How to Learn from What a Human Would Avoid? Intervention-Aware World Models with Real-World RL for Dexterous Manipulation

Jiaju Yin[1,*] **Zhenhui Zhang**[1,*] **Lixin Xu**[1] **Heng Zhang**[2]
Jun Shao[3] **Yating Feng**[1] **Arash Ajoudani**[2,†] **Renjing Xu**[1,†]
[1]HKUST (Guangzhou) [2]Italian Institute of Technology [3]Zhejiang University
[*]Equal contribution [†]Corresponding author

Project page: https://whirl-dexterous.github.io/

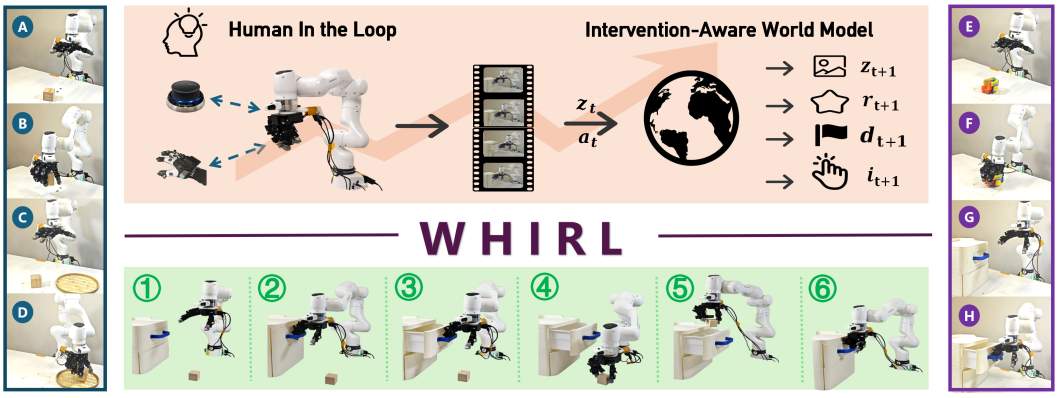

Figure 1: **WHIRL learns reusable risk from human takeovers.** Each pedal intervention becomes a next-step takeover label for an intervention-aware world model, whose prediction head shapes the residual actor away from intervention-prone states across dexterous manipulation tasks.

Abstract: Multi-fingered dexterous manipulation remains a frontier for real-world reinforcement learning (RL) due to the high-dimensional action space and the prohibitive cost of hardware failures. While human-in-the-loop (HIL) RL allows operators to intervene before failures occur, current pipelines often treat these interventions as reactive corrections, discarding the rich safety signal inherent in the operator's decision to take control. In this paper, we ask: *How can we learn from what a human would avoid?*

We present **WHIRL**, a safety-aware RL framework that transforms binary human interventions into forward-predictive signals for proactive risk avoidance. Our approach centers on an intervention-aware latent world model with four prediction heads: dynamics, reward, termination, and a novel per-state intervention-probability head that learns to predict the likelihood of a human takeover at future states. This head provides an actor-side risk-shaping term that discourages the policy from entering "intervention-prone" regions, modeling the operator's internal safety threshold.

We evaluate our framework on a 16-DoF LEAP Hand across tasks spanning convex and irregular object grasping, prismatic manipulation, and long-horizon multi-stage tasks. Our results show that predictive risk-shaping enables the system to achieve a 96.7% success rate on complex grasping tasks while reducing the operator intervention burden by up to 84% in step-weighted terms. By closing the loop between human intuition and predictive world modeling, this work provides a practical safety-aware recipe for training complex dexterous agents in the real world while reducing operator fatigue and hardware-risk exposure.

Keywords: Dexterous Manipulation, Human-in-the-Loop RL, World Models, Real-World Reinforcement Learning, Teleoperation Retargeting

1 Introduction

Dexterous manipulation [1, 2, 3], grasping, repositioning, and placing objects with multi-fingered hands, remains a central challenge in real-world robot learning. Behavior cloning from teleoperated demonstrations [4, 5] has become the practical starting point: a compact demonstration set can teach a dexterous hand the nominal motion needed to approach, grasp, and place objects. However, contact-rich manipulation is governed by small variations in object pose, finger timing, and incipient slip. These variations are difficult to cover exhaustively with demonstrations, so a behavior-cloning policy that appears competent on nominal trials can still fail under moderate deployment shift.

Online reinforcement learning offers a direct way to turn these failures into policy improvement [6, 7, 8, 9]. For low-dimensional grippers, SERL [10] and related actor–learner systems [8, 11] show that online RL can reach strong real-robot performance within hours. For a 16-DoF dexterous hand, the same recipe becomes substantially more expensive. The policy acts in a coupled arm–hand action space, contacts are stiff and discontinuous, and each exploratory mistake consumes real robot time, resets, and hardware attention. Sim-to-real approaches [12, 13, 2] reduce this burden when accurate simulation is available, but they do not remove the need for real-robot adaptation on a new hand, task, and teleoperation stack.

Human-in-the-loop (HIL) RL is a natural compromise: let the policy explore, but allow a human to take over before unrecoverable failures. HIL-SERL [14], SiLRI [15], and interactive imitation methods [16, 17, 18] use interventions to stabilize real-robot learning. The cost of HIL is expert attention: each takeover consumes operator time, so the central question is sharp—*does each second of operator attention pay off after the pedal is released?*

Current HIL pipelines *use each takeover once and throw the rest of the signal away*: the binary label becomes a behavior-cloning mask, a replay priority, or a safety metric, but never supervision for predicting where the autonomous policy will soon need help. This points to a simple use of latent world models [19, 20, 21, 22]: alongside dynamics, reward, and termination, learn a takeover-probability head that converts each intervention into a reusable actor-side risk signal.

We present **WHIRL** (**W**orld models, **H**uman-in-the-loop, **I**ntervention, **R**eal-world, reinforcement **L**earning), and make three contributions:

Our contributions.

- **Intervention-aware world model.** A compact latent model predicts dynamics, reward, termination, and next-step takeover probability; the first three heads provide uncertainty-gated critic targets, while the intervention head supplies actor-side risk shaping.
- **Mid-rollout HIL teleoperation system.** A tri-channel interface couples arm control, glove-based hand control, and pedal takeover; its LEAP + Manus retargeter combines non-thumb affine joint mapping, thumb fingertip IK, and command rebase for continuous interventions.
- **Controlled real-robot evidence.** On a 16-DoF LEAP Hand, we evaluate WHIRL across five dexterous tasks with a progressive baseline ladder and a mechanism ablation, isolating how predictive intervention modeling reduces operator burden.

2 Related Work

Real-world online RL for manipulation. Real-world online RL has advanced through direct actor–learner systems such as SERL [10] and related platforms [8, 11], and more recently through large-policy post-training such as RECAP ($\pi^*_{0.6}$) [23] and Fleet-Scale RL [24]. For dexterous hands, residual RL is a conservative alternative: ResFiT [25] freezes a behavior-cloning prior and learns a sparse-reward correction, providing the backbone we adopt. HIL methods, such as HIL-SERL [14], SiLRI [15], and interactive imitation [16, 17, 18], make exploration safer by adding takeovers to replay, but use each takeover once and throw the rest of the signal away: the label becomes current-

