia, W. Yu, A. Zeng, M. G. Arenas, M. Attarian, M. Bauzá, M. Bennice, A. Bewley, A. Dostmohamed, C. Fu, N. Gileadi, M. Giustina, K. Gopalakrishnan, L. Hasenclever, J. Humplik, J. Hsu, N. J. Joshi, B. Jyenis, C. Kew, S. Kirmani, T.-W. E. Lee, K.-H. Lee, A. H. Michaely, J. Moore, K. Oslund, D. Rao, A. Ren, B. Tabanpour, Q. H. Vuong, A. Wahid, T. Xiao, Y. Xu, V. Zhuang, P. Xu, E. Frey, K. Caluwaerts, T.-Y. Zhang, B. Ichter, J. Tompson, L. Takayama, V. Vanhoucke, I. Shafran, M. Mataric, D. Sadigh, N. M. O. Heess, K. Rao, N. Stewart, J. Tan, and C. Parada. Learning to learn faster from human feedback with language model predictive control. *ArXiv*, abs/2402.11450, 2024. URL https://api.semanticscholar.org/CorpusID:267751232.

[10] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan. Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 141–149, 2018.

[11] A. Mandlekar, D. Xu, R. Mart'in-Mart'in, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation. *ArXiv*, abs/2012.06733, 2020. URL https://api.semanticscholar.org/CorpusID:229158088.

[12] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa. Expert intervention learning: An online framework for robot learning from explicit and implicit human feedback. *Auton. Robots*, 46(1):99–113, Jan. 2022. ISSN 0929-5593. doi:10.1007/s10514-021-10006-9. URL https://doi.org/10.1007/s10514-021-10006-9.

[13] M. Liu, M. Zhu, and W. Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[14] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: the tamer framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*, K-CAP '09, page 9–16, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605586588. doi:10.1145/1597735.1597738. URL https://doi.org/10.1145/1597735.1597738.

[15] A. Xie, F. Tajwar, A. Sharma, and C. Finn. When to ask for help: Proactive interventions in autonomous reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[16] J. Luo, P. Dong, Y. Zhai, Y. Ma, and S. Levine. Rlif: Interactive imitation learning as reinforcement learning, 2024. URL https://arxiv.org/abs/2311.12996.

[17] Y. Korkmaz and E. Biyik. Mile: Model-based intervention learning. *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15673–15679, 2025. URL https://api.semanticscholar.org/CorpusID:276444697.

[18] D. Lindner, S. Tschiatschek, K. Hofmann, and A. Krause. Interactively learning preference constraints in linear bandits. *ArXiv*, abs/2206.05255, 2022. URL https://api.semanticscholar.org/CorpusID:249605747.

[19] J. Spencer, S. Choudhury, A. Venkatraman, B. Ziebart, and J. A. Bagnell. Feedback in imitation learning: The three regimes of covariate shift, 2021. URL https://arxiv.org/abs/2102.02872.

[20] S. Ainsworth, M. Barnes, and S. Srinivasa. Mo' states mo' problems: Emergency stop mechanisms from observation, 2019. URL https://arxiv.org/abs/1912.01649.

[21] A. Jain, M. Zhang, K. Arora, W. Chen, M. Torne, M. Z. Irshad, S. Zakharov, Y. Wang, S. Levine, C. Finn, W.-C. Ma, D. Shah, A. Gupta, and K. Pertsch. Polaris: Scalable real-to-sim evaluations for generalist robot policies, 2025. URL https://arxiv.org/abs/2512.16881.

[22] R. Shachnai, M. Kleiman-Weiner, M. Berke, and J. A. Leonard. When bayesians take over: A computational model of parental intervention. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47, 2025. URL https://escholarship.org/uc/item/1q346255.

[23] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

[24] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469 – 483, May 2009.

[25] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters. An algorithmic perspective on imitation learning. *Found. Trends Robotics*, 7(1-2):1–179, 2018. doi:10.1561/2300000053. URL https://doi.org/10.1561/2300000053.

[26] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[29] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[30] A. Nair, M. Dalal, A. Gupta, and S. Levine. {AWAC}: Accelerating online reinforcement learning with offline datasets, 2021. URL https://openreview.net/forum?id=OJiM1R3jAtZ.

[31] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

[32] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[33] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, R. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7553–7560. IEEE, 2023.

[34] J. Luo, C. Xu, J. Wu, and S. Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025.

[35] R. I. Brafman and M. Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3(null):213–231, Mar. 2003. ISSN 1532-4435. doi:10.1162/153244303765208377. URL https://doi.org/10.1162/153244303765208377.

