RL-100: Performant Robotic Manipulation with Real-World Reinforcement Learning

Kun Lei[1,2,*], Huanyu Li[1,2,*], Dongjie Yu[1,3,*], Zhenyu Wei[5,*], Lingxiao Guo[6], Zhennan Jiang[7], Ziyu Wang[4], Shiyu Liang[2] and Huazhe Xu[1,4]

Abstract

Real-world robotic manipulation in homes and factories demands reliability, efficiency, and robustness that approach or surpass skilled human operators. We present RL-100, a real-world reinforcement learning training framework built on diffusion visuomotor policies trained by supervised learning. RL-100 introduces a three-stage pipeline. First, imitation learning leverages human priors. Second, iterative offline reinforcement learning uses an Offline Policy Evaluation procedure, abbreviated OPE, to gate PPO-style updates that are applied in the denoising process for conservative and reliable improvement. Third, online reinforcement learning eliminates residual failure modes. An additional lightweight consistency distillation head compresses the multi-step sampling process in diffusion into a single-step policy, enabling high-frequency control with an order-of-magnitude reduction in latency while preserving task performance. The framework is task-, embodiment-, and representation-agnostic and supports both 3D point clouds and 2D RGB inputs, a variety of robot platforms, and both single-step and action-chunk policies. We evaluate RL-100 on seven real-robot tasks spanning dynamic rigid-body control, such as Push-T and Agile Bowling, fluids and granular pouring, deformable cloth folding, precise dexterous unscrewing, and multi-stage orange juicing. RL-100 attains 100% success across evaluated trials for a total of 900 out of 900 episodes, including up to 250 out of 250 consecutive trials on one task. The method achieves near-human teleoperation or better time efficiency and demonstrates multi-hour robustness with uninterrupted operation lasting up to two hours. The resulting policies generalize zero-shot to novel dynamics with an average success of 92.5% and adapt in a few-shot fashion to significant task variations, reaching an average of 86.7% after one to three hours of additional training. These results suggest a practical path to deployment-ready robot learning by starting from human priors, aligning training objectives with human-grounded metrics, and reliably extending performance beyond human demonstrations. For more results and videos, please visit our project website: https://lei-kun.github.io/RL-100/.

Keywords

Real-world Reinforcement Learning, Diffusion Policy, Robotic Manipulation, Offline-to-Online RL, Visuomotor Control

Introduction

Dexterous robotic manipulation stands as an iconic challenge in robotics (Cui and Trinkle 2021; Luo et al. 2025). Real-world deployment beyond laboratories requires human-level reliability, efficiency, and robustness. Recent learning-based advances across generative diffusion policies (Chi et al. 2023; Ze et al. 2024), diffusion-based robot foundation models (Black et al. 2024a; Intelligence et al. 2025), sim-to-real RL (Yuan et al. 2025; Lin et al. 2025), and real-world RL (Luo et al. 2025, 2024a) have narrowed this gap, demonstrating human-like manipulation proficiency. In particular, generative policies and foundation models are trained or fine-tuned on high-quality, human-collected real-robot datasets at varying scales, providing strong human priors and enabling robots to acquire the efficient strategies used by skilled tele-operators. However, high-quality real-robot data remain scarce: teleoperation incurs perceptual and control latency and favors slow, conservative motions (Guo et al. 2025). Moreover, large-scale collection depends on skilled operators and is labor-intensive and expensive. As a result, state–action coverage is limited, undermining generalization and reliability. Consequently, this supervised paradigm is constrained by an imitation ceiling: under purely supervised objectives, performance is effectively bounded by demonstrator skill and inherits human inefficiencies, biases, and occasional errors.