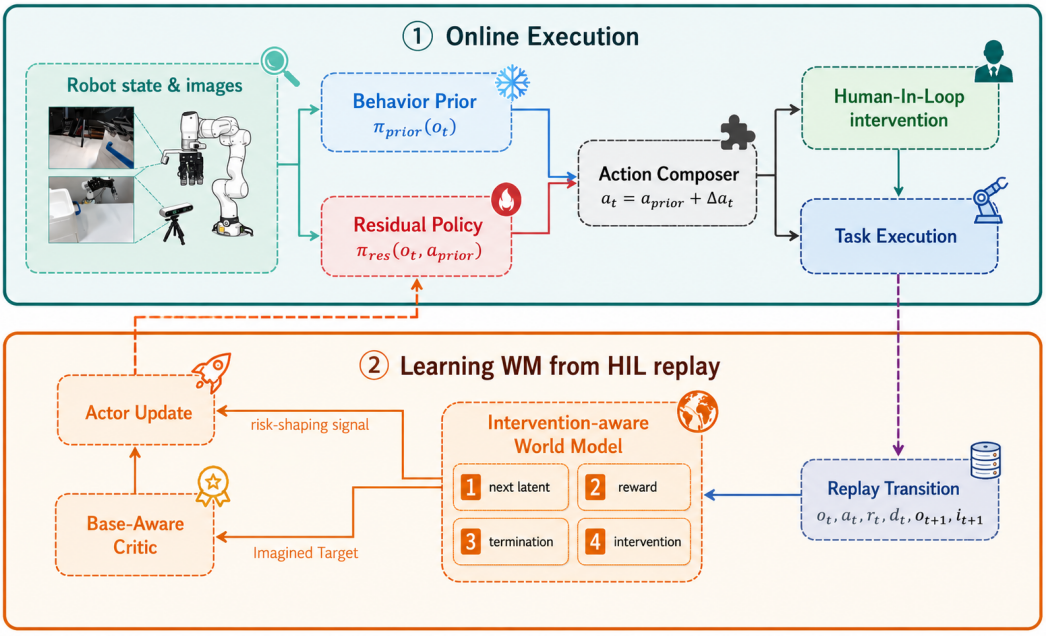

Figure 2: WHIRL overview: **takeovers become predictive risk.** During online execution, a frozen behavior prior and residual policy compose arm–hand actions while HIL takeovers create labeled replay. The intervention-aware world model then predicts dynamics, reward, termination, and next-step intervention probability; the first three heads support critic learning, while the intervention head supplies actor-side risk shaping.

step behaviour supervision, a replay weight, or a metric. WHIRL instead trains it as a next-step intervention-risk prediction target the actor consumes on every state.

World models for robot control. Dreamer [19, 20], MBPO [21], and TD-MPC [22] established latent world models that predict environment quantities and train actors or critics with imagined data. Hi-WM [26] moves this idea into HIL learning by letting the operator intervene inside a learned model. WHIRL instead keeps corrections on the real robot and uses a small one-step world model whose distinctive head predicts future human takeover. Dynamics, reward, and termination support the critic, while the intervention head supplies actor-side risk shaping.

Dexterous teleoperation and synergies. Dexterous teleoperation systems retarget human hand motion through fingertip objectives, joint-space mappings, or optimization [27, 28, 29, 30]; DexRe-targeting [27] and Bidex Manus Teleop [30] are the closest general-purpose pipelines for our LEAP + Manus rig. WHIRL adapts this line of work to mid-rollout HIL with a morphology-routed retargeter, non-thumb affine joint mapping plus thumb fingertip IK, and command rebasing at each takeover. Following hand-synergy priors [31, 32, 33], WHIRL restricts the online residual to a low-dimensional coordination subspace.

3 Method

WHIRL combines three components illustrated in Figure 2: a tri-channel HIL teleoperation stack that produces clean intervention labels (Section 3.1), a residual-RL backbone over a behavior prior with hand exploration restricted to a low-dimensional synergy basis (Section 3.2), and a 4-head intervention-aware world model that turns takeover labels into actor-side risk shaping (Section 3.3).

3.1 Human-in-the-Loop Teleoperation System

WHIRL collects HIL data with a tri-channel teleoperation stack that supports continuous mid-rollout takeovers and logs each pedal state as an intervention label.

Tri-channel control. A 3Dconnexion SpaceMouse drives the Franka end-effector, a Manus Quantum glove drives the hand, and a foot pedal toggles between autonomous execution and human teleoperation. The pedal state simultaneously labels every transition with a binary indicator i_t that serves both as an evaluation metric and as supervision for the world-model intervention head.

Morphology-routed hand retargeter. On our LEAP + Manus rig, per-frame full-hand fingertip IK (e.g. DexRetargeting [27]) is robust but introduces visible latency at the 120 Hz glove rate, while direct joint-space mapping (e.g. Bidex Manus Teleop [30]) is zero-latency but loses thumb–index opposition. We therefore route morphology-by-morphology: the non-thumb fingers (index, middle, ring) use a per-joint affine map fit once from three calibration postures, while the 4-DoF thumb is solved by workspace fingertip IK augmented with an *opposition-direction term* that recovers the power-grasp wrap position-only tracking misses.

Intervention rebase for continuity. Let t_0 be the takeover time, $\mathbf{g}_t$ the Manus glove state, $R(\cdot)$ the per-frame hand retargeter, and $\mathbf{q}_{t_0}^{L,\text{robot}}$ the LEAP joint state at takeover. We remove command jumps with a Δ-joint rebase:

$$\mathbf{q}_t^{L,\text{cmd}} = \mathbf{q}_{t_0}^{L,\text{robot}} \; + \; \big(R(\mathbf{g}_t) - R(\mathbf{g}_{t_0})\big), \tag{1}$$

Thus the first command equals the current robot hand pose, and later commands apply only the retargeted glove displacement since takeover. The rebase is independent of the particular retargeter.

3.2 Residual RL Based on a Behavior Prior

WHIRL performs residual reinforcement learning [34, 7] over a frozen chunk-based behavior prior π_{BC} fit to teleoperated demonstrations. The executed action is $\mathbf{a}_t = \mathbf{a}^{\text{BC}}(\mathbf{o}_t) + \Delta_\theta(\mathbf{o}_t)$, and only the residual Δ_θ is updated online. The replay buffer $\mathcal{D} = \mathcal{D}_{\text{demo}} \cup \mathcal{D}_{\text{auto}} \cup \mathcal{D}_{\text{int}}$ stores autonomous and intervention transitions with pedal labels $i_t \in \{0, 1\}$; intervention transitions store the operator command rather than the policy-composed action. Section 3.3 turns these labels into takeover-prediction targets, and Section 4.5 isolates their actor-side effect.

Random perturbations in the raw 16-DoF hand joint space mostly break the coordinated postures the prior has learned. We therefore restrict the online hand residual to a low-dimensional postural synergy basis [31, 33] $\mathbf{W} \in \mathbb{R}^{K\times 16}$ shared across tasks. The residual is $\Delta_\theta(\mathbf{o}_t) = \big[\Delta\mathbf{a}_\theta^{\text{arm}}(\mathbf{o}_t),\ \mathbf{W}^\top \mathbf{z}_\theta^{\text{hand}}(\mathbf{o}_t)\big]$, which maps the low-dimensional hand update back to the LEAP joint space. The actor is optimized to improve return while remaining a local correction around the prior:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_{\mathbf{o}_t\sim\mathcal{D}}\big[Q_\phi\big(\mathbf{o}_t,\, \mathbf{a}_t^{\text{BC}} + \Delta_\theta(\mathbf{o}_t)\big)\big] + \alpha_{\text{arm}}\|\Delta\mathbf{a}_\theta^{\text{arm}}\|_2^2 + \alpha_{\text{hand}}\|\mathbf{z}_\theta^{\text{hand}}\|_2^2, \tag{2}$$

with the hand regularizer applied in synergy space, so each penalty unit corresponds to a coordinated postural change instead of a single-joint deviation.

