Real-world Reinforcement Learning from Suboptimal Interventions

Yinuo Zhao [1 2] **Huiqian Jin** [1 3] **Lechun Jiang** [1 3] **Xinyi Zhang** [1 4] **Kun Wu** [1] **Pei Ren** [1] **Zhiyuan Xu**[† 1] **Zhengping Che**[† 1] **Junjie Ji** [1] **Lei Sun** [3] **Dapeng Wu** [2] **Chi Harold Liu** [4] **Jian Tang**[✉ 1]

Abstract

Real-world reinforcement learning (RL) offers a promising approach to training robotic manipulation policies through online interaction. While recent methods leverage human interventions to accelerate learning, they often assume interventions are consistently optimal or rely on offline filtering mechanisms that may discard valuable exploratory data. In practice, human operators exhibit varying performance across different states: operators may provide near-optimal guidance in familiar situations but struggle in novel or ambiguous states. The key challenge is *how to selectively leverage heterogeneous intervention quality across states while maintaining the benefits of online exploration*. To address this, we propose **SiLRI**, a state-wise Lagrangian RL algorithm that adaptively trades off between imitating interventions and maximizing future returns. We formulate online learning as a constrained optimization problem where constraint bounds vary across states according to estimated intervention uncertainty. This problem is then solved via state-wise Lagrangian relaxation, enabling the policy to selectively imitate interventions in high-confidence regions while relying more on RL exploration elsewhere. We evaluate **SiLRI** on nine real-world manipulation tasks using a human-as-copilot teleoperation system. Compared to HIL-SERL that treats interventions equally, SiLRI achieves at least 50% faster learning, effectively exploiting suboptimal human interventions without being constrained by them. Project website: SiLRI.

[1]Beijing Innovation Center of Humanoid Robotics, Beijing, China [2]City University of Hong Kong, Hong Kong, China [3]Nankai University, Tianjin, China [4]Beijing Institute of Technology, Beijing, China. [✉]Correspondence to: Jian Tang <jian.tang@x-humanoid.com>. [†]Project Leader.

Workshop on RLxF: Reinforcement Learning from World Feedback.

1. Introduction

Human-in-the-loop reinforcement learning (HIL-RL) has emerged as a promising paradigm for real-world robotic manipulation, combining the autonomous improvement of reinforcement learning (RL) with the strategic guidance of human-in-the-loop supervision (Luo et al., 2025; Chen et al., 2025). By mixing online policy transitions with timely human interventions, HIL-RL alleviates the safety and exploration burdens that hinder pure RL on physical hardware, while avoiding the high cost of collecting large-scale expert demonstrations required by imitation learning (Zhao et al., 2023; Chi et al., 2023; Xu et al., 2025; Wu et al., 2024). This hybrid strategy has enabled near-human performance on a range of manipulation tasks.

Within this context, human intervention data plays a pivotal role in shaping policy behavior. Existing HIL-RL methods consume this data in a uniform manner—either directly imitating all intervention actions via behavior cloning, or indiscriminately injecting them into the RL objective as off-policy transitions. Such uniform treatment, however, overlooks the inherent heterogeneity of intervention quality. In practice, intervention quality varies considerably across different states, probably due to differences in manipulation difficulty, task familiarity, and human factors such as fatigue. Treating all intervention transitions equally inevitably wastes the most informative samples while overfitting to the noisiest ones, leading to poor sample efficiency.

To illustrate this inconsistency, we analyze a *pick-then-hang* task (Hang Chinese Knot) by estimating intervention entropy along trajectories using a multivariate normal distribution fitted on 20 demonstrations. As shown in Fig. 1, action consistency varies clearly across states: during picking phases, human actions are highly consistent (low entropy); in contrast, phases requiring fine adjustments or repeated corrections exhibit notably higher entropy. This state-dependent variation in intervention quality motivates a learning strategy that adapts to local data quality rather than treating all transitions uniformly.

To realize this behavior, we propose **SiLRI**, a State-wise Lagrangian Reinforcement learning framework for learning from suboptimal Interventions. Specifically, we introduce a state-wise Lagrange multiplier before a behavior cloning (BC) objective and formulate the problem as a min-

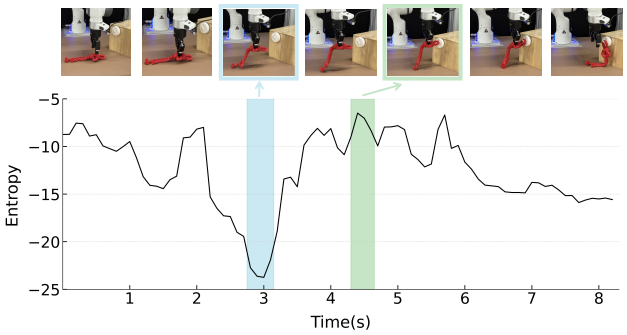

Figure 1. **Entropy of Human Interventions Across States.** Entropy is estimated with a multivariate normal distribution model. In low-entropy states (blue region), human operators intervene consistently and confidently, whereas in high-entropy states (green region), their interventions are inconsistent, indicating uncertainty.

max optimization. The policy objective is minimized while the dual objective over the Lagrange multiplier is maximized, enabling the state-wise multiplier to be adaptively adjusted to balance the RL and BC objectives. This design improves RL sample efficiency while ensuring convergence to a policy beyond suboptimal intervention data.

Using a bilateral teleoperation system, we conduct extensive real-world experiments across three robotic platforms. Compared to HIL-SERL (Luo et al., 2025), SiLRI requires at least 50% less training time to achieve over 90% success rate. Compared to classical online IL methods, our method achieves 20%-86% higher success rates against external disturbances. In the Open Cabinet task, SiLRI completes tasks in half the time required by human operators. Ablation studies further validate the effectiveness of SiLRI.

2. Related Works

Real-world Reinforcement Learning RL from suboptimal datasets has been widely studied, with offline RL being a crucial branch. When faced with mixed-quality data, offline RL methods (Kostrikov et al., 2022; Nair et al., 2020; Kumar et al., 2020) aim to learn a policy that improves over the behavior policy that collected the dataset while minimizing deviation to avoid Q overestimation. Offline RL poses minimal requirements on dataset quality, making it well-suited for offline pretraining. However, its unstable performance and offline-to-online gap limit its application in real-world online training.

Real-world RL demands high sample efficiency and safety, making human interventions crucial to incorporate. This approach is also called Human-in-the-loop reinforcement learning (HIL-RL). Recent HIL-RL methods like HIL-SERL (Luo et al., 2025) and ConRFT (Chen et al., 2025) have achieved near-human performance on highly dexterous manipulation tasks. *Their success stems from sampling from two buffers (Ball et al., 2023)*: an online buffer storing transitions from the current policy plus human interventions, and an intervention buffer storing only offline demonstrations and online human interventions. However, sample efficiency drops dramatically when facing larger exploration spaces. To address this, ConRFT (Chen et al., 2025) utilizes large foundation models, combining behavior cloning (BC) and Q-learning in a consistency policy for both offline and online training. Nevertheless, the BC term can limit policy improvement in later training stages. Several works propose leveraging suboptimal intervention data. For example, MILE (Korkmaz & Biyik, 2025) and RLIF (Luo et al., 2023) rely on intervention signals rather than direct actions to improve sample efficiency. Our method directly utilizes intervention actions but introduces a state-wise Lagrange multiplier to surpass intervention quality limitations.

Recently, an increasing number of studies propose finetuning large foundation models with RL (Guo et al., 2025; Xia et al., 2025). For example, iRe-VLA (Guo et al., 2025) alternates between supervised learning (SL) and RL for stable fine-tuning. In the SL stage, the VLM backbone and action head are trained on successful trajectories using LoRA (Hu et al., 2022). In the RL stage, only lightweight action and critic heads are fine-tuned for efficiency. These works still treat human interventions as optimal guidance and incorporate the corresponding data via an imitation loss term. In contrast, our work follows HIL-SERL, using a pretrained ResNet10 (He et al., 2016) visual backbone and small visuomotor network architectures, to focus on algorithmic improvements for reinforcement learning.

Lagrange Multipliers Lagrange multipliers are common in RL, yet few methods optimize them explicitly, relying instead on heuristic or fixed estimates. For example, the classical PPO algorithm (Schulman et al., 2017) imposes a trust-region-style constraint on the distance between the updated and old policy to avoid policy collapse. Due to the unbounded nature of both the Lagrange multiplier and the KL divergence, PPO replaces explicit multiplier optimization with a clipped surrogate objective. In offline RL, constraining the policy to remain close to the behavior policy is a common way to prevent bootstrapping on OOD actions. In AWAC (Nair et al., 2020), for example, the optimization objective is simplified to a forward KL form with a heuristic temperature (multiplier) parameter. Other works directly optimize Lagrange multipliers. PPO-Lagrangian extends PPO to a constrained RL setting, while other methods (Seo & Choi, 2025) introduce state-dependent multipliers so that the policy can account for state-wise safety constraints. Our work builds on these studies by introducing state-wise Lagrange multipliers into HIL-RL.

3. Preliminaries

Following (Luo et al., 2025), we model the MDP in robotic manipulation as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{I}, \rho, \mathcal{P}, \mathcal{R}, \gamma)$. The state $\boldsymbol{s} \in \mathcal{S}$ comprises multiple RGB images together with the robot's proprioceptive signals. The action $a \in \mathcal{A}$ (e.g., a desired end-effector 6D movement and the gripper status) is given either by a learned policy π or by a human via a teleoperation interface. An intervention indicator $I(s, a) \in \mathcal{I}$ marks the action's source, with $I(s, a) = 1$ for human intervention and $I(s, a) = 0$ for autonomous control. $\rho(\boldsymbol{s}_0)$ denotes the initial-state distribution; $\mathcal{P}$ denotes the environment's state-transition dynamics; and the reward function $\mathcal{R}$ maps $\mathcal{S} \times \mathcal{A}$ to scalar rewards. Finally, γ is the discount factor that trades off immediate and future returns, where larger γ emphasizes long-term rewards.