[36] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

[37] M. Torne, M. Balsells, Z. Wang, S. Desai, T. Chen, P. Agrawal, and A. Gupta. Breadcrumbs to the goal: Goal-conditioned exploration from human-in-the-loop feedback, 2023.

[38] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research*, 41(1):45–67, 2022.

[39] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1914185.

[40] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[41] 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.

[42] Z. Peng, W. Mo, C. Duan, Q. Li, and B. Zhou. Learning from active human involvement through proxy value propagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/file/f57ffe47d0b528fbb97901d16bd4eba2-Paper-Conference.pdf.

[43] S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning. *arXiv preprint arXiv:2106.06860*, 2021. URL https://arxiv.org/abs/2106.06860.

[44] P. Dong, K.-H. Hung, T. Gao, D. Sadigh, and C. Finn. Expo-ft: Sample-efficient reinforcement learning finetuning for vision-language-action models. *arXiv preprint arXiv:2605.25477*, 2026.

Appendix Contents

A Do decoupled critics help when learning from interventions? 16

B Human interventions suboptimally improve robot trajectories 16

C Simulated Robot Interface 17

D Human Experiment Details 17

D.1 Real Intervention Analysis 17

D.2 Comparing algorithm performance across synthetic and real interventions 18

E Real-World Setup 18

E.1 Extending `Q-OIL` to Pretrained VLAs 19

F Synthetic Intervention Models and Rank-Correlation Analysis 20

F.1 Parameter Selection for the Progress-Aware Model 20

F.2 Held-Out Intervention Prediction 21

F.3 Rank Correlation with Real-Human Evaluation 21

F.4 Synthetic Intervention Rollouts 21

F.5 Algorithm Performance under Different Synthetic Models of Human Intervention . 21

G Task Details 22

H How does quality of expert actions affect learning? 23

I Implementation Details 23

I.1 Hyperparameters 23

A Do decoupled critics help when learning from interventions?

In Section 5, we introduce a key design choice in `Q-OIL`: decoupling the critics into two components. The first is a task critic, which uses standard Bellman regression with only the sparse task reward. The second is an optimistic critic, which combines the task reward, the optimistic bonus, and backups from the task critic to construct Bellman targets. This design prevents the bonus from leaking to pre-intervention states. In contrast, a single critic that incorporates optimism and bootstraps from itself would propagate the bonus to states before the interventions. These are precisely the transitions where the robot was failing or acting sub-optimally, causing humans to intervene. By decoupling the critics, `Q-OIL` confines the bonus to the intervention states and avoids value overestimation, which otherwise leads to catastrophic performance degradation, as shown in Figure 10.

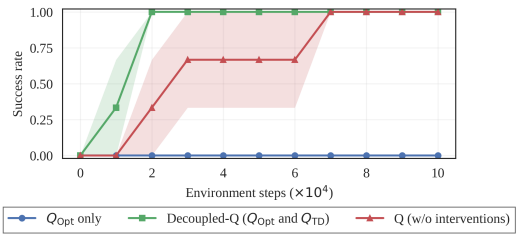
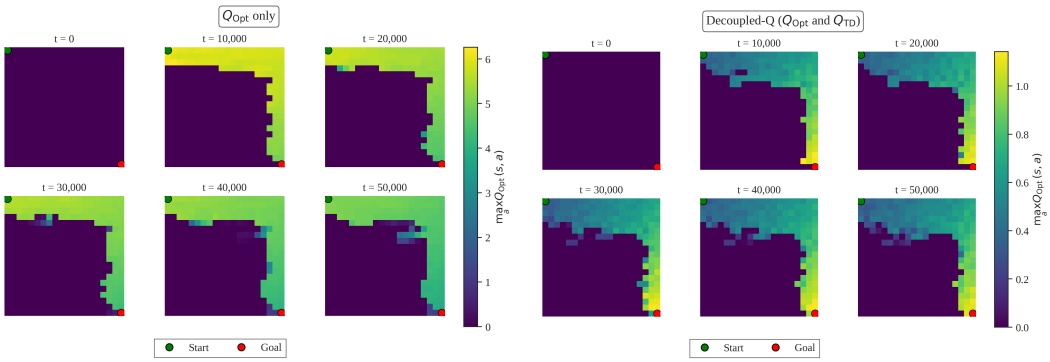


Figure 11: `Q-OIL` helps learning in the discrete 2D navigation setting, showing that decoupling is essential for learning from interventions.