Reinforcement learning (RL) offers a complementary route by optimizing returns from interaction rather than imitation fidelity, enabling the discovery of strategies that are rare or absent in human demonstrations. At the same time, sim-to-real RL must contend with visual and dynamics gaps between simulation and reality, while naïvely training a learning-based generative policy on real hardware is risky and sample-inefficient. This raises a central question: how can we build a robotic learning system that leverages strong human priors yet continues to refine itself through autonomous exploration? A useful analogy comes from human learning: babies learn to walk under parental

[1]Shanghai Qizhi Institute, China
[2]Shanghai Jiao Tong University, China
[3]The University of Hong Kong, HKSAR, China
[4]IIIS, Tsinghua University, China
[5]University of North Carolina at Chapel Hill, USA
[6]Carnegie Mellon University, USA
[7]Chinese Academy of Sciences, China
[*]Equal contribution

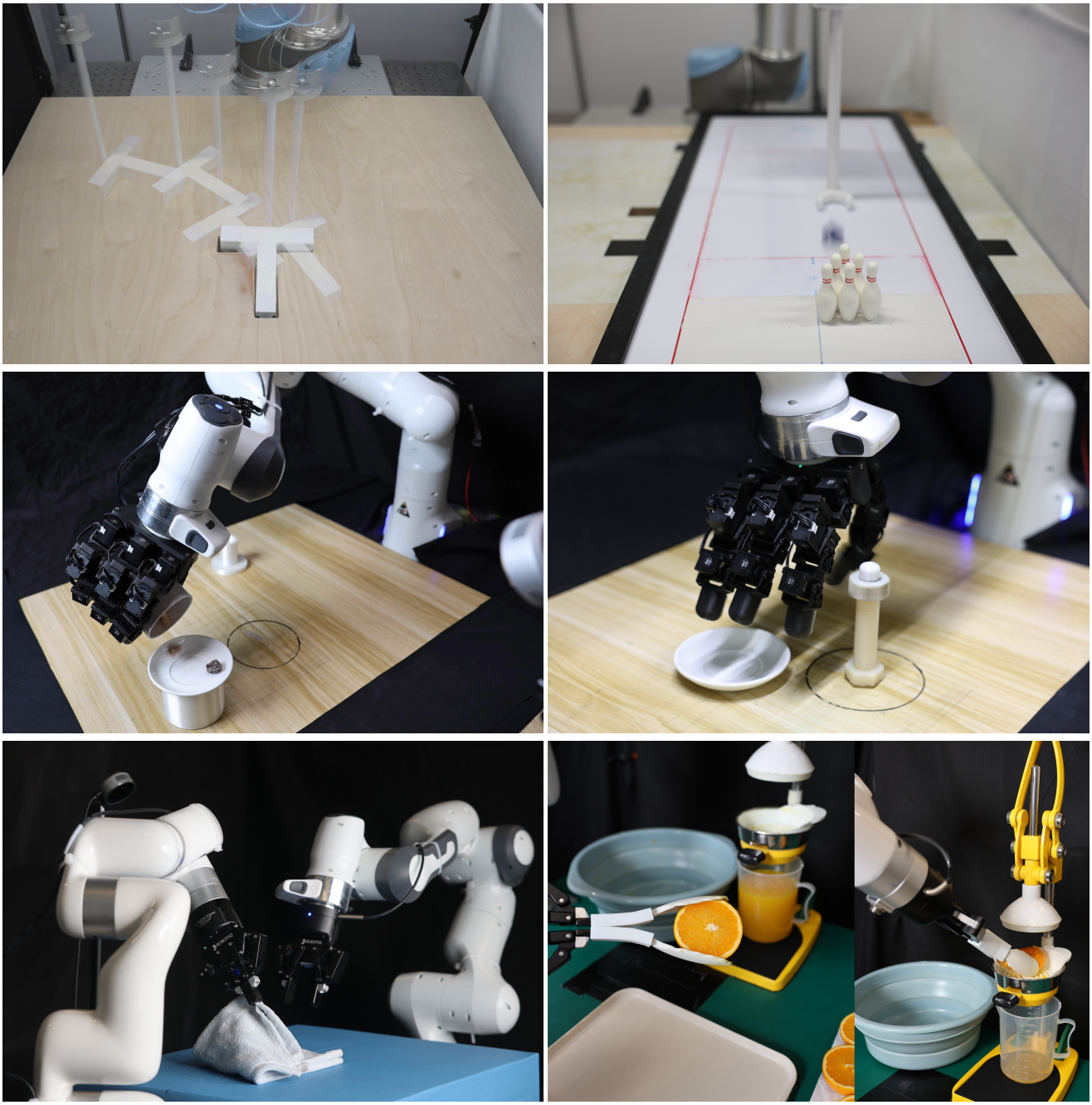

Figure 1. Teaser. Real-robot snapshots illustrating the diversity of our task suite. Panels are ordered *top-left* $\rightarrow$ *bottom-right*: (a) **Dynamic Push-T**; (b) **Agile Bowling**; (c) **Pouring**. (d) **Dynamic Unscrewing** with a dexterous hand; (e) Dual-arm **Folding**; (f) **Juicing**—*stage 1* and *stage 2* are shown side-by-side in the same panel; The grid highlights six of the seven tasks; juicing stage 1/2 are grouped into one panel for space.

supervision, then reinforce the skill on their own until they master it and eventually transfer it across terrains. Analogously, a generalizable robotic learning system should combine skilled human priors with self-improvement to reach—and in some cases exceed—human capability in reliability, efficiency, and robustness.

In this paper, we introduce RL-100, a framework that employs a real-world RL post-training phase on top of an imitation-based diffusion policy, preserving its expressive strengths while explicitly optimizing deployment metrics – success rate, time-to-completion, and robustness – under mild human-guided exploration. In short, we **start from human** priors, **align with human**-grounded objectives, and **go beyond human** performance. RL-100 has three stages with distinct roles and costs: (i) Imitation-learning (IL) pretraining on teleoperated demonstrations provides a competent, low-variance base, much like the sponge layer of a cake, on which subsequent learning can be built. (ii) Iterative offline RL post-training (offline updates on a growing buffer of policy rollouts) delivers the bulk of the improvement in success rate and efficiency across iterations, analogous to adding the cream layer. (iii) Online and on-policy RL post-training supplies the last-mile reliability, targeting rare failure modes that remain after iterative offline learning, which are the cherries on top. But it is resource-intensive on real hardware (parameter tuning,

resets, approvals). We therefore allocate most of the learning budget to iterative offline updates and use a small, targeted online budget to push performance from a high success rate (e.g., 95%) to near-perfect (e.g., 99%+).

Moreover, RL-100 is representation-agnostic: it operates in a vision-based setting and supports both 3D point clouds and 2D RGB images by simply swapping observation encoders, without modifying rest of the framework. While our real-robot experiments use 3D point clouds as the primary representation, ablations in simulation show the same performance trends with 2D inputs. In particular, we introduce a self-supervised visual encoder tailored for RL post-training, which furnishes stable, task-relevant features throughout policy exploration and updates.

During policy rollouts, a human operator gives sparse success signals when needed, and the controller follows conservative operating limits. We use a unified policy-gradient objective across both iterative offline and online phases to fine-tune the diffusion sampler's short-horizon denoising schedule (Song et al. 2021). This alignment yields stable updates across phases and strong fine-tuning sample efficiency. In addition, we interleave a lightweight distillation loss that compresses the K-step diffusion policy into a one-step consistency (Song et al. 2023) policy for deployment, reducing inference latency while maintaining or improving efficiency and robustness.