In real-world RL, reward signals are typically sparse: the agent receives positive feedback only upon successful task completion. In this work, we follow prior approaches (Luo et al., 2025; Chen et al., 2025) by providing a reward of +10 only upon successful task completion, while applying a penalty of -0.05 at every other step. Some baselines additionally include gripper or intervention penalties in their reward functions to match their original implementations. Detailed reward settings for all methods can be found in Appendix D.5.

Previous RL methods simply treat the learning process as an typical unconstraint optimization problem and the objective $J(\pi)$ is defined as:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{T} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right]. \tag{1}$$

Although the above optimization objective is compatible with suboptimal human intervention data, it overlooks the potential optimality embedded in such data. In states where humans can provide reliable interventions, simple behavior cloning offers more effective guidance than a Q-network. Conversely, in other states, a well-learned Q-network can guide the policy beyond suboptimal human behavior, enabling performance that surpasses human capability. Identifying these states and automatically balancing the BC and RL objectives is therefore crucial for efficient real-world training. In the following section, we present a constrained optimization formulation and its solution.

4. Methodology

In this section, we present SiLRI, State-wise Lagrangian Reinforcement learning from suboptimal Interventions, for real-world robot manipulation training. We begin by formulating a constrained optimization problem. Then, state-wise Lagrange multipliers are introduced to convert the constraint problem as a min-max optimization. Next, we detail the loss functions and network architectures employed in SiLRI to enable efficient and stable learning. Last, we summarize the overall human-in-the-loop (HIL) training paradigm. Other details on critical design decisions and parameter settings are provided in Appendix C.

4.1. Problem Formulation

Recall the unconstrained objective in Eq. (1), directly optimizing the policy towards increasing Q-values can be unreliable due to inaccurate value estimation, especially during the early stages of training. To address this challenge, we reformulate a constrained optimization problem as follows:

$$\begin{aligned} \max_{\pi} \quad & J(\pi) \\ \text{s.t.} \quad & \mathrm{D}\big(\pi(\cdot|\boldsymbol{s}) \| \beta(\cdot|\boldsymbol{s})\big) \leq \epsilon, \qquad \forall \boldsymbol{s}, \end{aligned} \tag{2}$$

where π denotes the deterministic robot policy, and β represents the reference policy derived from human interventions. While multiple trajectories may accomplish a given task, most robotic manipulation tasks admit a single optimal solution that minimizes execution time and maximizes success probability. For example, when inserting a peg into a hole, the optimal policy determines a unique approach angle, insertion speed, and corrective motion sequence at each state. Therefore, we model the optimal manipulation policy as deterministic, mapping each state to the single best action. In contrast, human operators inherently exhibit variability in continuous control, so we model the human reference policy β as stochastic. The constraint is designed to prevent the learned policy π from deviating excessively from β, which could otherwise lead to unrecoverable or unsafe behaviors.

Two key challenges arise when optimizing the above objective. First, measuring the distance between a deterministic policy π and a stochastic policy β is nontrivial. Second, even if we treat π as a stochastic policy and evaluate the Kullback-Leibler (KL) divergence between these two, empirical optimization remains difficult due to the unbounded nature of the KL divergence. To address these issues, we reformulate the constraint as follows:

$$\begin{aligned} \max_{\pi} \quad & J(\pi) \\ \text{s.t.} \quad & \parallel \mu(\pi(\cdot|\boldsymbol{s})) - \mu(\beta(\cdot|\boldsymbol{s})) \parallel \leq \kappa \cdot \sigma_\beta(\boldsymbol{s}), \qquad \forall \boldsymbol{s}, \end{aligned} \tag{3}$$

where μ denotes the output of π and the mean of β, σ_β is the standard deviation of policy β, and κ is a constant value that adjust the tightness of the constraint. In Appendix B, we proof that under mild assumptions, the constraint condition in Eq. (2) can be simplified into the constraint condition in Eq. (3). For simplicity, in the following we omit the symbol μ and use π and β to denote the outputs (i.e., means) of the corresponding policies.

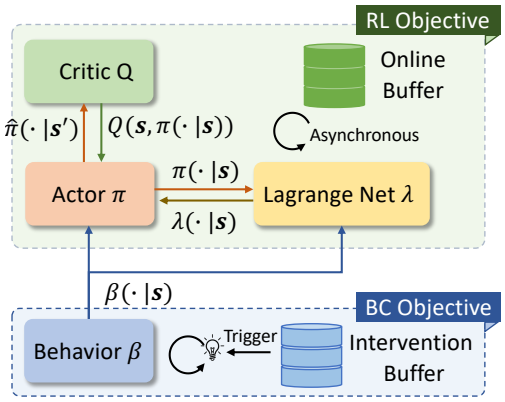

Figure 2. **Network Components in SiLRI.** The networks Q, π, and λ are updated asynchronously during data collection using online buffer, while β is updated periodically after a fixed number of new samples have been added to the intervention buffer.

The constraint in Eq. (3) enforces that the optimized policy π should remain close to the reference policy β in states where humans provide consistent intervention data. In contrast, for states where human behavior is more uncertain (i.e., where the reference policy exhibits high variance), the policy π is allowed to deviate and explore more freely, optimizing through RL under a relaxed constraint.

4.2. State-wise Lagrange Multipliers

A common approach to solving constrained optimization problems is to introduce a Lagrange multiplier $\lambda \geq 0$ that penalizes constraint violations. For the optimization problem in Eq. (3), the corresponding Lagrangian is

$$\mathcal{L}(\pi, \lambda) = -J(\pi) + \lambda^\top \left[\mathrm{D}(\pi, \beta) - \kappa \Sigma_\beta\right], \tag{4}$$

where λ is a state-wise Lagrange multiplier, i.e., a nonnegative function over states rather than a scalar value. $\mathrm{D}(\pi, \beta)$ represents the Euclidean distance in Eq. (3) and Σ_β denotes the standard deviation of β. The associated Lagrange dual problem can be written as

$$\sup_{\lambda} \inf_{\pi} \mathcal{L}(\pi, \lambda), \quad \text{s.t. } \lambda(\cdot) \geq 0. \tag{5}$$

The objective is to find a saddle point (π^*, λ^*) that satisfies

$$\mathcal{L}(\pi, \lambda^*) \geq \mathcal{L}(\pi^*, \lambda^*) \geq \mathcal{L}(\pi^*, \lambda) \tag{6}$$

for all admissible π and λ. Specifically, for the policy π and a fixed λ, the optimization problem becomes

$$\inf_{\pi} \left[-J(\pi) + \lambda^\top \mathrm{D}(\pi, \beta)\right], \tag{7}$$

since Σ_β does not depend on π. Conversely, for the Lagrange multiplier λ and a fixed π, the optimization problem is formulated as

$$\sup_{\lambda} \lambda^\top \left[\mathrm{D}(\pi, \beta) - \kappa \Sigma_\beta\right], \quad \text{s.t. } \lambda(\cdot) \geq 0. \tag{8}$$

Since finding the global saddle point is computationally intractable, we adopt gradient-based updates with function approximators. However, there are two challenges related to the behavior policy β when solving the above objectives. First, the true policy distribution of the human operator is unavailable. Second, human intervention data is unavailable for states visited exclusively by the learned policy π. Therefore, we cannot directly measure the distance between π and β without sampling from the behavior policy. In this work, we approximate β with a multivariate Gaussian distribution fitted on the human intervention data. In the following, we describe the network architectures and loss functions used to approximate the optimal solution.

4.3. Network Architectures and Loss Functions

As shown in Fig. 2, there are four networks to approximate, they are the actor network π, behavior policy β, Lagrange network λ and critic network Q. To improve training stability, we adopt a target actor network and a double Q-network architecture, following common practice in RL.

Critic optimization We follow the standard Bellman update and optimize the critic Q using a temporal-difference objective:

$$\mathcal{L}(\theta^Q) = \mathbb{E}_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}'}\Big[\big(r + \gamma \hat{Q}(\boldsymbol{s}', \hat{\pi}(\boldsymbol{s}')) - \min_{k=1,2} Q_k(\boldsymbol{s}, \boldsymbol{a}; \theta^Q)\big)^2\Big]. \tag{9}$$

where θ^Q denotes the parameters of the critic Q, and $\hat{\pi}$ and $\hat{Q}$ are the target policy and target critic, respectively. Both target networks are updated via Polyak averaging (Van Hasselt et al., 2016) from the current networks.

Actor optimization The actor network is modeled as a multivariate Gaussian policy whose mean is passed through a Tanh activation to constrain the actions to $(-1, 1)$. During inference, when we need to sample actions from the actor, we add a constant standard deviation of 0.05 to encourage sufficient exploration and apply clipping to avoid out-of-range samples. During training, recalling the constrained optimization problem in Eq. (7), the actor policy π is optimized with a joint objective that combines RL and BC, balanced by the self-adapting coefficient λ. The actor loss is defined as

$$\mathcal{L}(\theta^\pi) = \mathbb{E}_{\boldsymbol{s}} \frac{1}{\lambda(\boldsymbol{s}) + 1}\Big[-Q(\boldsymbol{s}, \pi(\boldsymbol{s}; \theta^\pi)) + \lambda(\boldsymbol{s})\big\|\pi(\boldsymbol{s}; \theta^\pi) - \beta(\boldsymbol{s})\big\|\Big], \tag{10}$$

where $\|\cdot\|$ denotes the Euclidean norm over the action space, and $\lambda(\boldsymbol{s})$ is the state-wise Lagrange multiplier. Following prior work (Seo & Choi, 2025), the division $\lambda(\boldsymbol{s})+1$

acts as a regularizer to mitigate instability caused by large λ values. Note that when $\lambda(\boldsymbol{s}) \to \infty$, the RL term is suppressed and the loss is dominated by the BC term, forcing the policy to follow the behavior policy. Conversely, when $\lambda(\boldsymbol{s}) \to 0$ and the constraint is satisfied, the RL term dominates, allowing the learned policy π to improve beyond the behavior policy under the RL objective.