(a) No Decoupling

(b) Decoupled Critics

Figure 12: Visualizing the Q-value heatmaps during training across the decoupled and non-decoupled versions of `Q-OIL`. Using a single critic overestimates the value because of the bonus leakage, while decoupling stabilizes Q-learning, enabling efficient learning from interventions.

In Figure 12, we present a toy example to visualize this behavior. The task is to navigate from a fixed start point to a fixed goal in a discrete 20×20 grid. We use a simple Q-learning algorithm to learn a policy, with a noisy expert in the loop providing interventions. As training progresses and human interventions occur in Figure 12a, the bonus propagates to all visited states in the grid, leading to value overestimation. The policy extracted from this overestimated value function fails to make progress toward the goal and gets stuck, as shown in Figure 11. With decoupling, the Q-value heatmaps are more stable and better shaped, exhibiting a clear gradient toward the goal. This enables the extraction of an efficient policy that completes the task.

B Human interventions suboptimally improve robot trajectories

To quantify action quality, we compare the value of a chosen action to the value of the expert action at the same state. For sparse goal-reaching tasks, the optimal value can be interpreted as a discounted measure of time-to-goal: if taking action a reaches the goal in $n(s,a)$ steps, then $Q^*(s,a) \propto \gamma^{n(s,a)}$. Thus, the log-ratio of action values converts value differences into differences in expected steps to the goal. For an action a, we define

$$\text{Subopt}(s,a) \triangleq 1 + \frac{\log_\gamma \left(Q^*(s,a)/Q^*(s,a^*)\right)}{\log_\gamma Q^*(s,a^*)} = 1 + \frac{n(s,a) - n(s,a^*)}{n(s,a^*)} = \frac{n(s,a)}{n(s,a^*)}.$$

Here, $n(s, a)$ denotes the number of steps to the goal after taking action a at state s. Therefore, $\text{Subopt}(s, a)$ measures the relative time-to-goal induced by action a compared to the optimal action a^*: a value of 1 indicates an optimal action, while larger values indicate actions that delay task completion. Under this metric, intervention actions have mean suboptimality 3.16, compared to 1.91 for human demonstrations and 1.00 for π^*. This suggests that intervention actions are more suboptimal than offline demonstrations under Q^*, while the learned expert policy is optimal by construction. We present some additional analysis to understand how interventions affect progress toward the goal. To measure whether a takeover moves the robot closer to success, we score each intervention segment from s_t to s_{t+k} using,

$$\text{Progress}(s_t \to s_{t+k}) = \frac{1}{k} \log_\gamma \left(\frac{V^*(s_t)}{V^*(s_{t+k})} \right),$$

where k is the segment length. Under sparse terminal rewards, $V^*(s) \approx \gamma^{d^*(s)}$, so this score approximates the reduction in optimal steps-to-go per environment step. A score of 1 indicates near-expert progress, values between 0 and 1 indicate slower, but positive, progress, and negative values indicate negative progress. We found that 83.0% of intervention segments fall below 1, and 30.1% are negative, showing that human takeovers frequently fail to make expert-like progress.

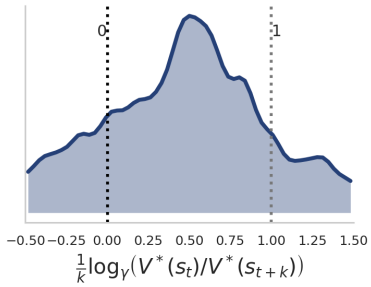


Figure 13: Human intervention segments often make limited value-based progress.

C Simulated Robot Interface

We conduct these experiments to analyze human interventions with simulated robots because it enables faster and more scalable data collection and, more importantly, we can train reference policies/critics for controlled quantitative analysis. Participants teleoperated the simulated robot end-effector using a 3Dconnexion SpaceMouse. The interface displayed two synchronized rendered camera views, shown in Figure 14: an external agent view showing the robot, workspace, objects, and target receptacle, and a close-up gripper view centered on the end-effector and manipulated object. During each episode, participants monitored these views and could intervene by taking control with the SpaceMouse. SpaceMouse inputs were mapped to Cartesian end-effector position deltas, while a binary command controlled gripper opening and closing. We restricted the teleoperation interface to these 4-DoF commands, rather than full pose control, to reduce the difficulty of providing timely and consistent interventions.

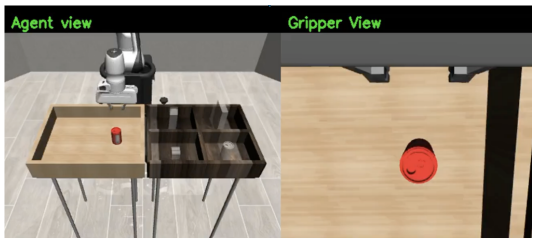


Figure 14: Rendered observation shown to the humans during simulated robot experiments.