3.3 Intervention-Aware World Model

The world model operates on the critic encoder's detached latent $z_t = E(s_t)$. Given (z_t, a_t), a dynamics ensemble predicts $\hat{z}_{t+1}^{(m)} = z_t + g_m(z_t, a_t)$ for $m = 1, \ldots, M$, while scalar heads predict reward probability $\hat{r}_{t+1}$, termination $\hat{d}_{t+1}$, and next-step intervention probability $\hat{p}_{t+1}^{\text{intv}}$. Ensemble disagreement $u(z_t, a_t) = \text{std}_m \hat{z}_{t+1}^{(m)}$ provides an epistemic uncertainty estimate. The heads are trained jointly on replay with

$$\mathcal{L}_{\text{WM}} = \text{MSE}(\hat{z}_{t+1}, z_{t+1}) + \text{BCE}(\hat{r}_{t+1}, \mathbf{1}[r_{t+1} > 0]) + \text{BCE}(\hat{d}_{t+1}, d_{t+1}) + m_t \cdot \text{BCE}(\hat{p}_{t+1}^{\text{intv}}, i_{t+1}) \tag{3}$$

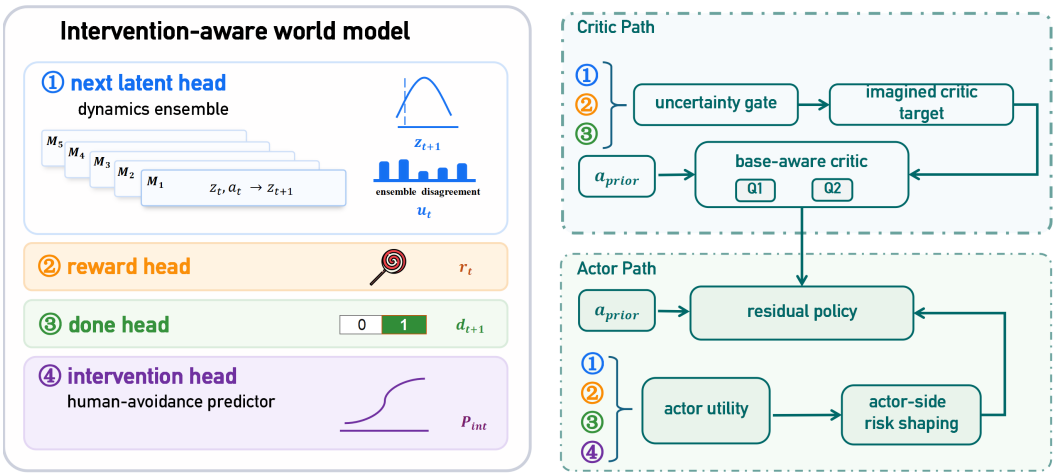

Figure 3: **Intervention-aware world model.** Dynamics, reward, and termination heads support uncertainty-gated critic updates, while the intervention head provides actor-side risk shaping.

where $m_t \in \{0, 1\}$ is a label-availability mask for the next-step pedal signal (set to 0 for replay segments without logged takeover state), and $\mathbf{1}[r_{t+1} > 0]$ is the binary target for the reward head. The intervention target i_{t+1} is the takeover label at the predicted next step, so the head is trained as a forward predictor rather than a per-step classifier of the current state.

Critic path. The critic is *base-aware*: it consumes the executed action together with the behavior-prior action as concatenated features, $Q(z, [\mathbf{a}, \mathbf{a}^{\mathrm{BC}}])$, so each value estimate is conditioned on the prior action being corrected. For each real transition, the WM produces $(\hat{z}_{t+1}, \hat{r}, \hat{d})$ and an uncertainty gate $g_u = \sigma((\tau - u)/\beta)$. Using the actor proposal $\tilde{\mathbf{a}}_t \sim \pi_\theta(\mathbf{o}_t)$ and the current prior action $\mathbf{a}_t^{\mathrm{BC}}$, the imagined target and critic auxiliary loss are

$$\begin{aligned} y^{\mathrm{imag}} &= \hat{r} + \gamma(1 - \hat{d}) \min_k Q_k^{\mathrm{tgt}}(\hat{z}_{t+1}, [\tilde{\mathbf{a}}_t, \mathbf{a}_t^{\mathrm{BC}}]), \\ \mathcal{L}_{\mathrm{critic}}^{\mathrm{WM}} &= \mathbb{E}_{\mathcal{D}}\left[g_u \left\| Q_\phi(z_t, [\tilde{\mathbf{a}}_t, \mathbf{a}_t^{\mathrm{BC}}]) - y^{\mathrm{imag}} \right\|_2^2 \right]. \end{aligned} \tag{4}$$

The full critic objective adds this term to the standard real-transition Bellman loss: $\mathcal{L}_{\mathrm{critic}} = \mathcal{L}_{\mathrm{critic}}^{\mathrm{TD}} + \lambda_c \mathcal{L}_{\mathrm{critic}}^{\mathrm{WM}}$. The takeover-probability head is intentionally excluded from this Bellman target, so operator habits do not directly rewrite the critic's return estimate.

Actor path. The actor consumes the intervention head through a model-derived utility:

$$J_{\mathrm{WM}}(z, a) = \hat{r} - \alpha_d \hat{d} - \alpha_i \hat{p}^{\mathrm{intv}} - \alpha_u u(z, a), \tag{5}$$

The corresponding auxiliary actor loss is

$$\mathcal{L}_{\mathrm{actor}}^{\mathrm{WM}} = -\mathbb{E}_{\mathcal{D}}[g_u \, J_{\mathrm{WM}}(z_t, \pi_\theta(o_t))], \tag{6}$$

The total actor objective is $\mathcal{L}_{\mathrm{actor}}^{\mathrm{total}} = \mathcal{L}_{\mathrm{actor}} + \lambda_a \mathcal{L}_{\mathrm{actor}}^{\mathrm{WM}}$, with $\mathcal{L}_{\mathrm{actor}}$ from Eq. 2. Model gradients are detached, so this loss shapes the actor without distorting WM training. The $\hat{r}$ term uses the calibrated reward-head output, while the termination, intervention, and uncertainty terms use raw probabilities or disagreement. Setting $\alpha_i = 0$ cleanly removes the only HIL-specific actor penalty.

Risk-shaping intuition. Eq. 5 gives the actor three penalties: termination for drop/slip failures, intervention probability for states the operator would take over, and ensemble disagreement for epistemic outliers. The new term is $\hat{p}^{\mathrm{intv}}$: because operators usually intervene before terminal failure, it supplies an earlier actor-side risk signal than $\hat{d}$ alone. WM gradients are detached in both actor and critic use, so shaping errors affect exploration without training the WM through its own predictions.

Design choice: actor shaping over reward bonus. We use $\hat{p}^{\mathrm{intv}}$ as actor-side shaping rather than an environment reward bonus because it predicts operator behavior, not task cost. Putting it into the reward would propagate operator habits through Bellman backups and entangle WM bias with value learning.

4 Experiments

We organize evaluation around three questions: **Q1:** final task success; **Q2:** operator-controlled training steps; and **Q3:** whether the gain comes from actor-side use of the intervention predictor.

4.1 Tasks

Our hardware setup comprises a Franka FR3 arm with a 16-DoF LEAP Hand [35] as the end-effector, two RGB cameras (one wrist-mounted, one third-person), and the tri-channel teleoperation rig described in Section 3.1. We evaluate on five tasks of increasing difficulty in the real world, as shown in Fig. 4: **Pick Cube** (convex cube with regular grasp affordances; geometric complement to Pick LEGO), **Pick LEGO** (irregular non-convex object with narrow grasp affordances), **Pick & Place** (grasp and placement at a target region), **Pull Drawer** (handle grasp and prismatic pull), and **Pull Drawer + Place + Close** (**Long Horizon**; pull the drawer open, place an object inside, close it).