Lagrange Multiplier Optimization The Lagrange multiplier network takes the current state as input and outputs a scalar value that modulates the weight of the constraint term, as defined in Eq. (4). To ensure that the multiplier remains non-negative, a Softplus activation function is applied at the network's output layer. The loss function used to train the Lagrange multiplier is given by:

$$\mathcal{L}(\theta^{\lambda}) = \mathbb{E}_{\boldsymbol{s}}\left[-\lambda(\boldsymbol{s};\theta^{\lambda})\left(\mathrm{D}(\pi,\beta) - \kappa\cdot\sigma_{\beta} - c\right)\right], \quad (11)$$

where $\mathrm{D}(\pi,\beta) = \|\pi(\boldsymbol{s};\theta^{\pi}) - \beta(\boldsymbol{s})\|$ denotes the Euclidean distance between the mean actions of the learned policy π and the reference policy β, consistent with that used in the actor optimization. Here, σ_{β} represents the average standard deviation over the continuous action dimensions of the multivariate Gaussian reference policy, and κ denotes the action dimensionality (i.e., $\kappa = 6$). Empirically, we add a constant $c = 0.1$ to relax the constraint. Intuitively, the Lagrange multiplier increases when the deviation between π and β exceeds the tolerance margin defined by $\kappa\cdot\sigma_{\beta}+c$, and gradually decays toward zero once the constraint is met.

Behavior Policy Optimization As shown in Fig. 2, the behavior policy β is optimized separately from the other three networks and updated at a lower frequency to maintain stable decision entropy (i.e., the standard deviation of the action distribution). Without such regulation, the entropy may decay to an extremely low value during training, which can hinder learning and reduce policy diversity. To address this, we update the behavior policy only when a fixed amount of new data is added to the replay buffer. The optimization objective of the behavior policy is given by:

$$\mathcal{L}(\theta^{\beta}) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\mathcal{D}_I}\left[-\log\beta(\boldsymbol{a}\mid\boldsymbol{s};\theta^{\beta})\right], \quad (12)$$

where the state-action pairs $(\boldsymbol{s},\boldsymbol{a})$ are sampled from the intervention buffer $\mathcal{D}_I$. This objective corresponds to standard behavior cloning and encourages β to maximize the likelihood of the observed human intervention actions.

4.4. Human-in-the-loop Training Paradigm

SiLRI builds on the asynchronous actor-critic architecture to enable a high update-to-data ratio. Before training, the human operator collects 20 demonstration trajectories for each task (30 for dual-arm tasks), which are stored in the intervention buffe $\mathcal{D}_I$. During training, at each actor timestep t, SiLRI stores a transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, I_t, r_t, d_t, \boldsymbol{s}_{t+1})$, where $I_t \in \{0,1\}$ indicates whether the action came from human intervention (1) or RL exploration (0), and $d_t \in \{0,1\}$ flags task success, which is given by a reward classifier. After each episode, the actor streams transitions to the learner and receives the actor policy parameters from the learner.

Algorithm 1 SiLRI (Learner)

```
Create empty intervention buffer D_I and Online RL buffer D_R.
Fill D_I with small-scale human demonstrations.
Create RL agent with critic network Q, actor network π, Lagrange network λ and behavior network β, with trainable parameters θ^Q, θ^π, θ^λ, θ^β, respectively.
Train β with data from D_I with Eq. (12).
Initialize n = 0
while LISTEN do
  if Receive transitions from Actor then
    for t in 1 ... T do
      Insert transitions into D_R and insert transitions (s_t, a_t, I_t = 1, r_t, d_t, s_{t+1}) into D_I.
      n = n + 1.
      if n mod 50 = 0 then
        for k in 1 ··· 50 do
          Minimize L(θ^β) in Eq. (12) with data from D_I.
        end for
      end if
    end for
  end if
  Sample a minibatch of data from D_R and D_I.
  Minimize L(θ^Q) in Eq. (9).
  Minimize L(θ^π) in Eq. (10).
  Minimize L(θ^λ) in Eq. (11).
  if Send Parameter Period then
    Send Parameters to the Actor.
  end if
end while
```

Algorithm 1 summarizes the learner-side training procedure. Before entering the main training loop, the behavior policy is pre-trained on data from $\mathcal{D}_I$ for 500 steps. During training, upon receiving online data from the actor side, all intervention samples are added to the intervention buffer $\mathcal{D}_I$, and all online samples, including both intervention and non-intervention data, are added to the online buffer $\mathcal{D}_R$. The behavior policy is updated whenever 50 new samples have been added into $\mathcal{D}_I$. After updating the buffers, we sample mini-batches of equal size 128 from $\mathcal{D}_I$ and $\mathcal{D}_R$, and then update the parameters θ^Q, θ^{λ}, and θ^{π} according to Eq. (9)-(11). The learner thread is terminated by the actor once the time limit is reached.

5. Experiments

In this section, we conduct real-world experiments to answer the following five research questions (**RQ**): (1) How much does SiLRI improve performance over SOTA reinforcement learning (RL) and imitation learning (IL) meth-

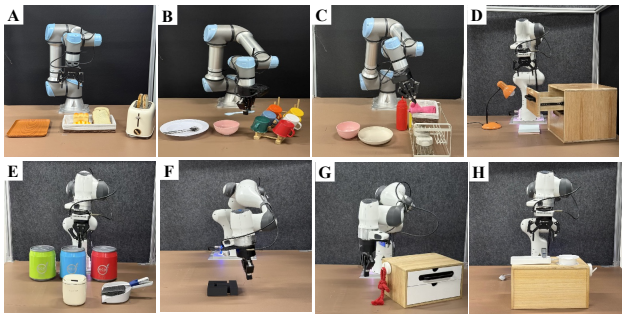

Figure 3. **Eight Evaluation Tasks.** We evaluate SiLRI on nine tasks across three embodiments, each equipped with a bidirectional teleoperation system that enables seamless human interventions during online training. The dual-arm task is in Appendix E.

ods? (2) How robust is the policy trained by SiLRI to external disturbances compared with other methods? (3) What is the contribution of each component (i.e., the Lagrange term, RL objective, and behavior cloning (BC) objective) to the performance of SiLRI? (4) How do the intervention quality affect the performance of different methods? (5) How does SiLRI perform when faced with operator's multimodal intents?

5.1. Experimental Setup

We conduct experiments on 3 robotic embodiments: UR5, Franka Emika, and a humanoid dual-arm robot. The state $\boldsymbol{s}$ consists of both visual and proprioceptive inputs. The visual inputs are captured by RGB cameras, and the proprioceptive input includes the absolute end-effector pose (in the robot base frame) and the gripper state. Detailed hardware setups can be found in Appendix D.1. Despite the difference in morphology, all embodiments adopt similar action space: a 6/12-DoF end-effector pose increment and, for some tasks, discrete gripper commands per timestep.

As shown in Fig. 3, we design 8 real-world manipulation tasks covering a range of skills: articulated-object manipulation, precision manipulation, deformable-object handling, and mixed-skill tasks. One additional task on the humanoid dual-arm robot is presented in Appendix E. These tasks pose challenges due to small interaction regions (e.g., spoon handle, USB port, hook), orientation constraints (e.g., articulated drawer, U-shaped groove), deformability (e.g., Chinese knot and cloth), and skill composition (e.g., reach-and-hang). Task details are in Appendix D.

The teleoperation system is based on a human-robot collaboration system HACTS (Xu et al., 2025) that bilaterally synchronizes the joint states of the robot arm and the operator device in real time. Human intervention signals are issued via a foot-pedal device. Besides, we adopt an asynchronous Actor-Learner training architecture: The Actor process runs on an GeForce RTX 4090 GPU server, while the Learner process runs on a RTX A6000 GPU server.

5.2. Baselines and Metrics

We compare SiLRI with five representative RL and IL methods as follows:

- **HIL-SERL** (Luo et al., 2025): integrates a pretrained vision backbone, sparse rewards, and the off-policy RLPD algorithm, leveraging both offline demonstrations and online data while learning from scratch.
- **ConRFT** (Chen et al., 2025): a two-stage reinforced fine-tuning framework for VLA models. In offline stage, it combines calibrated Q-learning (Nakamoto et al., 2023) with behavior cloning (BC) under a consistency policy. In online stage, it performs online fine-tuning with human interventions for safe and sample-efficient adaptation.
- **HG-Dagger** (Kelly et al., 2019): a classical online IL algorithm that learns solely from offline demonstrations and online interventions with BC loss.
- **RLIF** (Luo et al., 2023): An RL method leveraging suboptimal interventions. We modify the original implementation by adding success rewards (in addition to intervention and done signals) for faster convergence. Similar to HIL-SERL, a discrete DQN network is used to learn gripper control.
- **AWAC** (Nair et al., 2020): An efficient offline RL method. Similar to ConRFT, AWAC is pretrained offline using pre-collected demonstrations for 500 iterations before online deployment.

For a fair comparison, all online methods are trained within the same actor-learner asynchronous architecture, and all methods except ConRFT share identical network architectures and hyperparameter settings. ConRFT is evaluated using its original JAX implementation.

All methods are evaluated on three metrics:

- **Success Rate**: Unlike prior work measuring success on trajectories regardless of intervention, we only regard an episode as successful if the robot completes the task *without any human assistance*, to clearly reflect the performance of the learned policy itself. Concretely, the success rate is computed as the fraction of successful episodes within a fixed time window.
- **Intervention Ratio**: Human-intervention steps divided by total steps per episode during online training.
- **Episode Length**: Total number of steps per episode. An episode terminates either when the classifier predicts success or when the step limit is reached.