D Human Experiment Details

D.1 Real Intervention Analysis

To study how humans intervene during robot learning, we recruited eight participants familiar with robot teleoperation. Our institutional review board approved the study. We first onboarded participants to the intervention protocol and the simulated robot interface described in Appendix C. We conducted the study in simulation to enable faster and more scalable data collection, and to obtain converged reference behavior for controlled quantitative analysis.

Each participant supervised the robot on three robomimic tasks while it was trained with HIL [34]. During training, participants observed the robot and intervened when they judged its behavior was

unsuccessful or making insufficient progress. To limit variation and speed up convergence, we kept the starting end-effector pose and the goal position constant for each task. Across all participants and tasks, we logged the full robot trajectories, including autonomous actions, human intervention actions, and the final trained policies. Using these logs together with the task-specific reference functions described in Section 4.1, we analyze real human interventions. Our results show that humans tend to intervene when robot progress stalls, and that their corrective actions are often suboptimal and diverge from the converged policy.

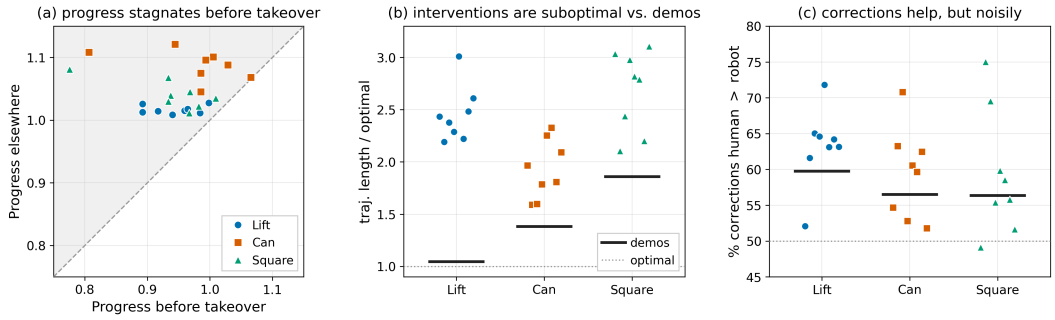


Figure 15: Per-participant results. The study has 8 participants and ∼2k trajectories (∼200k transitions) across 3 tasks; the findings hold for every participant and task. (Finding 1) Progress before takeovers is lower than elsewhere, i.e., humans intervene as the robot stalls. (Finding 2) Relative to optimal trajectory length, interventions are costlier than demonstrations for every participant. (Corrections) The human action has higher Q^* than the robot action in 51.8–69.9% of corrections, i.e., corrections help but are noisy, motivating their use as soft guidance.

D.2 Comparing algorithm performance across synthetic and real interventions

In Section 4.1, we observe that human supervisors intervene when they perceive the robot's task progress to be insufficient and that the corrections they provide are suboptimal. We next assess whether our progress-aware synthetic intervention model can act as a reliable proxy for real human supervision when benchmarking robot learning algorithms. To obtain a real-human reference, four participants supervised online training of six methods, `Q-OIL`, `Q-OIL` w/o BC, BC+RL, HG-DAgger, HIL [34] and RLIF, on the Lift and Can tasks with randomized robot initial positions. We trained each method for 8,000 environment steps on Lift and 12,000 environment steps on Can. We report the resulting success rates across training checkpoints in Figure 7 and rank the methods by their final success rates in Figure 17. We replicate the evaluation with three synthetic intervention models: our progress-based gating model, an intervention model based on value/action suboptimality, and a random intervention model tuned to match the observed human intervention frequency. For each synthetic model, we run every learning algorithm with multiple random seeds and present the resulting success-rate curves in Figure 18. For every intervention source, we order the learning algorithms by their final success rate and then evaluate how well the synthetic orderings correlate with the real-human ordering using Spearman rank correlation. As shown in Figure 4, our progress-based model attains perfect rank correlation with the real-human results. In other words, it reproduces the same algorithm ranking as real users, suggesting it more faithfully maintains the algorithm-level takeaways one would obtain from actual human-in-the-loop studies.

E Real-World Setup

Our real-world setup follows the HIL-SERL system of Luo et al. [34]. We use a robot manipulator equipped with a parallel-jaw gripper and train vision-based policies directly on the physical robot. RGB cameras placed around the workspace provide visual observations. For each task, the robot policy uses a wrist camera for observations. For tasks that require broader scene context, we also use an external side-view camera. The observations also include proprioceptive information, including the current end-effector state and gripper state.