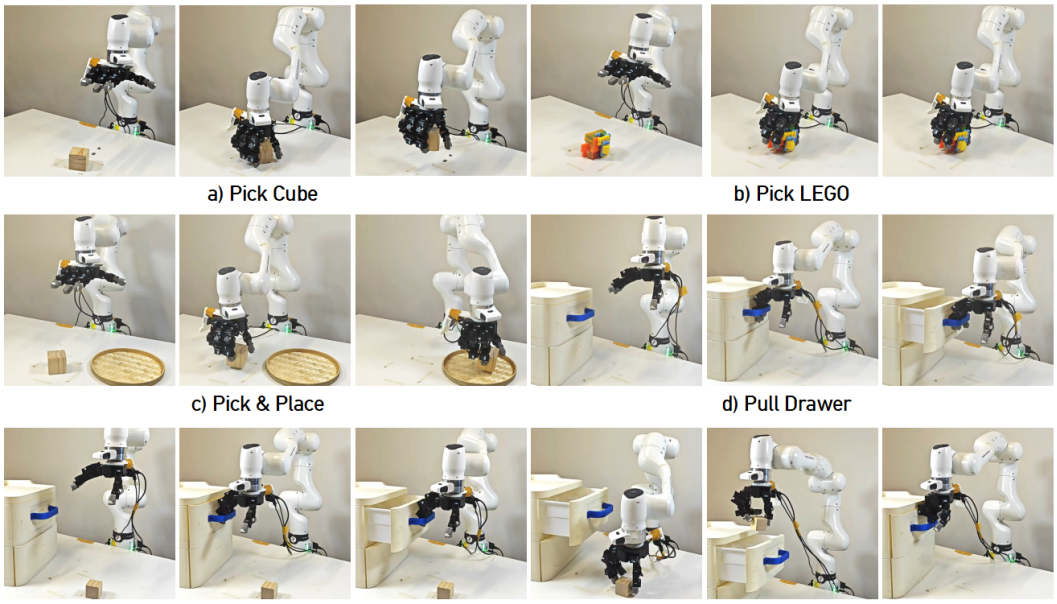

Figure 4: **Real-robot task suite.** Five LEAP-Hand tasks spanning grasping, placement, prismatic manipulation, and the three-stage Long Horizon sequence.

4.2 Baselines

We frame the comparison as a *progressive ladder*: four policies share the same observation and action space, and each row adds one mechanism to the previous one:

$$\text{Behavior prior} \;\rightarrow\; \text{ResFiT} \;\rightarrow\; \text{Residual RL (model-free)} \;\rightarrow\; \textbf{WHIRL}.$$

Behavior prior [4] π_{BC}: frozen chunk-based imitation policy on $\mathcal{D}_{\text{demo}}$. **ResFiT** [25]: autonomous residual RL on top under a sparse binary task reward, no operator interventions or world model. **Residual RL (model-free)**: our HIL baseline, which adds takeover-aware critic training to the residual actor but uses no world model; it follows the core residual-HIL mechanism used by HIL-SERL [14] and SiLRI [15]. **WHIRL (ours)**: the full pipeline.

4.3 Metrics

We report one headline trace metric and one offline success-rate measurement, and define the raw per-episode signal from which the trace is computed. **Rolling intervention fraction (5-episode,**

step-weighted) is the headline trace metric. For episode i with u_i operator-controlled steps and length T_i, $\rho_k = \left(\sum_{i=k-4}^{k} u_i\right)/\left(\sum_{i=k-4}^{k} T_i\right)$ measures the recent fraction of robot steps under human control; Figure 5 plots this smoothed trace. **Per-episode intervention fraction** u_i/T_i is the raw signal used to compute ρ_k. **Real-robot success rate**: per-method, per-task fraction over N rollouts of the converged policy under the same in-region randomization as training, reported in Table 1. Only the two HIL residual methods face an active takeover channel; BC and ResFiT run with the pedal disabled, so Figure 5 contrasts Residual RL (WHIRL w/o WM) and WHIRL.

4.4 Training and evaluation details

Each method starts from the same action-chunk behavior-cloning prior [4] fit to 15–30 teleoperated demonstrations per task and frozen during online learning, then trains online until trace metrics plateau on each (task, method) cell. We evaluate *in-region generalization*: object-pose, approach-offset, and task-specific handle-position variation within the same workspace markers used for demonstrations. During training, one operator follows a fixed rule shared across methods: take over only when the policy is about to violate workspace bounds or drop the object, and release once the failure mode is averted. For Long Horizon, evaluation is end-to-end rather than stage-wise: the policy must open the drawer, place the object, and close the drawer in a single rollout without per-stage resets.

Fully autonomous success-rate evaluation. Table 1 is measured on held-out deployments with the pedal disabled and the operator present only for emergency stop, using the checkpoint closest to the per-task nominal budget for each cell. Timeouts, drops, and workspace violations count as failures. This deployment is stricter than the HIL training trace because the operator may emergency-stop but cannot rescue or correct the trajectory.

Evaluation rigor under a single seed. Real-robot training cost makes seed averaging prohibitive, so each (task, method) cell uses one seed and one operator, as in prior real-robot RL studies [10, 14, 15]. We compensate with randomized in-region resets, step-weighted rolling traces, and held-out autonomous success evaluation. The four-method ladder shares the same seed/operator pair within each task, so row-to-row differences isolate the added mechanism.

4.5 Main Results

Q1. Final task success. WHIRL improves over the strongest baseline (Residual RL model-free) on every task by +15–30 percentage points (Table 1). Fisher's exact reaches $p < 0.05$ on the three pick tasks; Pull Drawer and Long Horizon are suggestive at $p \approx 0.08$. ResFiT and Residual RL differ by at most one trial per task, so the consistent gap appears when the world-model row is added.

Task	BC	ResFiT	Residual	WHIRL
Cube	20/30 66.7%	25/30 83.3%	25/30 83.3%	**30/30** **100.0%**
LEGO	19/30 63.3%	24/30 80.0%	23/30 76.7%	**29/30** **96.7%**
P&P	13/30 43.3%	19/30 63.3%	18/30 60.0%	**27/30** **90.0%**
Drawer	11/20 55.0%	14/20 70.0%	15/20 75.0%	**19/20** **95.0%**
Long Horizon	7/20 35.0%	13/20 65.0%	12/20 60.0%	**17/20** **85.0%**

Table 1: **Performance.** Each cell shows successes/rollouts over success rate.

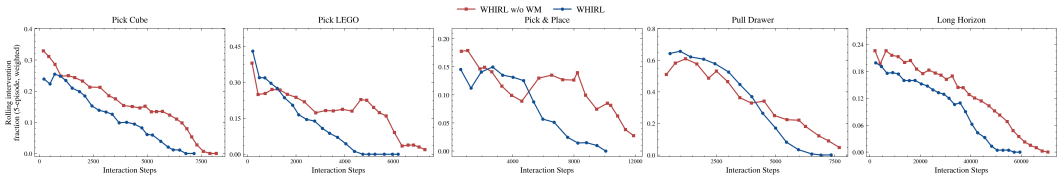

Figure 5: **Main result: rolling intervention fraction (lower is better).** Step-weighted rolling-5 intervention fraction ρ_k across five real-robot tasks. WHIRL stays below the no-WM baseline and reaches the near-zero regime earlier; success rates are in Table 1.

Q2: Operator intervention burden. WHIRL stays below the model-free baseline on the rolling-5 trace for every task (Figure 5). On pick tasks, both methods eventually reach $\rho_k = 0$, but WHIRL gets there earlier. On Pull Drawer and Long Horizon, the step-weighted curve separates the methods despite noisy raw traces. The step-weighted view matters because a brief pedal tap and a long recovery should not count equally. The largest relative drop is on Pull Drawer, where WHIRL reduces the final rolling fraction from the baseline's 0.113 to 0.018 (84%). Since autonomous success improves in the same direction, lower intervention burden is not caused by early termination or giving up. Operationally, the operator shifts from frequent recovery to rare boundary correction. Under the fixed operator protocol, a lower step-weighted fraction means fewer prolonged rescue segments, not merely fewer pedal events. On Pull Drawer, the remaining interventions are brief reach-margin corrections rather than sustained recoveries.