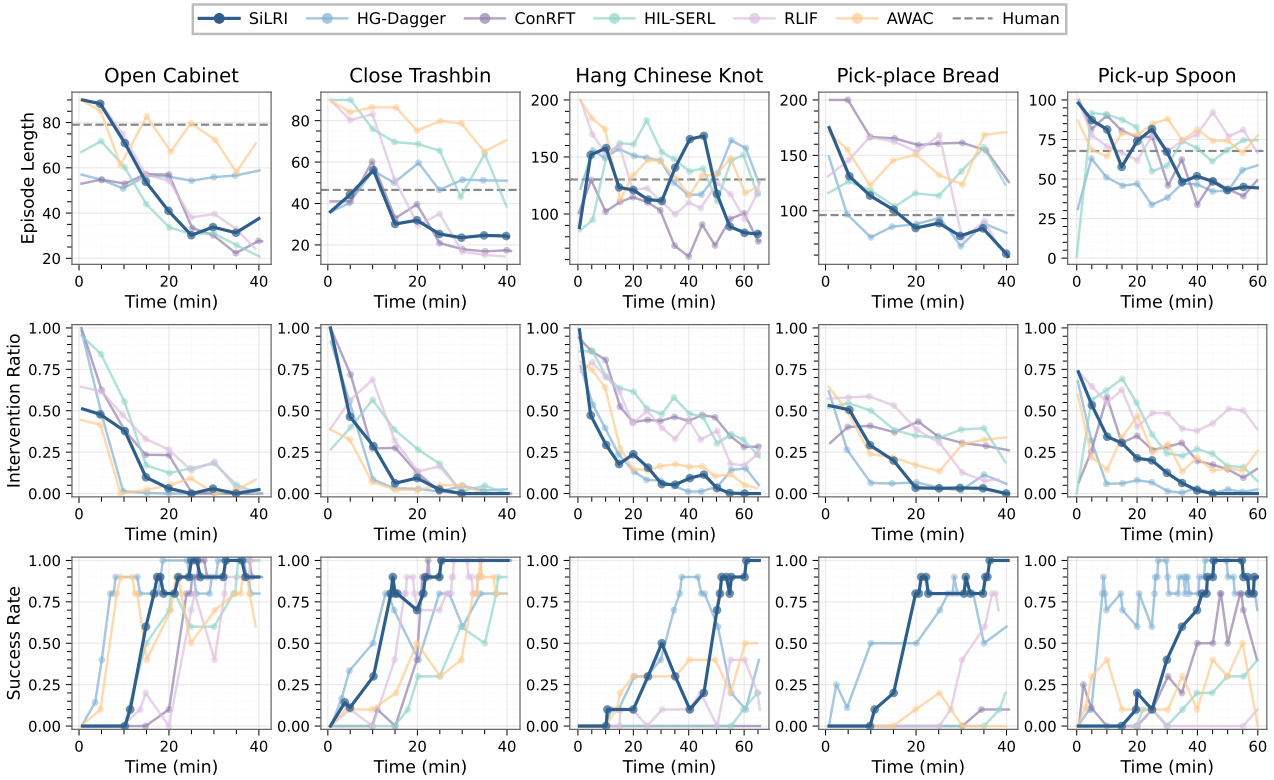

Figure 4. **Training Curves of Episode Length, Intervention Ratio, and Success Rate.** We present results on five different tasks. To ensure consistency, all models within the same task are trained and intervened by the same human operator.

5.3. Overall Performance Comparison

To address RQ1, we conduct online experiments under the same training duration and conditions for all methods. Fig. 4 presents the training curves across five tasks. For clarity, we select one data point every five minutes and retain all data points above 80% success rate. The complete original training curves for all 9 tasks are provided in Appendix E. We also compute the average episode length under human operation from the 20 successful offline demonstrations, denoted as "Human".

As shown in Fig. 4, SiLRI achieves a high success rate earlier than the other RL methods. For example, in *Open Cabinet* task, SiLRI reaches a success rate of around 90% after only 15 minutes of online training, whereas ConRFT and HIL-SERL attain similar performance 10-20 minutes later, respectively. This advantage stems from the imitation objective modulated by our proposed state-wise Lagrange multiplier. Although ConRFT also incorporates an imitation loss, it uses a small coefficient (0.5) during online training. In contrast, for SiLRI, the average value of the Lagrange multiplier is often larger than 1 at the beginning of training (see Fig. 10 in Appendix, for the convergence behavior of the multiplier). This highlights the critical role of carefully balancing the RL and BC objective when learning from suboptimal interventions. Offline RL method AWAC has an apparent advantage during the initial training stage due to its offline pretraining. However, its online training performance is unstable, showing lower success rates than other RL methods (SiLRI and RLIF), especially in tasks with long episode length, such as the Pick-up Bread task.

To our surprise, the online IL method HG-Dagger shows noticeably best performance in the first 10 minutes and achieves a comparable or even lower intervention ratio than other RL methods, which is different from the results reported in ConRFT and HIL-SERL. We attribute this to our expanded operation space and longer operation horizon. Prior works typically consider a smaller workspace and shorter episodes, focusing on a single skill. In our experiments, objects may be moved by up to 30 cm along each axis, resulting in a substantially larger exploration space compared to previous setups with less than 10 cm of motion, and we impose no constraints on rotation. Consequently, RL methods require more time for exploration. Nevertheless, even under this challenging exploration regime and sparse feedback, SiLRI still exhibits a rapid performance warm-up, as discussed above.

Although HG-Dagger attains better performance at the beginning, the learned policy struggles to maintain consis-

Table 1. **Robustness Experiments across 4 Robotic Manipulation Tasks.** In the original setting, the object's initial position is the same as in training. In the disturbance setting, external perturbations are introduced to evaluate the robustness.

Task / Method	Original				Disturbance			
	Close Trashbin	Push-T	Hang Chinese Knot	Pick-up Spoon	Close Trashbin	Push-T	Hang Chinese Knot	Pick Spoon
HG-Dagger	0.93	0.53	0.53	0.87	0.07	0.27	0.27	0.27
HIL-SERL	0.93	0.27	0	0.47	0.33	0.2	0	0.27
ConRFT	1	0.47	0	0.87	0.47	0.27	0	0.6
RLIF	0.93	0.6	0	0	0.67	0.27	0	0
AWAC	0.87	0.67	0.6	0.47	0.4	0.4	0.4	0.27
SiLRI	1	0.67	1	0.87	0.93	0.47	0.53	0.8

tently high performance later, especially in tasks where human operators are prone to making suboptimal interventions. For example, in the *Hang Chinese Knot* task, human operators may take a long time to hang the knot on the hook due to the deformability of the Chinese knot and the small interaction area around the hook. In this setting, SiLRI achieves a stable 100% success rate after 60 minutes, whereas HG-Dagger drops to 40% as more suboptimal human interventions are introduced. Across other tasks, HG-Dagger similarly fails to maintain a 100% success rate, while SiLRI remains consistently high once the policy has converged.

These results highlight the complementary yet crucial roles of RL objective and BC objective in real-world RL under suboptimal interventions. By employing self-adaptive Lagrange weights, SiLRI effectively marries the strengths of RL and BC, enabling robust learning on a broader class of tasks beyond precision manipulation.

5.4. Robustness Experiments

To address RQ2, we evaluate methods on four tasks where the robot is prone to making mistakes due to dynamic changes or its own execution errors. We deliberately introduce external disturbances to examine the robustness and failure recovery ability of each method. The evaluation tasks and disturbance rules are as follows:

- **Close Trashbin**: Move the trashbin to 5 different poses (varying in both translation and orientation) when the robot gripper is approaching the lid. Each pose is evaluated 3 times.
- **Push-T**: Move T-shaped object along 5 different directions while the robot is pushing it. Each direction is evaluated 3 times.
- **Hang Chinese Knot**: Move the Chinese knot to five different positions just before the robot is about to pick it up. Each position is evaluated 3 times.
- **Pick-up Spoon**: Move the spoon to the left or right along the bowl rim just before the robot is about to pick it up.

For each task, we evaluate each method in 15 trials in total and check whether it can successfully recover from these disturbances within the specified time limit.

As shown in Tab. 1, we report results under both the original setting (no external disturbances) and the disturbance setting, where objects are perturbed according to the above rules. Firstly, we observe that SiLRI achieves high success rates on these tasks under original settings. For example, in the *Hang Chinese Knot* task, only SiLRI reaches a 100% success rate, whereas the other two RL methods fail to complete this task fully autonomously due to the large exploration space.

Secondly, a high success rate in the original training setting does not necessarily indicate strong robustness. For example, in the *Close Trashbin* task, HG-Dagger attains 93% success rate in the original setting, but its performance drops to 7% under disturbances, highlighting the tendency of online BC methods to overfit. SiLRI also exhibits noticeable performance degradation on the *Hang Chinese Knot* task, the success rate decreases by 47% under disturbances, primarily because the policy fails to close the gripper after moving above the knot. In our implementation, the gripper is updated via a simple BC strategy, chosen due to the low control frequency and large exploration space. Nevertheless, for the other tasks, SiLRI maintains good recovery performance. For example, in *Pick-up Spoon*, SiLRI achieves a success rate of 80% under disturbances, reliably re-approaching and re-grasping the spoon after it falls into the bowl or is moved by the human.

5.5. Ablation Studies

Recall that in Eq. (10), the policy optimization objective in SiLRI consists of three terms: the RL term, the Lagrange term, and the BC term. To assess the impact of each component and address **RQ3**, we conducted experiments on the *Close Trashbin* task by removing these terms individually.

As shown in Fig. 5, removing the RL objective (green line) leads to a significant performance drop, although some success is observed around 10 minutes of online training. This highlights the necessity of guidance from the Q function, which approximates value information from interactions.