Moreover, our framework is task- and embodiment-agnostic. We evaluate RL-100 across simulated tasks and on a real-world suite of seven manipulation tasks as illustrated in Fig. 1 and summarized in Tab. 1: Dynamic Push-T, Agile Bowling, Pouring, Soft-towel Folding, Dynamic Unscrewing, and Orange Juicing. Orange Juicing comprises two subtasks, placing and removal, which are trained and evaluated separately but reported as one task family. The suite includes rigid-body dynamics, deformable objects, fluids, and precision assembly, and the framework is deployed across multiple embodiments. For these tasks and embodiments, we select a specific control mode per task: a single-step control mode is used when a fast closed-loop reaction is necessary; action-chunk-based control is preferred for coordination-heavy or high precision tasks where smoothing mitigates jitter and limits error compounding (Zhao et al. 2023). Both regimes share the same diffusion backbone; only the action heads differ.

Because we target deployment in homes and factories, we emphasize deployment-centric metrics: **reliability** (success rate), **efficiency** (time-to-completion), and **robustness** (sustained stability under long deployment time and perturbation). Real-world experiments show that RL-100 attains 100% reliability across all seven tasks (up to *250/250 consecutive* successful trials) and maintains long-horizon stability. In terms of efficiency, RL-100 approaches human teleoperation-level time-to-completion and, on several tasks, *matches or even surpasses* skilled operators. In summary, our main contributions are as follows:

(i) **Unified training framework.** We propose RL-100, a three-stage real-world RL training framework on top of teleoperation-trained generative diffusion policies. The pipeline chains IL pretraining, iterative offline RL and online RL, with most updates allocated to iterative offline and a small, targeted online budget for the last mile to deployment-grade performance.

(ii) **One objective, fast deployment.** A unified policy gradient loss fine-tunes the diffusion sampler's denoising schedule; a lightweight distillation compresses the multi-step diffusion policy to a one-step consistency policy.

(iii) **Generality across tasks, embodiments, and visual representations.** Our framework is task-, embodiment-, and visual-representation–agnostic. To the best of our knowledge, RL-100 is the first system to demonstrate vision-based RL post-training on real robots across diverse task modalities and embodiments.

(iv) **Deployment-centric results.** On real robots, RL-100 achieves 100% reliability across seven tasks, including runs of up to 250 consecutive successes, attains efficiency comparable to or exceeding human teleoperation on multiple tasks, and demonstrates long-horizon robustness with uninterrupted operation for up to two hours, offering a promising route to deployment in homes and factories

(v) **RL-specific network backbone.** Our policy backbone is tailored for diffusion-based visuomotor control and is agnostic to the execution regime, supporting both single-step and action-chunk control. We use a self-supervised visual encoder to produce stable, drift-resistant representations throughout RL fine-tuning.

Preliminaries

Reinforcement Learning

We formulate the robotic manipulation problem as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition dynamics, R is the reward function, and γ is the discount factor. The robot policy π chooses action a_t at state s_t to maximize discounted cumulative rewards $\Sigma_{t=0}^{\infty} \gamma^t R(s_t, a_t)$. The value function $V(s) = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, a_t)|_{s_0=s, a_t \sim \pi(\cdot|s_t)}$ is defined to measure robot performance starting from a given state s and the Q function $Q(s,a) = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, a_t)|_{s_0=s, a_0=a, a_t \sim \pi(\cdot|s_t)}$ starts from given s and a.

Offline-to-online RL. Our post-training follows an offline-to-online paradigm. Following Lei et al. (2024), we employ a proximal policy optimization (PPO)-style objective (Schulman et al. 2017) to unify both stages. The core learning objective incorporates importance sampling for proximal policy updates:

$$J_i(\pi) = \mathbb{E}_{s \sim \rho_\pi, a \sim \pi_i} \Big[\min \Big(r(\pi) A(s,a), \; \mathrm{clip}\big(r(\pi), 1-\epsilon, 1+\epsilon\big) A(s,a) \Big) \Big] \tag{1}$$

where ρ_π is the stationary state distribution induced by policy π, $r(\pi) = \frac{\pi(a|s)}{\pi_i(a|s)}$ is the importance ratio, and $A(s,a) = Q(s,a) - V(s)$ is the advantage function.