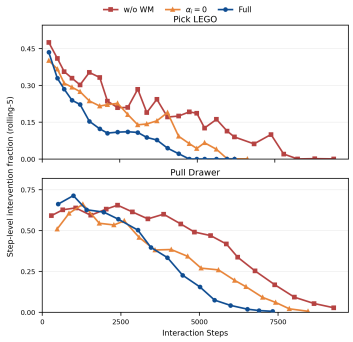

Figure 6: **Mechanism ablation.** Removing actor-side intervention shaping or the WM delays convergence.

Q3: Source of the gain. We isolate the mechanism on Pick LEGO (multi-finger grasping) and Pull Drawer (prismatic contact). *WHIRL ($\alpha_i = 0$)* trains the world model but blocks actor use of the intervention head; *WHIRL w/o WM* removes the model pathway. Full WHIRL reaches the near-zero intervention regime first on both tasks. The $\alpha_i = 0$ curve stays above full WHIRL even though the critic-facing scaffold remains, so the gain is not just auxiliary BCE training. WHIRL w/o WM is slowest, confirming that the model pathway also carries part of the burden. Pull Drawer shows a brief early cost before crossover, consistent with risk shaping helping only after the intervention head fits enough replay.

5 Limitations

WHIRL remains bounded by real-robot HIL constraints. Evaluation covers *in-region* pose and approach randomization rather than new objects, layouts, or fine in-hand manipulation. Training uses one seed and one operator per cell. The Pull Drawer and Long Horizon success gaps remain suggestive at current sample sizes, with the mechanism ablation providing complementary evidence. The intervention head also inherits the takeover threshold and habits of the labeling operator, so deployment with a substantially more or less cautious operator may require recalibration. Natural next settings include precision in-hand reorientation and fingertip gaiting on different hand morphologies or tactile-equipped platforms, where failures may arise from subtle contact-quality changes rather than gross drops.

6 Conclusion

WHIRL turns takeovers into reusable risk predictions for actor-side shaping. Across five real-robot dexterous tasks, it improves autonomous success by 15–30 percentage points over the strongest baseline (Residual RL model-free) and reduces operator-controlled training steps by up to 84%. The result suggests a practical recipe for dexterous HIL-RL: keep corrections on the real robot, learn when the operator would intervene, and route that prediction to the actor rather than the Bellman target. More broadly, WHIRL treats operator attention as reusable supervision instead of a one-off rescue signal. **To our knowledge, WHIRL is the first HIL-RL system to use per-state takeover prediction as an actor-side risk signal on a 16-DoF dexterous hand.**

References

[1] M. Andrychowicz, B. Baker, M. Chociej, et al. Learning dexterous in-hand manipulation. *International Journal of Robotics Research (IJRR)*, 39(1):3–20, 2020. doi:10.1177/0278364919887447.

[2] A. Handa et al. DeXtreme: Transfer of agile in-hand manipulation from simulation to reality. *arXiv preprint arXiv:2210.13702*, 2022.

[3] T. Chen, M. Tippur, S. Wu, V. Kumar, E. H. Adelson, and P. Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023. doi:10.1126/scirobotics.adc9244.

[4] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023. doi:10.15607/RSS.2023.XIX.016.

[5] C. Chi, Z. Xu, S. Feng, E. Cousineau, et al. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. doi:10.15607/RSS.2023.XIX.026.

[6] A. Rajeswaran, V. Kumar, A. Gupta, et al. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems (RSS)*, 2018.

[7] T. Johannink, S. Bahl, A. Nair, J. Luo, et al. Residual reinforcement learning for robot control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6023–6029, 2019. doi:10.1109/ICRA.2019.8794127.

[8] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning (ICML)*, 2023.

[9] H. Zhang, G. Solak, G. J. Lahr, and A. Ajoudani. Srl-vic: A variable stiffness-based safe reinforcement learning for contact-rich robotic tasks. *IEEE Robotics and Automation Letters*, 9(6):5631–5638, 2024.

[10] J. Luo, Z. Hu, C. Xu, Y. L. Tan, et al. SERL: A software suite for sample-efficient robotic reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 16961–16969, 2024. doi:10.1109/ICRA57147.2024.10610275.

[11] H. Hu, S. Mirchandani, and D. Sadigh. IBRL: Imitation bootstrapped reinforcement learning. In *Robotics: Science and Systems (RSS)*, 2024.

[12] I. Akkaya, M. Andrychowicz, et al. Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[13] Y. Qin et al. DexPoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pages 594–605, 2023.

[14] J. Luo, C. Xu, J. Wu, and S. Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024.

[15] Y. Zhao, H. Jin, L. Jiang, X. Zhang, K. Wu, P. Ren, Z. Xu, Z. Che, L. Sun, D. Wu, C. H. Liu, and J. Tang. Real-world reinforcement learning from suboptimal interventions. *arXiv preprint arXiv:2512.24288*, 2025.

[16] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8077–8083, 2019. doi:10.1109/ICRA.2019.8793698.

[17] R. Hoque, A. Balakrishna, et al. ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In *Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pages 598–608, 2022.

[18] 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. doi:10.15607/RSS.2023.XIX.005.

[19] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

[20] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *Nature*, 640:354–359, 2025. doi:10.1038/s41586-025-08744-2.

[21] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[22] N. Hansen, H. Su, and X. Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022.

[23] Physical Intelligence. $\pi^*_{0.6}$: A VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

[24] Y. Wang, X. Li, P. Xie, P. Yang, B. Nie, Y. Cai, Q. Zhang, C. Qu, J. Wu, J. Song, X. Ren, J. Huang, M. Pan, S. Feng, Z. Chen, and J. Luo. Learning while deploying: Fleet-scale reinforcement learning for generalist robot policies. *arXiv preprint arXiv:2605.00416*, 2026.

[25] L. Ankile, Z. Jiang, R. Duan, G. Shi, P. Abbeel, and A. Nagabandi. Residual off-policy RL for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*, 2025.

[26] Y. Li, Z. Zhou, Y. Chen, Y. Guo, J. Liu, S. Zhang, J. Chen, and Y. Zhu. Hi-WM: Human-in-the-world-model for scalable robot post-training. *arXiv preprint arXiv:2604.21741*, 2026.

[27] Y. Qin et al. AnyTeleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems (RSS)*, 2023. doi:10.15607/RSS.2023.XIX.046.

[28] A. Handa, K. Van Wyk, et al. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170, 2020. doi:10.1109/ICRA40945.2020.9197124.

[29] C. Wang, H. Shi, et al. DexCap: Scalable and portable mocap data collection system for dexterous manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[30] K. Shaw, Y. Li, J. Yang, M. K. Srirama, R. Liu, H. Xiong, R. Mendonca, and D. Pathak. Bimanual dexterity for complex tasks. In *Conference on Robot Learning (CoRL)*, 2024.

[31] M. Santello, M. Flanders, and J. F. Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998. doi:10.1523/JNEUROSCI.18-23-10105.1998.

[32] M. T. Ciocarlie and P. K. Allen. Hand posture subspaces for dexterous robotic grasping. *International Journal of Robotics Research (IJRR)*, 28(7):851–867, 2009. doi:10.1177/0278364909105606.

[33] M. Ciocarlie, C. Goldfeder, and P. K. Allen. Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem. In *Robotics: Science and Systems (RSS)*, 2007.

[34] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

[35] K. Shaw, A. Agarwal, and D. Pathak. LEAP Hand: Low-cost, efficient, and anthropomorphic hand for robot learning. In *Robotics: Science and Systems (RSS)*, 2023. doi:10.15607/RSS.2023.XIX.089.