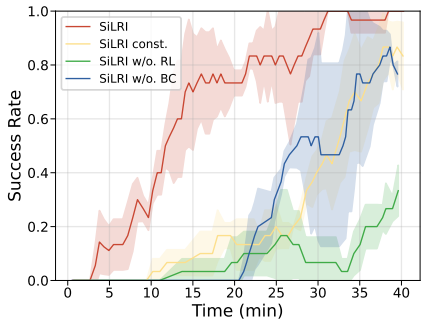

Figure 5. **Ablation Experiments in Close Trashbin.**

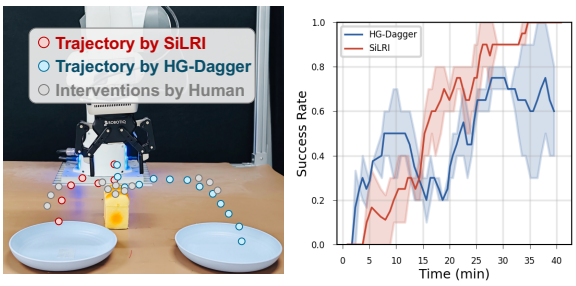

Figure 7. **Task: Place Bread into the Blue Plate.**

Next, we remove the BC term entirely from Eq. (10), optimizing only the RL objective without Lagrange multiplier normalization. In this setting, SiLRI w/o BC (blue line) converges similarly to the full method, but approximately 13 minutes later, highlighting the critical role of the BC term in accelerating training. Finally, we replace the Lagrange multiplier with a fixed coefficient of 0.5. The policy (yellow line) warms up quickly, matching SiLRI in reaching autonomous task completion within 10 minutes, but the fixed coefficient degrades long-term performance: additional suboptimal human interventions during online training cause a notable performance drop.

5.6. Investigations

Investigation on Data Quality For **RQ4**, we investigate how intervention quality affects training performance. We consider two human operators: a skilled operator familiar with the teleoperation system and intervention timing, and an unskilled operator who is more prone to intervention errors. Quantitatively, in the Push-T task, the skilled operator completes an episode in 99 steps on average, whereas the unskilled operator requires 125 steps. As shown in Fig. 6, we plot the training curves of HG-Dagger and SiLRI on the *Push-T* task, where the mean and confidence interval are computed from two training processes conducted by the skilled and unskilled human operators. During 50–60 minutes of online training, SiLRI exhibits more stable performance, maintaining a success rate of around 74%. In contrast, HG-Dagger is more sensitive to intervention quality: although it also reaches a relatively high average success rate of 63% at 60 minutes, its lower confidence bound can drop to 40%. This demonstrates that, compared with the online IL method HG-Dagger, the RL-based SiLRI is less dependent on near-optimal interventions and remains effective across human operators with varying skill levels.

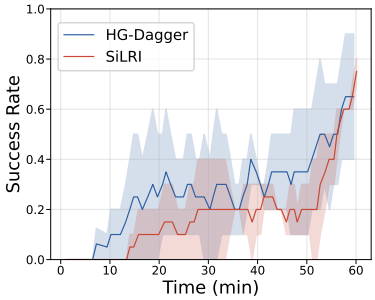

Figure 6. **Push-T Task Under Two Human Operators.**

Investigation on Multi-modal Intention High intervention entropy can also stem from multi-modal intent rather than task failures. In Fig. 7, we consider a task where the human places the bread into either of two plates with equal probability. In this setting, SiLRI can still handle multi-modal interventions by relying more heavily on RL guidance to find a single efficient solution, typically choosing the closer plate. In contrast, HG-Dagger may spend more time moving to the farther plate, leading to more fluctuating success rates. As mentioned before, we focus on deterministic policies that map each state to a single optimal action. For methods that aim to preserve multi-modal intent with stochastic policies, researchers could replace the unimodal Gaussian policy with more expressive models, such as diffusion models or mixture density networks.

6. Conclusion

In this work, we propose a state-wise Lagrangian reinforcement learning (RL) algorithm from suboptimal interventions, for real-world robot manipulation training. Observing the fact that human operators have different confidence level and manipulation skill over different states, a state-dependent constraint is added to the RL objective to automatically adjust the distance between human policy and learned policy. Building on a human-as-copilot teleoperation system, we evaluate our method with other popular online RL, offline RL and imitation learning methods on 9 manipulation tasks on 3 embodiments. Our method exhibits advantages in training time to achieve high success rate, robustness against external disturbances, and fewer execution steps. Ablation studies and investigation experiments also conducted to learn the advantage of SiLRI.

Acknowledgement

This research was funded and supported by the Beijing Innovation Center of Humanoid Robotics.

References

Albanie, S., Liu, Y., Nagrani, A., Miech, A., Coto, E., Laptev, I., Sukthankar, R., Ghanem, B., Zisserman, A., Gabeur, V., et al. The end-of-end-to-end: A video understanding pentathlon challenge (2020). *arXiv preprint arXiv:2008.00744*, 2020.

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

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al. $\pi 0$: A vision-language-action flow model for general robot control. *arXiv preprint ARXIV.2410.24164*, 2024.

Chen, Y., Tian, S., Liu, S., Zhou, Y., Li, H., and Zhao, D. Conrft: A reinforced fine-tuning method for vla models via consistency policy. In *Robotics: Science and Systems*, 2025.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Ghosh, D., Walke, H. R., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., Tan, Y. L., Chen, L. Y., Vuong, Q., Xiao, T., Sanketi, P. R., Sadigh, D., Finn, C., and Levine, S. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

Guo, Y., Zhang, J., Chen, X., Ji, X., Wang, Y., Hu, Y., and Chen, J. Improving vision-language-action model with online reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pp. 15665–15672. IEEE, 2025.

Hao, X., Zhou, Y., Wu, D., Zhang, W., Li, B., Wang, W., and Meng, D. What matters: Attentive and relational feature aggregation network for video-text retrieval. In *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Kelly, M., Sidrane, C., Driggs-Campbell, K., and Kochenderfer, M. J. Hg-dagger: Interactive imitation learning with human experts. In *International Conference on Robotics and Automation*, pp. 8077–8083, 2019.

Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M. K., Chen, L. Y., Ellis, K., et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Korkmaz, Y. and Biyik, E. MILE: model-based intervention learning. In *IEEE International Conference on Robotics and Automation*, pp. 15673–15679. IEEE, 2025.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33: 1179–1191, 2020.

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

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

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Nakamoto, M., Zhai, S., Singh, A., Sobol Mark, M., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Advances in Neural Information Processing Systems*, 36:62244–62269, 2023.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Seo, M. and Choi, K. State-dependent lagrange multipliers for state-wise safety in constrained reinforcement learning. *Authorea Preprints*, 2025.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Wu, P., Shentu, Y., Liao, Q., Jin, D., Guo, M., Sreenath, K., Lin, X., and Abbeel, P. Robocopilot: Human-in-the-loop interactive imitation learning for robot manipulation. In *Conference on robot learning*, 2024.

Xia, W., Yang, Y., Wu, H., Ma, X., Kong, T., and Hu, D. Human-assisted robotic policy refinement via action preference optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 36746–36768, 2025.

Xu, Z., Zhao, Y., Wu, K., Liu, N., Ji, J., Che, Z., Liu, C. H., and Tang, J. Hacts: a human-as-copilot teleoperation system for robot learning. *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

A. Additional Related Works on Imitation Learning

While this work focuses on reinforcement learning, we also provide a critical analysis of related imitation learning (IL) methods, including standard IL and value-weighted behavior cloning (e.g., AWR). Imitation learning has dominated the end-to-end learning paradigm (Hao et al., 2021; Albanie et al., 2020) for real-world manipulation for decades. With high-quality demonstrations collected via teleoperation systems (e.g., VR headsets (Khazatsky et al., 2024), homologous bilateral system (Xu et al., 2025; Wu et al., 2024)), it has served as a simple, efficient paradigm for vision-based robot manipulation. Previous methods, including ACT (Zhao et al., 2023) and DP (Chi et al., 2023), improve real-world performance through more expressive models. More recently, Vision-Language-Action (VLA) models, such as Octo (Ghosh et al., 2024), OpenVLA (Kim et al., 2024) and π_0 (Black et al., 2024), have shown promise for general manipulation task learning by leveraging large parameter counts and Internet-scale pretraining. However, their substantial computational cost makes online learning of manipulation skills challenging. In this work, we develop algorithms that efficiently leverage suboptimal interventions while keeping the model architecture simple, following prior work (Luo et al., 2025). A closely related line of IL (often viewed as RL) is value-weighted behavior cloning, which uses estimated values to filter and re-weight training data. Representative examples include offline-to-online methods such as AWAC and recent VLA post-training approaches (e.g., π-0.6), which report stable offline-to-online improvements. Nevertheless, compared to value-based RL methods, these approaches rely heavily on large-scale human intervention data and make less effective use of robot-generated data.

B. Theoretical Analysis

This section shows that, under standard Gaussian modeling assumptions, the state-wise KL constraint in Eq. (2) implies a state-wise constraint on the distance between policy means, as used in Eq. (3).

Assumption B.1 (Isotropic Gaussian Policies)**.** For any state $\boldsymbol{s}$, the human reference policy is modeled as an isotropic Gaussian

$$\beta(\cdot \mid \boldsymbol{s}) = \mathcal{N}\big(\boldsymbol{\mu}_\beta(\boldsymbol{s}),\ \sigma_\beta(\boldsymbol{s})^2 \boldsymbol{I}\big), \tag{13}$$

and the learned policy is modeled as an isotropic Gaussian with *fixed* standard deviation

$$\pi(\cdot \mid \boldsymbol{s}) = \mathcal{N}\big(\boldsymbol{\mu}_\pi(\boldsymbol{s}),\ \sigma_\pi^2 \boldsymbol{I}\big), \tag{14}$$

where $\sigma_\pi > 0$ is a constant, $\boldsymbol{I}$ is the $d \times d$ identity matrix, and d denotes the action dimension.

Assumption B.2 (Bounded Human Uncertainty)**.** There exist constants $0 < \underline{\sigma} \le \overline{\sigma} < \infty$ such that $\sigma_\beta(\boldsymbol{s}) \in [\underline{\sigma}, \overline{\sigma}]$ for all $\boldsymbol{s}$.