Table 1. Task suite, embodiments, and control modes. *Embodiments listed are typical examples; our framework is task- and embodiment-agnostic. Control modes are selected per task as either* Single-step *(one action per control tick) or* Action-chunk *(short c-step segments).*

Task	Control mode	Embodiments (examples)	Modality	Key challenges
Dynamic Push-T	Single-step	*UR5 + 3D-printed end-effector (single-arm)*	Rigid-body dynamics	Fast reaction to a moving goal pose and online perturbations
Agile Bowling	Single-step	*UR5 + 3D-printed end-effector (single-arm)*	Rigid-body dynamics	Release-timing control at high velocity; trajectory and release-pose accuracy
Pouring	Single-step	*Franka + LeapHand (single-arm)*	Fluids / granular	Spillage minimization; flow control; container alignment under motion
Dynamic Unscrewing	Action-chunk	*Franka + LeapHand (single-arm)*	Precision assembly	Time-varying alignment; torque/pose regulation; cross-thread avoidance
Soft-towel Folding	Action-chunk	*xArm + Franka + Robotiq (dual-arm)*	Deformable cloth	Large deformation; coordinated contacts; fold accuracy
Orange Juicing[‡]	Action-chunk	*xArm + Robotiq (single-arm)*	Deformable manipulation	Confined-space insertion/ejection; generalization to fruit variability

Control modes: Single-step (one action per control tick); Action-chunk (short c-step segments). [‡] Two subtasks: *Placing* (place fruit into the press zone) and *Removal* (remove the spent fruit); the spent fruit is *deformable*, fruit sizes vary substantially (requiring strong generalization), and operation occurs in a *confined* cavity with narrow clearances.

The key distinction between offline and online stages lies in advantage estimation:

- **Offline:** Implicit Q Learning (IQL)-style (Kostrikov et al. 2022) value functions: $A^{\text{off}}(s,a) = Q(s,a) - V(s)$.
- **Online:** Generalized Advantage Estimation (GAE) (Schulman et al. 2016): $A^{\text{on}}(s,a) = \text{GAE}(R_t, V)$ to balance variance and bias.

Diffusion models

We overload the subscript t from timestep in MDP to the step index in diffusion process in the following two subsections. Diffusion models (Ho et al. 2020a) learn to reverse a noising process that gradually corrupts clean data $x_0 \in \mathbb{R}^d$ into Gaussian noise to reconstruct the original clean data distribution. Given a schedule $\{\alpha_t\}_{t=1}^T$ with $\alpha_t = 1 - \beta_t$, a sample x_0 drawn from the clean distribution, the forward noising process follows a closed form:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 \; + \; \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{2a}$$

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{2b}$$

A denoiser ε_θ is trained to recognize the noise inside the noisy sample via

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{x_0,\, t,\, \varepsilon}\Big[\, \big\|\varepsilon - \varepsilon_\theta(x_t, t)\big\|_2^2 \Big]. \tag{3}$$

to recover the clean sample.

DDIM sampling with stochastic form

Denoising Diffusion Implicit Models (DDIM) (Song et al. 2021) provide a family of samplers that interpolate between deterministic and stochastic generation. Given a learned denoiser ε_θ, the predicted clean sample at time t is commonly written as

$$\hat{x}_0(x_t, t) = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\, \varepsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}, \tag{4}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ denotes the cumulative noise schedule.