Following Luo et al. [34], we implement a hybrid action space: a Gaussian policy outputs 3-DoF Cartesian end-effector position deltas (x, y, z), while a separate discrete critic selects the binary gripper command. We remove end-effector rotations to simplify teleoperation, and clip all commanded motions to a safety bounding box. We also synchronize the learner and actor loops to achieve a UTD ratio of 10 across all methods.

For each task, we first collect a small set of SpaceMouse demonstrations to initialize the demonstration replay buffer. During online training, the learned policy controls the robot by default. At the same time, the supervisor can intervene with the SpaceMouse when the robot is likely to fail, becomes stuck, or stops making progress. Intervention actions replace the policy actions on the robot and are logged for policy updates. The supervisor also manually resets each episode by randomizing the relevant objects and robot configuration within valid task ranges. We train each real-world task for up to 10k environment steps and evaluate checkpoints on a fixed set of reset conditions. The tasks are described in Appendix G. **We include all videos of training and evaluation on our project website:** `https://weirdlabuw.github.io/qoil`**.**

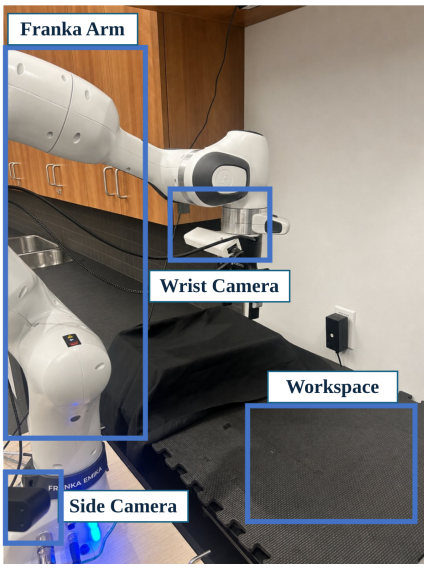


Figure 16: Real-world robot setup. The policy receives RGB observations from a wrist-mounted camera and, optionally, a side-view camera. The human supervisor monitors the robot during online training and provides interventions using a SpaceMouse.

E.1 Extending `Q-OIL` to Pretrained VLAs

Task and Robot Setup. We evaluate our method on two bimanual tasks. In bag assembly, the robot must use two arms to pick up a bottle, stretch the side pocket of a backpack, and insert the bottle accurately. In coffee pouring, which mimics a wet-lab task, the robot must pick a test tube filled with coffee beans from a rack and pour the beans into a beaker. We use the I2RT YAM bimanual robot with a leader-follower setup, where the leader arms mirror the follower arms during autonomous learning, allowing the human supervisor to seamlessly take over the leaders and intervene when needed. We use joint-space control and RGB observations from an external scene camera. We first collect a small set of human demonstrations to fine-tune the pretrained VLA, then freeze it during online learning.

Online Training. During online training, the robot executes actions from the pretrained VLA with learned residual corrections by default. The human supervisor can take over the leader arms when the robot gets stuck or fails to make progress, and these corrective actions update the residual policy online. We manually reset the bottle and backpack between episodes.

Residual Policy Learning. For each task, we first fine-tune a pretrained VLA on 25 human demonstrations and use the resulting policy as a frozen base policy. Following EXPO [44], we learn a residual policy that corrects the VLA's predicted actions. The base VLA predicts action chunks of length $H_b = 32$, and the residual policy predicts corrections over shorter chunks of length $H_r = 8$. For each base action, the residual policy is queried multiple times using updated RGB observations, allowing the policy to make closed-loop corrections at a higher frequency without repeatedly querying the base VLA. We compose the executed residual action chunk as

$$A_t = A_t^{\text{base}} + \eta \pi_\theta^{\text{res}}(o_t, A_t^{\text{base}}), \tag{2}$$

where A_t^{base} is the corresponding H_r actions from the base VLA and η controls the magnitude of the residual actions. We apply per-dimension scaling to account for the different limits of joint and gripper actions. This enables the residual policy to locally improve the pretrained VLA's behavior without updating its parameters.