Lemma B.3 (KL Divergence Between Gaussians)**.** *For two non-degenerate Gaussians $\mathcal{N}(\boldsymbol{m}_1, \Sigma_1)$ and $\mathcal{N}(\boldsymbol{m}_2, \Sigma_2)$, the KL divergence admits the closed form*

$$\begin{aligned} &\mathrm{D}_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{m}_1, \Sigma_1) \| \mathcal{N}(\boldsymbol{m}_2, \Sigma_2)\right) \\ &\quad = \frac{1}{2}\Big(\mathrm{tr}(\Sigma_2^{-1}\Sigma_1) + (\boldsymbol{m}_2 - \boldsymbol{m}_1)^\top \Sigma_2^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1) \\ &\qquad - d + \ln \frac{\det \Sigma_2}{\det \Sigma_1}\Big). \end{aligned} \tag{15}$$

Applying Lemma B.3 under Assumption B.1 with $\Sigma_\pi = \sigma_\pi^2 \boldsymbol{I}$ and $\Sigma_\beta(\boldsymbol{s}) = \sigma_\beta(\boldsymbol{s})^2 \boldsymbol{I}$ yields, for any $\boldsymbol{s}$,

$$\begin{aligned} &\mathrm{D}_{\mathrm{KL}}\big(\pi(\cdot \mid \boldsymbol{s}) \| \beta(\cdot \mid \boldsymbol{s})\big) \qquad (16) \\ &\quad = \frac{1}{2}\Big(d\frac{\sigma_\pi^2}{\sigma_\beta(\boldsymbol{s})^2} + \frac{\|\boldsymbol{\mu}_\pi(\boldsymbol{s}) - \boldsymbol{\mu}_\beta(\boldsymbol{s})\|_2^2}{\sigma_\beta(\boldsymbol{s})^2} - d + d \ln \frac{\sigma_\beta(\boldsymbol{s})^2}{\sigma_\pi^2}\Big) \\ &\quad = \frac{1}{2} \cdot \frac{\|\Delta\boldsymbol{\mu}(\boldsymbol{s})\|_2^2}{\sigma_\beta(\boldsymbol{s})^2} + \frac{1}{2} c(\sigma_\beta(\boldsymbol{s})), \qquad (17) \end{aligned}$$

where $\Delta\boldsymbol{\mu}(\boldsymbol{s}) \triangleq \boldsymbol{\mu}_\pi(\boldsymbol{s}) - \boldsymbol{\mu}_\beta(\boldsymbol{s})$ and

$$c(\sigma) \triangleq d\Big(\frac{\sigma_\pi^2}{\sigma^2} - 1 + \ln \frac{\sigma^2}{\sigma_\pi^2}\Big). \tag{18}$$

Moreover, since $\ln x \le x - 1$ for any $x > 0$, we have $c(\sigma) \ge 0$ for all $\sigma > 0$.

Theorem B.4 (From a KL Constraint to a Mean-Distance Constraint)**.** *Suppose Assumptions B.1–B.2 hold and*

$$\mathrm{D}_{\mathrm{KL}}\big(\pi(\cdot \mid \boldsymbol{s}) \| \beta(\cdot \mid \boldsymbol{s})\big) \leq \epsilon, \qquad \forall \boldsymbol{s}. \tag{19}$$

Then, for all $\boldsymbol{s}$,

$$\frac{\|\Delta \boldsymbol{\mu}(\boldsymbol{s})\|_2^2}{\sigma_\beta(\boldsymbol{s})^2} \leq 2\epsilon - c(\sigma_\beta(\boldsymbol{s}))\,, \tag{20}$$

where $\Delta \boldsymbol{\mu}(\boldsymbol{s}) \triangleq \boldsymbol{\mu}_\pi(\boldsymbol{s}) - \boldsymbol{\mu}_\beta(\boldsymbol{s})$ and $c(\sigma)$ is defined in Eq. (18). Moreover, letting

$$c_{\min} \triangleq \min_{\sigma \in [\underline{\sigma}, \overline{\sigma}]} c(\sigma) \geq 0, \tag{21}$$

if $\epsilon > c_{\min}/2$, then

$$\|\Delta \boldsymbol{\mu}(\boldsymbol{s})\| \leq \kappa\, \sigma_\beta(\boldsymbol{s}), \qquad \forall \boldsymbol{s}, \tag{22}$$

with $\kappa \triangleq \sqrt{2\epsilon - c_{\min}}$.

Proof. By Eq. (17), the KL constraint in Eq. (19) implies $\frac{1}{2}\frac{\|\Delta \boldsymbol{\mu}(\boldsymbol{s})\|_2^2}{\sigma_\beta(\boldsymbol{s})^2} + \frac{1}{2}c(\sigma_\beta(\boldsymbol{s})) \leq \epsilon$ for all $\boldsymbol{s}$, which rearranges to Eq. (20). Under Assumption B.2, $c(\sigma_\beta(\boldsymbol{s})) \geq c_{\min}$, hence $2\epsilon - c(\sigma_\beta(\boldsymbol{s})) \leq 2\epsilon - c_{\min} = \kappa^2$. Therefore,

$$\frac{\|\Delta \boldsymbol{\mu}(\boldsymbol{s})\|_2^2}{\sigma_\beta(\boldsymbol{s})^2} \leq 2\epsilon - c(\sigma_\beta(\boldsymbol{s})) \leq \kappa^2. \tag{23}$$

Multiplying by $\sigma_\beta(\boldsymbol{s})^2$ and taking square roots yields Eq. (22). □

C. Key Design Choices and Parameters

Unlike ideal simulation environments, training real-world RL policies must cope with complex scenarios and imperfect estimations that can significantly affect overall training efficiency. Here, we highlight three key design choices that are critical for the successful training of all real-world RL methods considered in our work.

Intervention Standard Operating Procedure Empirically, we find that the timing of interventions is critical to the successful training of all RL policies, particularly for ConRFT and HIL-SERL. Intervening too frequently, such as whenever the model makes even minor mistakes, can actually reduce training efficiency by making the policy overly reliant on human assistance. In some high-uncertainty states, even with human intervention, the robot must still explore alternative actions and their outcomes. In such cases, allowing the robot to explore independently (constrained by safety boundaries) is often more effective than frequent human guidance. To ensure fair comparisons and improve efficiency, we design an Intervention Standard Operating Procedure (ISOP) to guide the human operator's intervention behavior.

1. **Initial full interventions.** Fully intervene for the first 3-5 episodes, completing the task each time to provide stable initial guidance.
2. **No-intervention exploration after severe deviation.** Once a severe deviation is observed (e.g., when the robot should move toward the trash bin but the end-effector instead explores in an unrelated direction), **do not intervene**. Let the robot explore until the episode **times out**.
3. **Early intervention after repeated severe deviations.** If step (2) is repeated and a severe deviation is observed **twice** at the **same location/state**, then in the next episode, **intervene earlier** at the earliest point where the deviation first occurs.
4. **Fallback to full demonstration after repeated failures.** If the robot fails to complete the task for **five** consecutive episodes under the above protocol, then in the next episode, **fully intervene from the beginning** and complete the task, providing a complete human demonstration trajectory.

While it is impractical for human operators to strictly follow the SOP during execution, it still serves as a valuable guideline for efficient online training.

Ever-correcting Reward Classifier As pointed out by previous work (Luo et al., 2025), even when a reward classifier achieves up to 95% precision after offline training, it can still produce many false-negative or false-positive examples when deployed in online environments. A common strategy is to collect additional data using a random policy and to add hand-crafted rules to improve classifier robustness, but this is labor-intensive and difficult to scale. Since a highly precise reward classifier is crucial for successful training under extremely sparse feedback, we instead introduce an ever-correcting reward classifier module during online training. The classifier is retrained whenever newly corrected labels are added to the reward buffer, whether they correct false negatives or false positives.

Slow-learning Lagrange Network Since there is no explicit upper bound on the output of the Lagrange network, the multiplier $\lambda(\boldsymbol{s})$ can easily grow to extremely large values whenever the constraint is violated, which leads to unstable training for both the Lagrange and policy networks. Following prior work on state-wise Lagrange multipliers (Seo & Choi, 2025), we set the learning rate of the Lagrange network to be 0.01 times that of the other networks. Specifically, we use a learning rate of 3×10^{-4} for the Q, π, and β networks, and a learning rate of 3×10^{-6} for the Lagrange network λ.

D. Experiment Details

D.1. Details on Experimental hardware setups

We experiment on three embodiments. For UR5 and Franka, we provide two RGB-D camera views (a right-side view and a wrist-mounted camera): UR5 is equipped with Orbbec Gemini 336 and 336L units, while Franka uses two Intel RealSense cameras. Both robots use the Robotiq 2F-85 gripper. The humanoid dual-arm robot shares a similar setup, but with three cameras (a top-mounted head camera and two wrist cameras) and a humanoid-style gripper. The details are shown in Fig. 8.

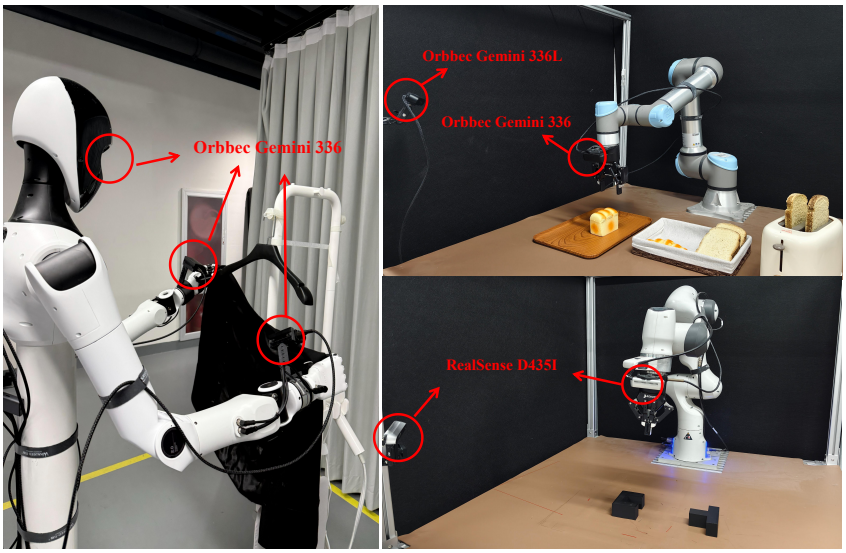

Figure 8. **Experimental hardware setups.** Left: humanoid dual-arm robot with three cameras (one head-mounted and two wrist-mounted); top-right: UR5 with Orbbec Gemini 336/336L cameras; bottom-right: Franka with two Intel RealSense cameras.