We consider a (possibly) coarse time-subsequence for sampling $\tau_K > \tau_{K-1} > \cdots > \tau_1$, with $K \ll T$ (for example, $T = 50 \sim 1000$ and $K = 5 \sim 10$). To cover both the single-step ($t \to t-1$) and subsampled ($\tau_k \to \tau_{k-1}$) cases in a unified notation, denote a generic transition from time t to an earlier time m (with $m < t$) by $t \to m$. DDIM then defines a stochastic update with variance parameter $\sigma_{t\to m} \geq 0$ as

$$\mu_\theta(x_t, t \to m) = \sqrt{\bar{\alpha}_m}\, \hat{x}_0(x_t, t) + \sqrt{1 - \bar{\alpha}_m - \sigma_{t\to m}^2}\; \varepsilon_\theta(x_t, t), \tag{5a}$$

$$x_m = \mu_\theta(x_t, t \to m) + \sigma_{t\to m}\, \varepsilon_{t\to m}, \quad \varepsilon_{t\to m} \sim \mathcal{N}(0, \mathbf{I}). \tag{5b}$$

The radical in (5a) requires the constraint

$$1 - \bar{\alpha}_m - \sigma_{t\to m}^2 \geq 0, \qquad \text{hence} \qquad 0 \leq \sigma_{t\to m} \leq \sqrt{1-\bar{\alpha}_m}. \tag{6}$$

In particular, the deterministic DDIM update is recovered when $\sigma_{t\to m} = 0$ (the distribution degenerates to a Dirac at μ_θ). Conversely, positive values of $\sigma_{t\to m}$ inject stochasticity into the transition.

Policy perspective and log-likelihood. When $\sigma_{t\to m} > 0$ the transition from x_t to x_m can be viewed as a Gaussian

sub-policy

$$\pi_\theta(x_m \mid x_t, t \to m) = \mathcal{N}\big(\mu_\theta(x_t, t \to m),\, \sigma^2_{t\to m}\mathbf{I}\big), \tag{7a}$$

$$\log \pi_\theta(x_m \mid x_t, t \to m) = -\tfrac{1}{2\sigma^2_{t\to m}} \big\|x_m - \mu_\theta(x_t, t \to m)\big\|^2 + C, \tag{7b}$$

where C is a constant independent of the parameters θ. Note that (7b) is only valid for $\sigma_{t\to m} > 0$; when $\sigma_{t\to m} = 0$ the density becomes singular and the transition is best described as the deterministic mapping $x_m = \mu_\theta(x_t, t \to m)$ or equivalently a Dirac distribution.

A full DDIM sampling process recovers a clean sample x_0 by chaining the sub-policies $\{\pi_\theta(x_{\tau_{k-1}} \mid x_{\tau_k}, \tau_k \to \tau_{k-1})\}_{k=K}^{1}$, starting from $x_{\tau_K} \sim \mathcal{N}(0, \mathbf{I})$. In practice, one may set $\sigma_{t\to m} = 0$ for fully deterministic sampling, or choose a small positive $\sigma_{t\to m}$ (subject to (6)) to trade off between sample diversity and stability. If the log-likelihood in (7b) is later used as an objective or as part of a fine-tuning criterion, care must be taken in handling the $\sigma_{t\to m} \to 0$ limit (e.g., by restricting likelihood-based terms to steps with strictly positive variance).

Notation and scheduling conventions. Throughout the paper we distinguish MDP timesteps from diffusion (denoising) timesteps. Environmental timesteps are denoted by t, while diffusion indices follow a (possibly subsampled) schedule

$$\tau_K > \tau_{K-1} > \cdots > \tau_1,$$

and are written as superscripts (e.g. a^{τ_k}). A generic denoising transition from time t to an earlier time m is denoted by $t \to m$ (for the subsampled schedule this will typically be written $\tau_k \to \tau_{k-1}$). Variance parameters are indexed consistently as $\sigma_{\tau_k \to \tau_{k-1}}$ (abbreviated as σ_{τ_k} when unambiguous). We always enforce the constraint

$$0 \le \sigma_{\tau_k} \le \sqrt{1 - \bar{\alpha}_{\tau_{k-1}}}, \tag{8}$$

so that all square roots appearing in the DDIM updates are real. When $\sigma_{\tau_k} = 0$ the corresponding transition degenerates to a deterministic mapping (Dirac), and Gaussian log-densities are not defined; therefore any likelihood-based objective (e.g., policy-gradient using $\log \pi$) must only use steps with strictly positive variance.