A Hand Retargeter Implementation Details

This appendix gives the closed-form expressions and solver hyperparameters of the hybrid hand retargeter summarized in Section 3.1. Let $\mathbf{g}_t$ denote the calibrated Manus glove state and $\mathbf{q}_t^L \in \mathbb{R}^{16}$ the commanded LEAP joints; superscripts L and G refer to the LEAP and glove frames, respectively.

Non-thumb fingers (per-joint affine map). For each finger $f \in \mathcal{F}_{\text{map}} = \{\text{index}, \text{middle}, \text{ring}\}$ and joint j, the commanded LEAP joint is

$$q^L_{f,j,t} = \text{clip}\big(\alpha_{f,j}\, g_{f,j,t} + \beta_{f,j},\ q^L_{f,j,\min},\ q^L_{f,j,\max}\big)\,, \tag{7}$$

with scalar gain $\alpha_{f,j}$ and offset $\beta_{f,j}$ fit once per operator from three reference postures (fully open, half-closed, fully closed) by two-parameter, three-point least squares; the closed-form solution is recomputed offline rather than online.

Thumb (workspace fingertip IK with opposition alignment). For the 4-DoF thumb chain, we first define three residuals:

$$\begin{aligned} \mathbf{r}_x(\mathbf{q}) &= \mathbf{x}^L_{\text{th}}(\mathbf{q}) - \mathbf{T}_{GL}\mathbf{x}^G_{\text{th}}(\mathbf{g}_t), \\ \mathbf{r}_o(\mathbf{q}) &= \hat{\mathbf{n}}^L(\mathbf{q}) - \mathbf{R}_{GL}\hat{\mathbf{n}}^G(\mathbf{g}_t), \\ \mathbf{r}_s(\mathbf{q}) &= \mathbf{q} - \mathbf{q}^L_{\text{th},t-1}. \end{aligned} \tag{8}$$

The commanded thumb posture is then the constrained minimizer

$$\mathbf{q}^L_{\text{th},t} = \arg\min_{\mathbf{q}\in[\mathbf{q}_{\min},\mathbf{q}_{\max}]} \|\mathbf{r}_x(\mathbf{q})\|_2^2 + \lambda_o\|\mathbf{r}_o(\mathbf{q})\|_2^2 + \lambda_{\text{smooth}}\|\mathbf{r}_s(\mathbf{q})\|_2^2, \tag{9}$$

where $\mathbf{r}_x$ tracks the thumb fingertip target, $\mathbf{r}_o$ aligns the thumb–index opposition direction, and $\mathbf{r}_s$ smooths frame-to-frame commands. Here $\mathbf{x}^L_{\text{th}}$ is the LEAP thumb forward kinematics, $\mathbf{x}^G_{\text{th}}$ the glove-derived fingertip target, $(\mathbf{T}_{GL}, \mathbf{R}_{GL})$ the translational and rotational parts of the one-shot palm-to-palm calibration from G to L, and $\hat{\mathbf{n}}^L(\mathbf{q}) = (\mathbf{x}^L_{\text{idx}} - \mathbf{x}^L_{\text{th}}(\mathbf{q}))/\|\cdot\|$, $\hat{\mathbf{n}}^G(\mathbf{g}_t) = (\mathbf{x}^G_{\text{idx}} - \mathbf{x}^G_{\text{th}})/\|\cdot\|$ are the thumb-tip $\to$ index-tip unit vectors in the two frames. The LEAP index-tip $\mathbf{x}^L_{\text{idx}}$ is taken from the affine command of Eq. 7 at the current frame and held fixed during the thumb solve, so $\hat{\mathbf{n}}^L$ depends on $\mathbf{q}$ only through the thumb-tip term.

Gauss–Newton derivation. Let $J = \partial\mathbf{x}^L_{\text{th}}/\partial\mathbf{q} \in \mathbb{R}^{3\times 4}$ be the position Jacobian of the LEAP thumb tip, $\mathbf{e} = \mathbf{T}_{GL}\mathbf{x}^G_{\text{th}}(\mathbf{g}_t) - \mathbf{x}^L_{\text{th}}(\mathbf{q}_k)$, $\mathbf{m} = \mathbf{R}_{GL}\hat{\mathbf{n}}^G(\mathbf{g}_t)$, $\mathbf{v} = \mathbf{x}^L_{\text{idx}} - \mathbf{x}^L_{\text{th}}(\mathbf{q}_k)$, and let $\mathbf{P} = I_3 - \hat{\mathbf{n}}^L\hat{\mathbf{n}}^{L\top}$ be the projector orthogonal to $\hat{\mathbf{n}}^L$. The opposition residual is $\mathbf{r}_o = \hat{\mathbf{n}}^L(\mathbf{q}) - \mathbf{m} \in \mathbb{R}^3$ with Jacobian $\partial\mathbf{r}_o/\partial\mathbf{q} = -(1/\|\mathbf{v}\|)\,\mathbf{P}\,J$. Linearizing all three terms of Eq. 9 around $\mathbf{q}_k$ and using $\mathbf{P}\hat{\mathbf{n}}^L = 0$ to simplify the gradient, the damped Gauss–Newton step solves the normal equations

$$\Big(J^\top J + \frac{\lambda_o}{\|\mathbf{v}\|^2}J^\top\mathbf{P}\,J + (\lambda_{\text{smooth}}{+}\mu^2)I_4\Big)\,\Delta\mathbf{q}_k \;=\; J^\top\mathbf{e} - \frac{\lambda_o}{\|\mathbf{v}\|}\,J^\top\mathbf{P}\,\mathbf{m} + \lambda_{\text{smooth}}\big(\mathbf{q}^L_{\text{th},t-1}{-}\mathbf{q}_k\big), \tag{10}$$

with Levenberg–Marquardt damping $\mu = 2\times10^{-2}$ for stability near singularities of the 4-DoF chain. The negative sign on the opposition RHS term follows from $\partial\hat{\mathbf{n}}^L/\partial\mathbf{q} = -(1/\|\mathbf{v}\|)\,\mathbf{P}\,J$ and $\mathbf{P}\hat{\mathbf{n}}^L = 0$: collecting the gradient of $\lambda_o\|\hat{\mathbf{n}}^L - \mathbf{m}\|^2$ at $\mathbf{q}_k$ and applying the GN step $\Delta\mathbf{q} = -H^{-1}\nabla L$ yields $-(\lambda_o/\|\mathbf{v}\|)J^\top\mathbf{P}\,\mathbf{m}$. The opposition contribution to the Hessian is rank-≤ 2 (it lives in the 2-D subspace orthogonal to $\hat{\mathbf{n}}^L$), so it never inflates the solve in directions along the opposition axis.

Solver schedule. Up to 40 iterations per control step, step size 0.3, per-joint clipping to LEAP limits, temporal smoothing weight $\lambda_{\text{smooth}} = 5\times10^{-2}$, opposition weight $\lambda_o = 2\times10^{-1}$. Solver state is warm-started from $\mathbf{q}^L_{\text{th},t-1}$, and we early-stop at $\|\mathbf{e}\|_2 < 10^{-3}$ m; under typical glove rates the solver converges in fewer than 10 iterations.

Intervention rebase formula. Let $R(\cdot)$ denote the per-frame retargeter (Eq. 7 on the non-thumb fingers, Eq. 9 on the thumb), evaluated for this rebase as a memoryless function of the current glove sample so that the rebase anchor and delta are deterministic in $\mathbf{g}_t$; the running solver of Eq. 9 still

retains its temporal warm-start during continuous control. After a takeover at time t_0 the command is

$$\mathbf{q}_t^{L,\mathrm{cmd}} = \mathbf{q}_{t_0}^{L,\mathrm{robot}} \; + \; \big(R(\mathbf{g}_t) - R(\mathbf{g}_{t_0})\big), \tag{11}$$

so the first commanded pose equals the current robot hand pose and subsequent commands carry only the joint-space *increment* of the retargeted posture since t_0.