D.2. Details on Experiment Tasks

The description of nine real-world manipulation tasks are as follows:

1. **Pick-place Bread**: Pick the bread from the wooden plate and place it into the basket.
2. **Pick-up Spoon**: Pick up the spoon from the pink bowl.
3. **Fold Rag**: Fold the half-hanging rag into the basket.
4. **Open Cabinet**: Open the closed cabinet drawer.
5. **Close Trashbin**: Close the trash-bin lid, which is spring-loaded and requires appropriate force to close.

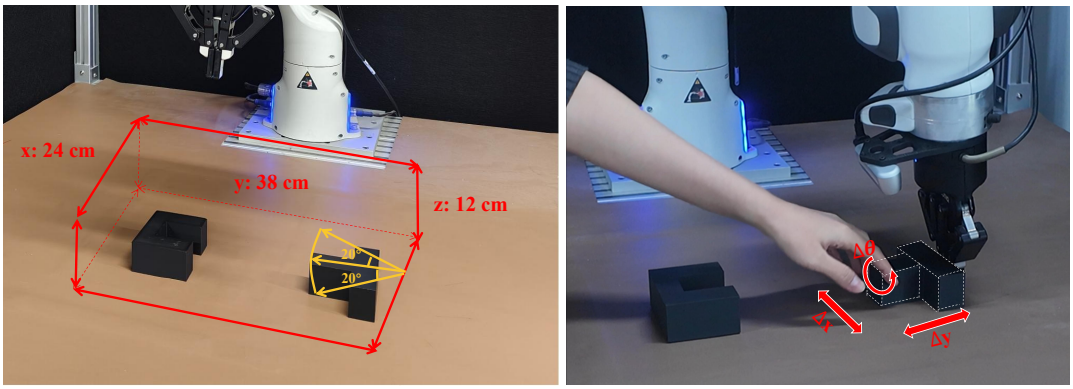

Figure 9. **Training and Evaluation Examples.** Left: Safety box and initial positions of objects. Right: Disturbance mode and range.

6. **Push-T**: Push a T-shaped object into a U-shaped groove using a closed gripper.
7. **Hang Chinese Knot**: Pick up the Chinese knot from the table and hang it onto a hook.
8. **Insert USB**: Insert a USB stick from the right-side container into a power strip.
9. **Hang Clothes**: The robot first need to close the grippers to grasp the clothes, and then reach the gap between the black hanger and white holder to hang clothes from back to front, and lastly open the grippers to let clothes stable in the hanger.

D.3. Details on Parameter Settings

Here, we provide detailed descriptions of the overall task settings and training parameters.

The task-specific settings are summarized in Tab. 2. The space constraint is crucial for efficient training of HIL-SERL and ConRFT. This is because their Q-functions rely on self-collected experience to obtain accurate value estimates; simply intervening when the robot drifts far from the target is insufficient to provide informative data for these methods. In contrast, for SiLRI and HG-Dagger, the boundary primarily serves as a safety mechanism, since the imitation loss already guides the policy toward human behavior. We visualize the safety box and other settings in Fig. 9.

Table 2. **Task Setting Details**. For single arm tasks, they share a 7-dimensional action space: 6-DoF end-effector pose increments and discrete gripper actions (open or close). When the gripper is fixed for a task, the last dimension remains a constant value; when the gripper is not fixed, the last dimension is generated by a separate policy network. For the dual-arm task, it has a 12+2 action space.

Task/Parameter	Fix Gripper	Max episode length	Train time	Space constraint	Embodiment
Close Trashbin	True	90	40 min	$0.47 \leq x \leq 0.68$ m, $-0.12 \leq y \leq 0.10$ m, $0.29 \leq z \leq 0.48$ m	Franka
Open Cabinet	True	90	40 min	$0.37 \leq x \leq 0.52$ m, $-0.11 \leq y \leq 0.13$ m, $0.39 \leq z \leq 0.49$ m	Franka
Insert USB	True	120	40 min	$0.60 \leq x \leq 0.67$ m, $-0.14 \leq y \leq -0.03$ m, $0.38 \leq z \leq 0.51$ m	Franka
Push-T	True	120	60 min	$0.45 \leq x \leq 0.69$ m, $-0.05 \leq y \leq 0.33$ m, $0.16 \leq z \leq 0.28$ m	Franka
Hang Chinese Knot	False	200	65 min	$0.47 \leq x \leq 0.74$ m, $-0.16 \leq y \leq 0.13$ m, $0.17 \leq z \leq 0.45$ m	Franka
Pick-place Bread	False	200	40 min	$-0.65 \leq x \leq -0.27$ m, $-0.42 \leq y \leq -0.06$ m, $0.19 \leq z \leq 0.31$ m	UR
Fold Rag	False	100	40 min	$-0.55 \leq x \leq -0.37$ m, $-0.62 \leq y \leq -0.42$ m, $0.29 \leq z \leq 0.41$ m	UR
Pick-up Spoon	False	100	60 min	$-0.51 \leq x \leq -0.35$ m, $-0.43 \leq y \leq -0.27$ m, $0.22 \leq z \leq 0.31$ m	UR
Hang Clothes	False	200	150 min	$0.25 \leq x \leq 0.50$ m, $-0.40 \leq y \leq 0.40$ m, $-0.15 \leq z \leq 0.55$ m	Humanoid

Tab. 3 lists the key hyperparameters used in our experiments. The same training hyperparameters are applied across all tasks. To ensure a fair comparison, HG-Dagger, HIL-SERL, and SiLRI share identical parameter settings, while ConRFT follows its officially released hyperparameters and network architectures.

D.4. Details on Lagrange Multipliers

We also provide the training curves of the Lagrange multipliers in Fig. 10 to illustrate the self-adaptive trade-off between the RL objective and the BC objective in SiLRI. Specifically, we plot the batch-mean output of the Lagrange multiplier network during training. As shown in Fig. 10, at the beginning of training, the large discrepancy between the behavior

Table 3. **Training Details**.

Parameter	Value
Online Buffer Batch Size	128
Offline Buffer Batch Size	128
Actor Learning Rate	0.0003
Critic Learning Rate	0.0003
Expert Learning Rate	0.0003
Lagrange Learning Rate	0.000003
Optimizer	Adam
Discount Factor	0.97
Initial Offline Demonstrations	20 for single and 30 for dual-arm
κ	6 for single and 12 for dual-arm

policy β and the actor policy π causes the Lagrange multiplier to quickly rise above 1, so that updates to π are mainly driven by the BC loss. As training progresses and π becomes closer to β, the Lagrange multiplier gradually decreases and stabilizes around 0.3, shifting the emphasis toward the RL objective.

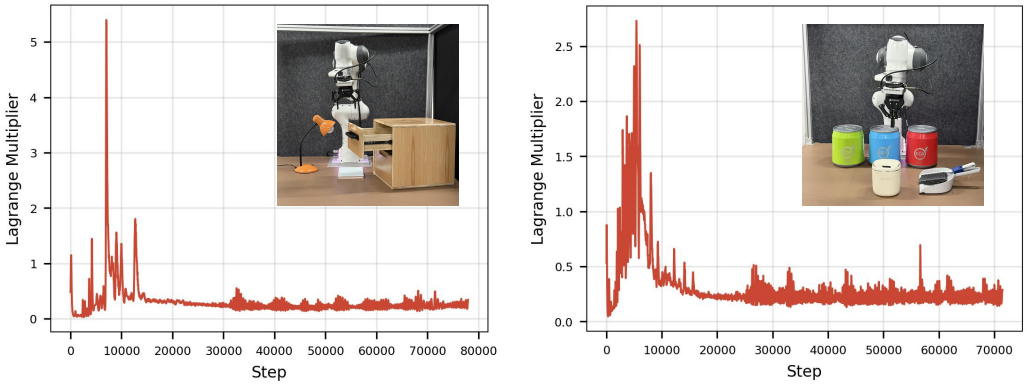

Figure 10. **Task: Lagrange Multiplier Training Curves.**

D.5. Details on Reward Functions

As mentioned in the main text, we use a simple sparse reward setting: +10 for successful task completion and -0.05 penalty per step otherwise. Since gripper actions are learned via behavior cloning, they require no reward signal. For other RL baselines, we apply a -0.5 penalty when the gripper switches between open and closed states to discourage unnecessary changes. Additionally, RLIF applies a -1 penalty to the transition just before each intervention frame.

D.6. Details on Metric Computation

Beyond the algorithm itself, real-world online training results are influenced by many external factors. As a result, the training curves can be noisy and exhibit substantial fluctuations, especially for binary metrics such as the success rate and intervention ratio. To more clearly show performance trends over the course of training without loss of authenticity, we apply the following smoothing function to post-process the Success Rate, Intervention Ratio, and Episode Length metrics.

We use the sliding window average method to process the index sequence, and calculate the arithmetic mean of the data in the specified window to smooth the noise fluctuation, while retaining the time series characteristics of the training trend. Specifically, the sliding window is composed of the data of each episode and its first nine episodes, and the average success rate in the window is calculated point by point.

E. Additional Experiment Results

Dual-arm Task For the dual-arm manipulation task, we compare SiLRI with HG-Dagger, and exclude other RL methods due to safety concerns and low training efficiency under a large exploration space (approximately 0.8 m in width and 0.7 m in height). As shown in Fig. 11, both methods have a good training performance. HG-Dagger converges faster, reaching ∼80% success after 60 minutes of online training, but its performance drops sharply when the hanger position changes. SiLRI reaches a high success rate after 100 minutes of training and is more robust to hanger movement. In particular, HG-Dagger overfits to a fixed hanger position and executes a learned trajectory rigidly, whereas SiLRI adaptively adjusts its approach based on the hanger's actual position until success or the time limit is reached. More recovery results are in website.