Consistency Models

Consistency models (Song et al. 2023) learn a single-step mapping from noisy inputs at arbitrary noise levels to clean data. Denote the consistency model by $C_\theta(x^\tau, \tau)$ where the superscript indicates the diffusion index (per the notation above). Given a frozen diffusion teacher Ψ_φ (for instance a K-step DDIM teacher that follows the same subsampled schedule $\{\tau_k\}$), consistency distillation minimizes the squared regression objective

$$\mathcal{L}_{\mathrm{CD}}(\theta) = \mathbb{E}_{x_0,\tau,\varepsilon}\Big[\big\|C_\theta(x^\tau, \tau) - \mathrm{sg}\big[\Psi_\varphi(x^\tau, \tau \to 0)\big]\big\|_2^2\Big], \tag{9}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient and $\Psi_\varphi(x^\tau, \tau \to 0)$ is the teacher's output after running the teacher's denoising chain from x^τ down to (approximate) x^0. The teacher must be run using the same subsampled schedule $\{\tau_k\}$ that the student will emulate or distill from.

At inference time a consistency model requires only a single evaluation:

$$x^0 \approx C_\theta(x^{\tau_K}, \tau_K), \qquad x^{\tau_K} \sim \mathcal{N}(0, \mathbf{I}). \tag{10}$$

Diffusion policy and RL fine-tuning

Diffusion Policy (Chi et al. 2023) performs diffusion over robot actions conditioning on observations. Given an observation o, we roll out along the K-step subsampled schedule $\{\tau_K > \cdots > \tau_1\}$:

$$a^{\tau_{k-1}} \;=\; f_\theta(a^{\tau_k}, \tau_k \,|\, o)\,, \qquad k = K, \ldots, 2, \tag{11}$$

where f_θ follows the DDIM schedule (5) conditioned on o. Then we can retrieve a clean action $a_t := a^{\tau_0}$ from the so-called diffusion policy.

RL formulation. Now we embed the K-step diffusion process as a sub-MDP into a single step in robotic manipulation MDP. Each denoising step can be viewed as sampling a cleaner noisy action $a^{\tau_{k-1}}$ from the Gaussian sub-policy in (7a). We therefore model the process as a K-step sub-MDP with:

- **Initial state:** $s^K = (a^{\tau_K}, \tau_K, o)$ with $a^{\tau_K} \sim \mathcal{N}(0, \mathbf{I})$.
- **State:** $s^k = (a^{\tau_k}, \tau_k, o)$, $k = K, \ldots, 1$.
- **Action:** $u^k = a^{\tau_{k-1}}$ drawn from the denoising sub-policy $\pi_\theta(u^k \mid s^k) = \mathcal{N}(\mu_\theta(a^{\tau_k}, \tau_k, o),\, \sigma^2_{\tau_k}\mathbf{I})$.
- **Transition:** $s^{k-1} = (u^k, \tau_{k-1}, o)$.
- **Reward:** this sub-MDP only receives terminal reward $R(a^{\tau_0})$ from the upper environment MDP.

The log-likelihood defined in (7b) then computes the density of sub-policies, enabling end-to-end optimization of *task* rewards with respect to the denoising *sub-policies* via policy-gradient updates (e.g., PPO (Schulman et al. 2017)).

Methods

We present RL-100, a unified framework for robot learning that combines IL with RL. As illustrated in Fig. 2, our approach consists of three stages: (1) imitation learning from human demonstrations, (2) iterative offline RL with progressive data expansion, and (3) online fine-tuning. The key innovation lies in unifying offline and online RL through a shared PPO-style objective applied across diffusion denoising steps.