Algorithm 2 Residual Q-OIL for Pretrained VLAs

Require: Frozen VLA π_{base}, residual policy π_θ, human π^{int}, buffers D_π, D_{int}, bonus b, chunk sizes H_b, H_r

for trial $i = 1$ to N **do**
  // Online rollout
  for timestep $t = 1$ to T in steps of H_r **do**
    if t starts a new base chunk **then**
      $A^{\text{base}} \leftarrow \pi_{\text{base}}(o_t)$
    end if
    $\Delta A_t \leftarrow \pi_\theta(o_t, A_t^{\text{base}})$
    $\hat{A}_t \leftarrow A_t^{\text{base}} + \eta \Delta A_t$
    Execute $\hat{A}_t$ for H_r steps, allowing human takeover
    for transition (o_s, a_s, r_s, o_{s+1}) in chunk **do**
      if π^{int} intervened at s **then**
        $r_s^b \leftarrow b$
        Add (o_s, a_s^{int}) to D_{int}
      else
        $r_s^b \leftarrow 0$
      end if
      Add $(o_s, a_s, r_s, r_s^b, o_{s+1})$ to D_π
    end for
  end for
  // Decoupled critic updates
  Train Q_{TD} on D_π via $\mathcal{L}_{\text{TD}}$
  Train Q_{Opt} on D_π via $\mathcal{L}_{\text{Opt}}$
  // Residual policy improvement
  Update π_θ using Q_{Opt} and BC on D_{int}
end for

Learning from Interventions. We use residual SAC with BC regularization on intervention actions as our baseline. To extend Q-OIL, we maintain two critics over the executed actions: a task critic Q_{TD} trained on the environment reward and an optimistic critic Q_{Opt} trained with an additional positive reward on intervention transitions. Importantly, the optimistic critic bootstraps from the task critic rather than itself, preventing the intervention bonus from propagating to preceding states. We update the residual policy to maximize Q_{Opt}, with an auxiliary BC loss on human corrections. The base VLA remains frozen throughout online training. We use a UTD ratio of 2 for both Q-OIL and BC+RL. Algorithm 2 summarizes this procedure.

Evaluation. For bag assembly, we evaluate the learned policies on three sets of 10 episodes each and report both bottle pickup and full task completion rates. For coffee pouring, we evaluate on 10 episodes and report handover success and full task success, where at least half of the coffee beans are poured into the beaker. We additionally measure human effort during online adaptation as the fraction of control steps under intervention. For bag assembly, the BC+RL and Q-OIL training runs include 36 and 17 takeovers, corresponding to intervention fractions of 25.2% and 18.1%, respectively; for coffee pouring, the intervention fractions are 23.8% and 10.7%.

F Synthetic Intervention Models and Rank-Correlation Analysis

F.1 Parameter Selection for the Progress-Aware Model

Our synthetic model predicts whether an intervention should occur based on recent task progress. Its two main parameters are the window length k, which sets the time horizon over which we assess progress, and the threshold δ, which specifies when we consider progress stalled; we set the intervention probabilities α, β based on empirical intervention rates observed in the human data. We tune these parameters using held-out human intervention data. Concretely, we run a grid search over $k \in \{1, 2, 3, \ldots, 32\}$ and $\delta \in [-0.5, 0.5]$, and choose the setting that maximizes F1 score for

binary intervention prediction. We optimize F1 over accuracy because interventions are sparse: a model can obtain high accuracy by predicting not to intervene always, but such a model is not useful for simulating human supervision. The F1 objective balances precision and recall, encouraging the model to identify true intervention states without triggering excessively. We fit a single set of parameters across all tasks and participants, rather than tuning separate parameters per task, so that the resulting model captures general task-agnostic intervention behavior and can transfer to new tasks. We then fix the selected parameters for all simulated intervention experiments.

F.2 Held-Out Intervention Prediction

Table 1 reports intervention prediction performance on held-out human data. The random model is calibrated to the empirical human intervention rate. The action-suboptimality model triggers interventions based on whether the robot action is suboptimal under a learned value estimate. The progress-aware model achieves the best precision and recall, indicating that recent lack of task progress is a better predictor of when humans intervene than either random triggering or instantaneous action suboptimality.

Table 1: Intervention prediction on held-out human data.

Method	Precision	Recall
Action Suboptimality	0.1136	0.4489
Random	0.0837	0.0837
PAM (Ours)	**0.1750**	**0.6059**

F.3 Rank Correlation with Real-Human Evaluation

We next evaluate whether synthetic intervention models preserve the same algorithm-level conclusions as real human supervisors. To obtain the real-human reference ranking, participants supervised online training of the evaluated algorithms as discussed in Appendix D.2. We ranked the algorithms by final success rate. Figure 17 shows the rankings used for the Spearman rank-correlation analysis in Figure 4. A Spearman correlation of $\rho = 1.0$ means that the synthetic intervention model produces the same ordering of algorithms as real-human supervision, even if the absolute success rates differ.