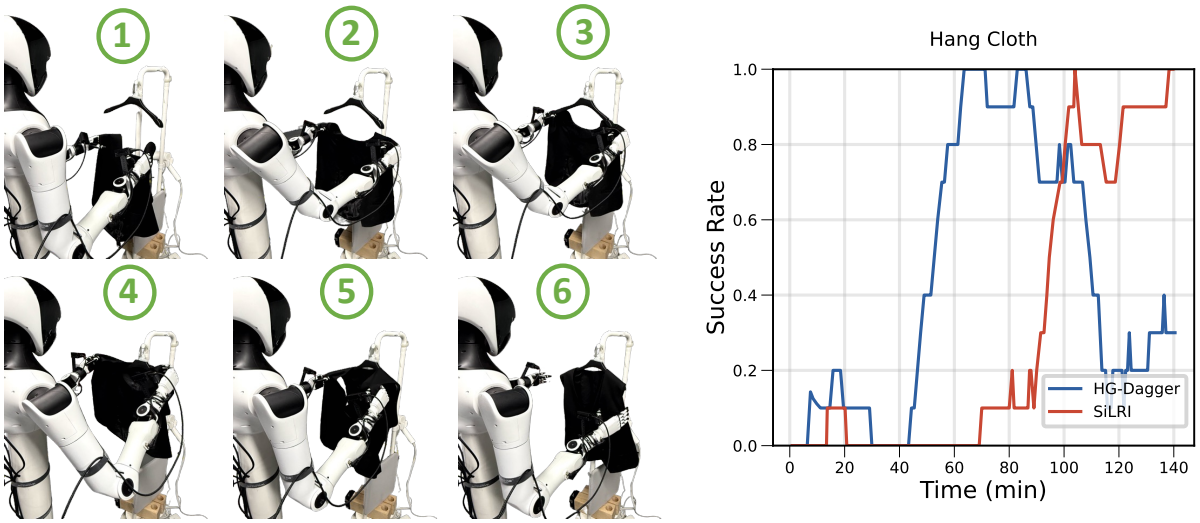

Figure 11. **Experimental results on the Hang Clothes task.** Left: Execution trajectory of SiLRI. Right: Performance comparison between HG-Dagger and SiLRI. Other RL-based methods are excluded from this task due to safety concerns and low sample efficiency.

Detailed Training Time Improvements As shown in Tab. 4, we compare the training time for our method and other RL methods to reach 90% success rate. Negative values indicate faster convergence for our method; positive values indicate slower convergence; dashes indicate that other methods failed to reach 90% success. Our method achieves high performance with shorter training time and successfully reaches high success rates in Hang Chinese Knot, Pick-place Bread, and Pick-up Spoon, where other RL methods fail.

Table 4. **Training Time Improvements (↓).**

Task	ConRFT	HIL-SERL	RLIF	AWAC
Open Cabinet	-7.8 min (↓31.0%)	-17.7 min (↓50.4%)	-15.3 min (↓46.8%)	+8.2 min (↑89.1%)
Close Trashbin	-6.6 min (↓31.3%)	-23.5 min (↓61.8%)	-3.1 min (↓17.6%)	-19.1 min (↓56.8%)
Hang Chinese Knot	—	—	—	—
Pick-place Bread	—	—	—	—
Pick-up Spoon	—	—	—	—
Insert USB	+13.6 min (↑66.3%)	—	—	—
Push-T	—	—	—	—
Fold Rag	-20.0 min (↓55.2%)	—	-11.7 min (↓41.9%)	—

Additional Comparison Results Due to space constraints, we move the results for the whole 8 tasks to the appendix, as shown in Fig. 12. Beyond the main analysis, most RL methods demonstrate significantly reduced episode lengths and execution times compared to HG-Dagger. For example, in *Open Cabinet* and *Close Trashbin* tasks, HG-Dagger requires approximately 20 and 25 more steps, respectively, than SiLRI to complete the task. Notably, even compared to human operators, SiLRI demonstrates lower time in tasks that require precise manipulation, such as *Insert USB* and

Close Trashbin. These results indicate that RL methods, represented by SiLRI, are well suited for tasks demanding high execution speed and performance, potentially exceeding human-operated efficiency.

Although our method demonstrates faster convergence on most tasks, it achieves a lower success rate than RLIF and AWAC on Push-T. But it still requires the minimal intervention ratio, indicating continued learning optimal policies at specific states.

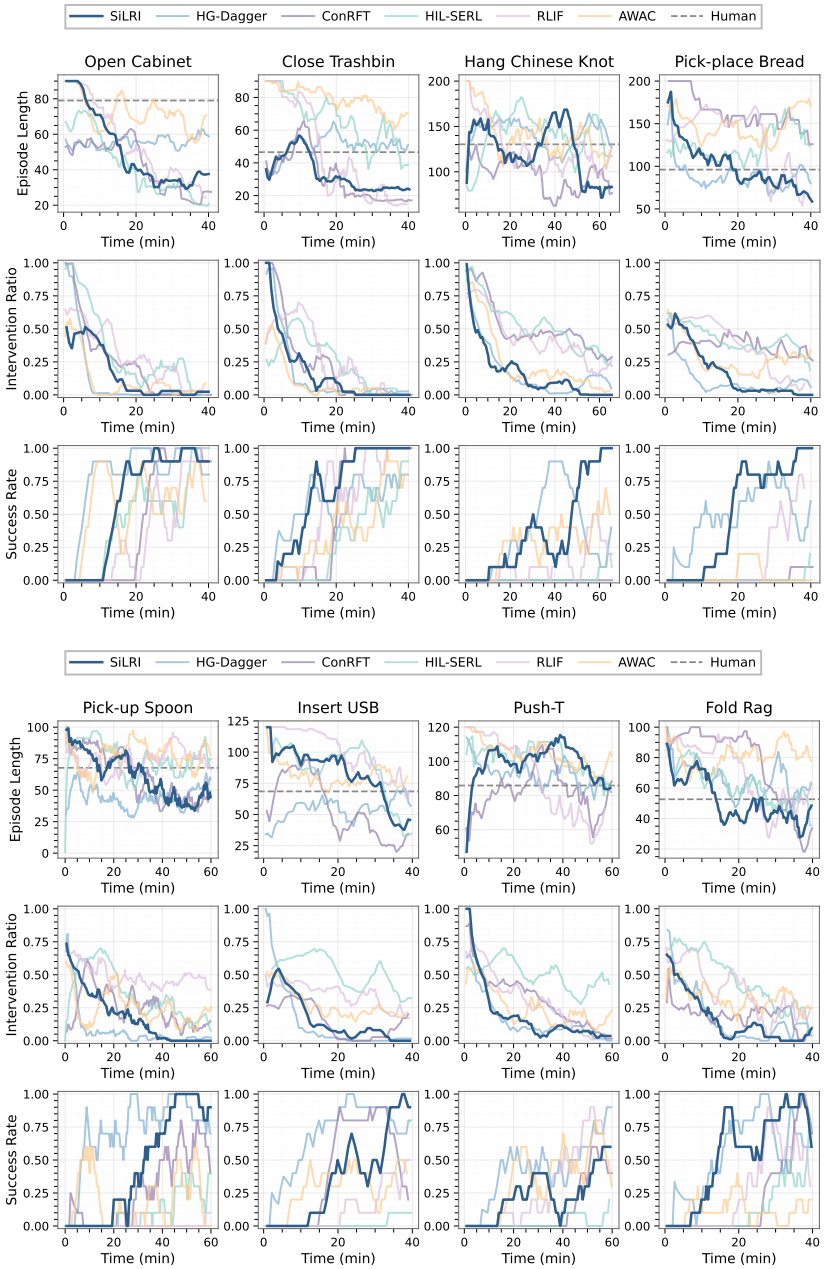

Figure 12. **Complete Training Curves.**

Investigation on Smaller Randomization In the main text, we explore larger task randomization (Rand.) than the original ConRFT and HIL-SERL settings to demonstrate that RL can solve tasks beyond small-space precise manipulation. Here, we add an experiment to clarify how performance changes with randomization difficulty. We evaluate Hang Chinese Knot under both high randomization (our setting) and low randomization (the original ConRFT setting, with rotation con-

straints and 25 cm inter-object distance). As shown in Table 5, RL-based methods show advantages in high-randomization regimes where exploration space is larger. For example, ConRFT achieves 80% success in the low-randomization regime but fails in the high-randomization regime. This aligns with expectations: as exploration space grows, prior information from interventions becomes more critical for guiding meaningful exploration. Our method achieves the best robustness across both regimes, benefiting from the combination of self-exploration and imitation learning.

Method	Original		Disturbance	
	High Rand.	Low Rand.	High Rand.	Low Rand.
SiLRI	**1.0**	**1.0**	**0.53**	**0.80**
ConRFT	0.0	0.80	0.0	0.40
HG-Dagger	0.53	0.60	0.27	0.27

Table 5. **Evaluation over Different Levels of Training Randomization.**

F. Future Works

During real-world experiments, we identify several limitations of current online training methods, including SiLRI.

First, the training time required for online methods to reach human-comparable performance remains long. Even for an unskilled teleoperation operator, the Hang Chinese Knot task can typically be handled after only a few minutes of practice. In contrast, online learning methods require at least 60 minutes of training to achieve a workable solution. In future work, we plan to incorporate vision-language-action models with pre-existing manipulation skills to reduce the required online fine-tuning time.

Second, there remains substantial room for improvement in robustness. While SiLRI demonstrates strong performance in static settings, its robustness degrades in dynamic environments, particularly for challenging tasks with large exploration spaces. Beyond using intervention data directly, richer intervention signals could be leveraged to implicitly constrain the exploration space and improve robustness.

Despite these limitations, we believe that online reinforcement learning remains a promising paradigm for achieving superhuman intelligence, given its efficiency in daily manipulation tasks and its potential to surpass human performance in precise manipulation scenarios.