B World Model and RL Hyperparameters

Table 2 lists the world-model and critic/actor hyperparameters used in all real-robot runs; values are shared across the five tasks unless noted. Ablations keep these values except for their named removals (e.g. $\alpha_i = 0$ or no WM). The daggered residual scales are paired with explicit L2 penalties on arm and hand residuals; these penalties keep early online updates local around the frozen behavior prior and are annealed once the residual policy reaches stable task execution. The exact per-task residual-penalty schedule is fixed before training and included in the task configuration files of the supplementary code.

Real-transition critic loss. The backbone critic is trained on real replay transitions with the standard clipped-double-Q Bellman target

$$\begin{aligned} y_t^{\mathrm{TD}} &= r_{t+1} + \gamma(1 - d_{t+1}) \min_k Q_k^{\mathrm{tgt}}\big(z_{t+1}, [\tilde{\mathbf{a}}_{t+1}, \mathbf{a}_{t+1}^{\mathrm{BC}}]\big)\,, \\ \mathcal{L}_{\mathrm{critic}}^{\mathrm{TD}} &= \mathbb{E}_{\mathcal{D}}\Big[\big\|Q_\phi(z_t, [\mathbf{a}_t, \mathbf{a}_t^{\mathrm{BC}}]) - y_t^{\mathrm{TD}}\big\|_2^2\Big]\,, \end{aligned} \tag{12}$$

where $\tilde{\mathbf{a}}_{t+1} \sim \pi_\theta(\cdot \mid z_{t+1})$ is a reparameterized sample from the current actor, $\mathbf{a}_{t+1}^{\mathrm{BC}}$ is the frozen behavior-prior action at the next state, and z_{t+1} is the encoded real next observation. The standard SAC entropy term $-\alpha \log \pi_\theta(\tilde{\mathbf{a}}_{t+1} \mid z_{t+1})$ is absorbed into y_t^{TD} as in clipped-double-Q SAC and is omitted from Eq. 12 for brevity. Section 3.3 adds the uncertainty-gated world-model auxiliary target on top of this loss.

Component	Hyperparameter	Value
World model	Latent dimension d_z	256
	Head MLP width / depth	2×512 hidden
	Dynamics ensemble size M	5
	Optimizer / learning rate	Adam / 1×10^{-4}
	Batch size	128
	Calibrated reward $(R_{\mathrm{succ}}, R_{\mathrm{step}})$	$(10.0, -0.01)$
Actor & Critic	Discount γ	0.97
	Imagined critic weight λ_c	0.05
	Actor WM weight λ_a	0.40
	Uncertainty soft gate (τ, β)	$(0.03, 0.01)$
	Actor utility coeffs. $(\alpha_d, \alpha_i, \alpha_u)$	$(1.0, 0.4, 1.0)$
Residual RL backbone	Synergy dimension K	3
	Residual scales (arm / synergy / hand)†	$(1.5 \times 10^{-3},\ 0.25,\ 0.15)$
	Replay capacity (online / offline / intervention)	(10k, 500, 5k)

Table 2: World-model and residual-RL hyperparameters used for reported WHIRL runs.

C Design Choices for the Intervention-Aware World Model

1-step deterministic-latent design vs. RSSM / Hi-WM. Each WM head in Section 3.3 is a small MLP on the same encoded latent z_t: the dynamics ensemble supplies an epistemic mask, the reward and termination heads form a 1-step Bellman target, and the intervention head turns the binary takeover label into forward supervision the actor can consume directly. RSSM-style world models [20] learn a recurrent latent and train on imagined rollouts of length 15–16, but on our small-data

real-robot budget we found multi-step imagination compounds per-step dynamics error on a 16-DoF hand without sharpening the 1-step Bellman target. Hi-WM [26] also pairs a world model with HIL, but as a sandbox in which the operator intervenes *inside* the model; WHIRL keeps every correction on the real robot and uses the model only as a 1-step data multiplier (critic) and a per-state risk predictor (actor).

WHIRL deliberately routes the takeover-probability head $\hat{p}^{\text{intv}}$ into the *actor objective alone*, rather than into the environment reward or the critic's Bellman target. This decomposition is the central design choice of the intervention-aware world model and rests on four properties, which we contrast against the natural alternative of using $\hat{p}^{\text{intv}}$ as a reward bonus $r'(s,a) = r(s,a) - \beta\,\hat{p}^{\text{intv}}(s,a)$.

Optimality invariance. $\hat{p}^{\text{intv}}$ is a learned *predictor* of operator behavior, not a true cost: it reflects *which states an operator chose to take over in the past*, which conflates real failure risk with operator habit, attention budget, and task framing. Used as a reward bonus, this signal changes the optimal policy and can push the actor into over-conservative regimes that avoid even free-space approach states that an operator never actually flagged at deployment time. Used in the auxiliary actor loss of Eq. 6, the same signal biases action updates through J_{WM} without redefining the task reward, and $\alpha_i \to 0$ exactly removes intervention shaping.

Decoupling from credit assignment. A reward bonus propagates through every Bellman backup, so $\hat{p}^{\text{intv}}$ silently re-weights the entire downstream value function and entangles WM bias with credit assignment. In our decomposition, the critic's Bellman target y^{imag} (Section 3.3) consumes only the dynamics, reward, and termination heads; the takeover-probability head touches the actor objective alone. Errors in $\hat{p}^{\text{intv}}$ therefore cannot leak into Q, and the critic's uncertainty-gated targets remain a faithful estimate of return under the calibrated reward.

No conflict with reward calibration. WHIRL already calibrates its sparse-binary task reward through the reward head, $\hat{r} = \hat{p}_r R_{\text{succ}} + (1-\hat{p}_r) R_{\text{step}}$ (Section 3.3, Appendix B). Adding a $-\beta\hat{p}^{\text{intv}}$ bonus would compete with this calibration on the same scale: β and $(R_{\text{succ}}, R_{\text{step}})$ would have to be co-tuned per task to prevent the shaping term from dominating success signal. Keeping the takeover term out of $\hat{r}$ preserves a single, calibrated reward channel. The reported $\alpha_i = 0.4$ is therefore a moderate actor-utility weight rather than a redefinition of the environment reward; the mechanism ablation tests its role by setting the weight to zero.

Clean ablation surface. The two roles of the WM are factored into two orthogonal knobs: removing the world model disables both the uncertainty-gated imagined critic target and actor-side shaping, while setting $\alpha_i = 0$ disables only actor-side intervention shaping and leaves the dynamics/reward/termination scaffold intact. This factoring is exactly what the three-line ablation in Section 4.5 exploits: *WHIRL w/o WM* measures the value of the WM scaffold, and *WHIRL* ($\alpha_i = 0$) isolates the value of actor consumption of the intervention predictor. A reward-bonus design collapses both pathways into the shaping coefficient β and prevents this attribution.

D Detailed Training and Evaluation Protocol

This appendix expands the protocol summarized in Section 4.4.

Demonstrations and replay seeding. Each task uses 15–30 teleoperated demonstrations collected with the tri-channel interface of Section 3.1; the same set both fits the frozen action-chunk behavior-cloning prior [4] and seeds the replay buffer used by the world model. No additional offline data are introduced during online training.

Single seed and within-seed smoothing. Because each (task, method) cell costs hours of real-robot interaction time, we report a single training seed per cell and rely on trailing 5-episode smoothing (Section 4.3) to absorb within-seed noise. Trace metrics in Figures 5 and 6 tally interaction steps

at episode termination, so final logged points may extend past the per-task nominal checkpoint (≈8,000 steps for the pick and drawer tasks, ∼50,000 for Long Horizon); the success-rate table (Table 1) instead uses the checkpoint closest to that budget on each cell so that all methods are compared at comparable training cost.