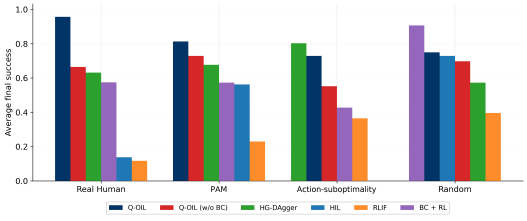


Figure 17: Rankings of algorithms under real human evaluation and three simulated intervention models, based on final success rate.

F.4 Synthetic Intervention Rollouts

In each synthetic evaluation, the intervention model determines when the simulated supervisor intervenes. Once an intervention is triggered, the simulated supervisor provides corrective control for a sampled intervention duration. We sample these durations from the lower 75% of the empirical human intervention-length distribution, and actions are generated from a suboptimal RL checkpoint with $\sim$50–70% success rate matching empirical success rates from real human intervention experiments. This design choice ensures simulated interventions are suboptimal (matching our analysis in Section 4.1) while still providing useful directional guidance.

F.5 Algorithm Performance under Different Synthetic Models of Human Intervention

We evaluate all learning algorithms under the synthetic intervention models described in Section 4. Figure 18 shows success rates for the PAM, Random, and Action-suboptimality models. For each intervention model, we run each learning algorithm across multiple random seeds and report success

rate as a function of environment steps. The results show that the choice of intervention model can substantially affect conclusions about baseline performance. In particular, the relative ordering of the baselines changes across intervention models: HIL improves under Random interventions but performs worse under Action-suboptimality, while HG-DAgger is favored by the Action-suboptimality model. In contrast, PAM matches the real-human ranking (from both simulated and real robot tasks), suggesting that modeling when humans intervene based on stalled progress better preserves the conclusions obtained from human-in-the-loop experiments.

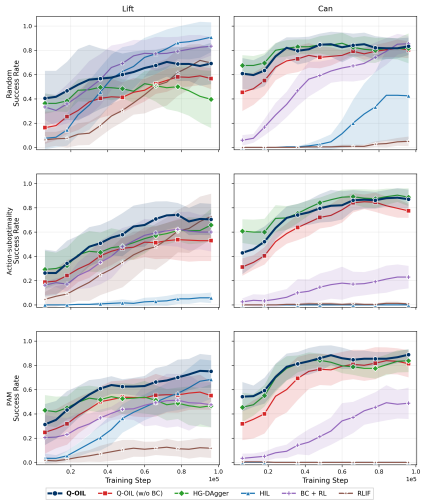


Figure 18: Success rate vs. env steps under different intervention models

G Task Details

We evaluate `Q-OIL` on a set of real-world and simulated manipulation tasks that require learning from sparse rewards with human interventions (see Figure 19).

Pen-in-Bowl. A Franka arm must pick up a pen and drop it into a fixed bowl using side- and wrist-camera observations. At the start of each episode, the pen pose is randomized within a bounded region. The task uses a sparse reward, $r = 1$, when the pen is detected inside the bowl, with the reward learned using a classifier as in Luo et al. [34]. Episodes have a horizon of 200 environment steps. We evaluate on 20 fixed pen positions. This task benefits from higher interaction throughput: when the robot initially fails to place the pen in the bowl, it can often recover by regrasping the pen and completing the task within the same episode.

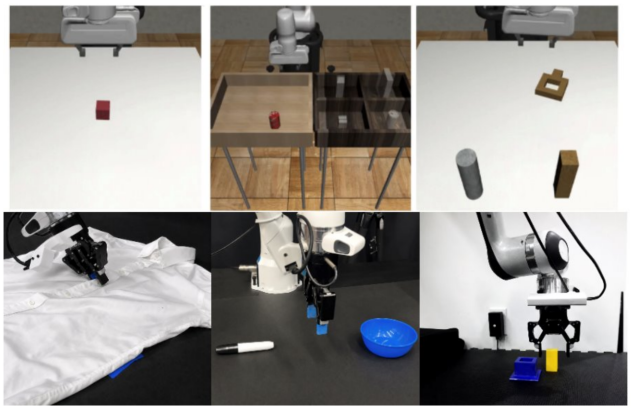


Figure 19: Simulated Robomimic tasks (top) and Franka tasks (bottom). The bimanual YAM tasks are shown in Figure 9.

Unbutton Shirt. A robot must manipulate a deformable shirt placed on a table and open a button using wrist-camera observations. To simplify grasping, we fix the roll, pitch, and yaw of the end effector. The task uses a sparse success reward, $r = 1$, upon successfully unbuttoning the shirt. We evaluate over 10 episodes.