Operator standard operating procedure. A single human operator runs every (task, method) cell. Because the operator necessarily knows the active method during real-robot training, we hold the intervention rule fixed across conditions to keep labels comparable. Specifically the operator presses the pedal in exactly two situations: (i) the policy is about to violate the Cartesian workspace bounds listed in Table 3, or (ii) the policy has dropped the object or has visibly lost grasp closure with the object still graspable. The operator releases the pedal as soon as the failure mode is averted. No preemptive corrections, posture cleanup, or smoothing demonstrations are injected during online runs.

Implications for variance reporting. The combination of a single seed and a single operator with method awareness means our trace metrics lack across-seed bands. We rely on three complementary signals to mitigate this: (i) the step-weighted rolling intervention fraction (Figure 5, Section 4.3) is less volatile than the raw per-episode fraction (Appendix E) and is therefore more discriminative when the operator's role shrinks; (ii) the success-rate evaluation in Table 1 does not depend on per-step takeover decisions and so removes operator bias from the headline outcome; and (iii) the four-row baseline ladder of Table 1 lets each cell share the same operator and seed pair, so the *differences* between rows isolate the corresponding mechanism even if the absolute level reflects a particular run.

Statistical analysis for Table 1. We report two complementary statistics per task: a Wilson score 95% confidence interval for WHIRL's success rate (small-N appropriate; binomial), and a one-sided uncorrected Fisher's exact p-value for the WHIRL-vs-Residual RL contrast (testing whether WHIRL succeeds strictly more often than the strongest no-WM baseline given the row/column totals). No multiple-comparison correction is applied (five tasks), so the reported p-values should be read as per-task evidence rather than as a family-wise significance.

Task	WHIRL Wilson 95% CI	Fisher one-sided p (WHIRL vs. Residual RL)
Pick Cube	$[89, 100]\%$	0.026
Pick LEGO	$[83, 99]\%$	0.026
Pick & Place	$[74, 97]\%$	0.008
Pull Drawer	$[76, 99]\%$	0.091
Long Horizon	$[64, 95]\%$	0.078

The three pick tasks (each at $N = 30$) reach $p < 0.05$; Pull Drawer and Long Horizon (each at $N = 20$) are in the suggestive regime ($p \approx 0.08$–0.09), consistent with the smaller sample size on those two tasks rather than a smaller underlying effect. The wider lower bound on Long Horizon's WHIRL CI (64%) compared with Pull Drawer (76%) reflects Long Horizon's $\hat{p} = 17/20 = 85\%$ being further from 1 than Pull Drawer's $19/20 = 95\%$ at the same N.

Baseline implementation rationale. The *Residual RL (model-free)* baseline (Section 4.2) is our in-house implementation of the HIL residual-actor + intervention-aware-critic mechanism shared by HIL-SERL [14] and SiLRI [15], rather than the published codebases. The motivation is implementation-difference isolation: the published HIL-SERL and SiLRI codebases target different end-effectors (parallel-jaw grippers and bimanual setups) and different observation spaces from our 16-DoF LEAP Hand + dual-RGB setup, so running them verbatim would conflate any gap with hardware/observation mismatch rather than algorithmic content. By sharing the behavior prior, residual actor, synergy-space residual parameterization, replay buffer, and HIL pedal channel with WHIRL, we make the ResFiT → Residual RL (model-free) → WHIRL ladder a clean three-step isolation of (i) autonomous residual RL on top of the prior, (ii) adding the HIL pedal channel into

the critic, and (iii) adding the intervention-aware world model. Each row differs from the previous by exactly one mechanism, which is what the row-to-row gaps in Table 1 report.

E Raw Per-Episode Intervention Traces

Figure 5 in the main paper reports the step-weighted 5-episode rolling intervention fraction ρ_k. Figure 7 below shows the underlying per-episode fraction u_i/T_i from which ρ_k is deterministically derived, for the same five tasks and the same WHIRL vs. WHIRL w/o WM contrast. The raw signal is noisier episode-to-episode (single-episode fluctuations in T_i amplify into u_i/T_i), which is exactly the variance the rolling-5 metric absorbs; both curves converge to the same near-zero regime once the policy stops requiring operator help. We keep this raw view in the appendix because it confirms the rolling-5 trace is not hiding a slow drift, while leaving the main paper figure uncluttered.

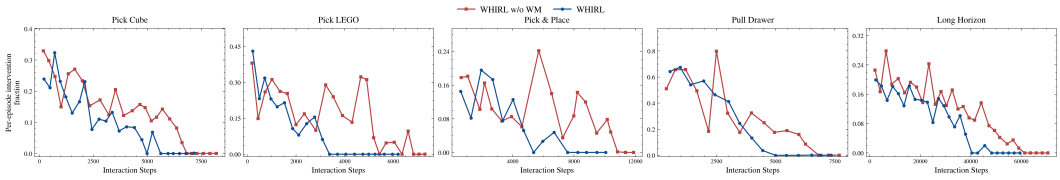

Figure 7: **Raw per-episode intervention fraction u_i/T_i (lower is better).** Same five tasks and same WHIRL (blue, circles) vs. WHIRL w/o WM (red, squares; = Residual RL model-free) contrast as Figure 5. The step-weighted rolling-5 aggregation of this signal is the headline metric in Figure 5.

F Hardware, Workspace, and Data Collection

Robot and sensors. A Franka FR3 arm with a 16-DoF LEAP Hand [35] is observed by wrist-mounted D405 and third-person D435 RGB cameras at 30 Hz. The arm runs at 20 Hz in Cartesian impedance mode; the LEAP Hand and Manus glove stream at 120 Hz.

Workspace and reset. Table 3 lists the in-region randomization, reset pose, workspace bounds, and horizon for the five tasks; the dagger marks Pick Cube as the convex-object control and Long Horizon as the three-stage task. Reset orientations are end-effector-down (RPY $\approx (180, -2, 89)$ deg for pick-family tasks, $(-179, 3, 19)$ deg for drawer-family tasks). Workspace bounds are in the FR3 `link8` FK frame from raw teleop episodes (p01–p99 plus a 2 cm margin). Each behavior prior uses 15–30 demonstrations; online residual RL runs for ≈8 k steps on pick/drawer tasks and ∼50 k on Long Horizon using one RTX 4090.

Property	Pick Cube†	Pick LEGO	Pick & Place	Pull Drawer	Long Horizon†
Obj. rand.	±15 cm	±15 cm	±15 cm	±5 cm (handle)	±5 cm
Reset EEF (m)	$(0.34, 0.09, 0.46)$	$(0.34, 0.09, 0.46)$	$(0.34, 0.09, 0.46)$	$(0.42, 0.16, 0.55)$	$(0.42, 0.16, 0.54)$
WS x (m)	$[0.32, 0.62]$	$[0.31, 0.58]$	$[0.32, 0.60]$	$[0.39, 0.51]$	$[0.35, 0.58]$
WS y (m)	$[-0.08, 0.16]$	$[-0.02, 0.12]$	$[-0.03, 0.31]$	$[-0.03, 0.18]$	$[-0.12, 0.26]$
WS z (m)	$[0.19, 0.48]$	$[0.19, 0.48]$	$[0.19, 0.48]$	$[0.38, 0.57]$	$[0.21, 0.64]$
Horizon	300 / 15 s	300 / 15 s	600 / 30 s	450 / 22.5 s	2100 / 105 s

Table 3: Task randomization, reset pose, workspace bounds, and episode horizon.

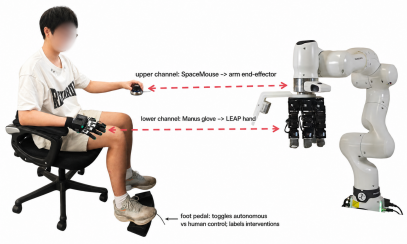

Figure 8: **Tri-channel HIL teleoperation rig:** SpaceMouse (arm), Manus glove (hand), and foot pedal (takeover).