Peg Insertion. A robot must insert a peg into its corresponding hole in a 3-DoF setup using wrist-camera observations. The initial state is randomized at reset, and the task uses a sparse success reward. We evaluate on 10 fixed test positions.

Bimanual YAM Tasks. The bag assembly and coffee pouring tasks are described in Appendix E.1.

Robomimic Simulation. We also evaluate on the Can, Square, and Lift tasks from Robomimic. Simulation experiments use a 7-DoF robot, while human-intervention experiments use a 4-DoF control interface. Initial states are randomized over the default environment ranges. Policies are evaluated over 100 episodes, and simulation results with synthetic intervention models are reported over three random seeds.

H How does quality of expert actions affect learning?

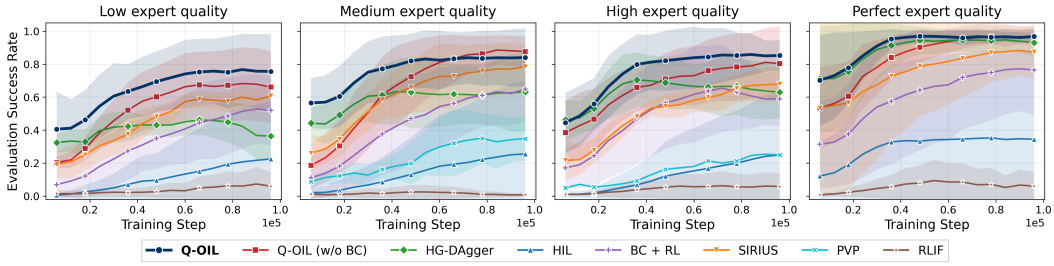


Figure 20: Success across takeover-quality tiers, averaged over Lift, Can, and Square (3 seeds). Tiers group intervention checkpoints by per-task success rate (Low: 42/44/58%, Medium: 60/80/72%, High: 68/94/82%, Perfect: 100%).

In Figure 20, we ablate the quality of the corrective actions by grouping intervention policies into four tiers by success rate. With perfect experts (100% success rate), most methods converge to a high success rate efficiently, and imitation-based baselines (e.g., HG-DAgger) are competitive as they receive near-optimal corrections. As expert quality decreases, `Q-OIL` (with or without BC) remains the best-performing method, with the largest margins under the weakest experts, while other baselines degrade as they inefficiently use noisy and suboptimal corrections. This aligns with our observations across experiments with real humans and real robots. Overall, `Q-OIL` depends on intervention quality, but less than the baselines, making it practical for real-world deployment where intervention quality varies.

I Implementation Details

I.1 Hyperparameters

All methods use the same demonstrations, reward definitions, actor–critic architecture, replay buffer size, evaluation episodes, and intervention interface unless otherwise stated.

Table 2: Hyperparameters for Q-OIL and baselines.

Parameter	Value
Core RL Parameters	
Architecture	MLP with Gaussian Head
Hidden layers	3 layers of width 256
Optimizer	Adam
Learning rate	3e-4
Discount γ	0.99
Soft update τ	0.005
UTD Ratio	10
Batch size	256
Replay buffer size	1e6
Learnable temperature	False
Q-OIL-Specific	
Optimism bonus b	0.025
Critic ensemble size (each of TD / Opt)	10
BC regularization weight	0.1
Intervention Model	
Stagnation window k	9
Stagnation threshold δ	-0.158
Intervention probability α	0.599
Intervention probability (no stagnation) β	0

Table 3: Hyperparameters for residual VLA learning with Q-OIL and BC+RL.

Parameter	Value
Base Policy and Residual Architecture	
Base policy	Pretrained VLA (frozen)
Base policy demonstrations	25
Base policy fine-tuning steps	20k
Residual policy	Gaussian MLP + ResNet-10
Actor / critic hidden layers	2 layers of width 256
Action space	Bimanual joint space (14-DoF)
Base action chunk size H_b	32
Residual action chunk size H_r	8
Residual Policy Parameters	
Residual joint scale	0.02 rad
Residual gripper scale	0.10
Initial edit scale	0.25
Final edit scale	1.0
Edit scale step size	0.05
Core RL Parameters	
Algorithm	Residual SAC [44]
Optimizer	Adam
Learning rate	3e-4
Discount per control step	0.97
Soft update τ	0.005
UTD Ratio	2
Batch size	256
Critic ensemble size	2
BC regularization weight	0.1
Initial temperature	0.01
Training starts (transitions)	200
Q-OIL-Specific	
Optimism bonus b	0.1
Additional optimistic critic	Yes
Optimistic critic bootstrap	Task critic