Imitation Learning

We initialize the policy by behavior cloning on human-teleoperated trajectories. Our approach uses conditional diffusion to learn robust visuomotor policies from demonstrations. Each episode provides synchronized tuples

$$\big\{(o_t,\, q_t,\, a_t)\big\}_{t=1}^{T_e}, \tag{12}$$

where o_t are visual observations (RGB images or 3D point clouds), q_t denotes robot proprioception (joint positions/velocities, gripper state), and a_t is either a single-step action or an action chunk.

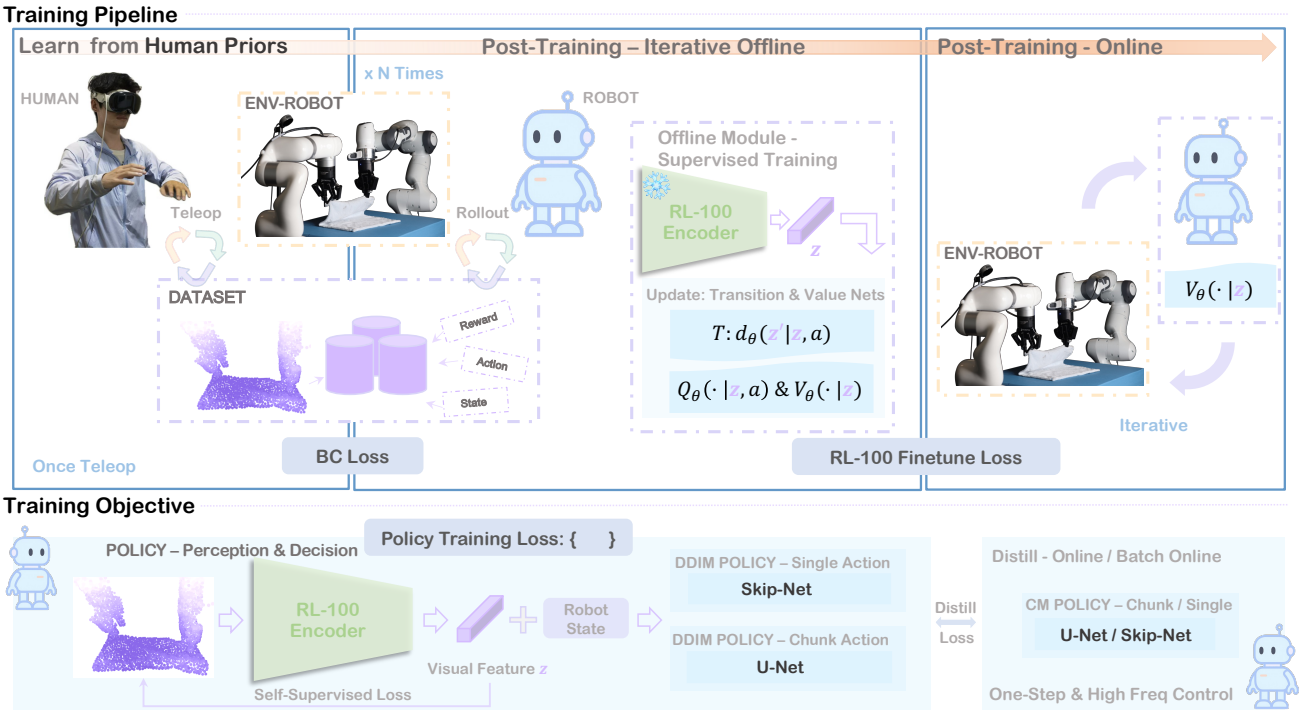

Figure 2. Overview of RL-100. We first learn from human demonstrations through imitation learning with diffusion policies, then apply iterative offline RL with data expansion, followed by online fine-tuning for final performance optimization.

Conditioning and prediction horizon. We fuse recent observations into a conditioning vector

$$c_t = [\phi(o_i, q_i)]_{i=t-n_o+1}^{t}, \tag{13}$$

where the perception encoder $\phi(\cdot)$ processes the most recent n_o frames (typically $n_o = 2$) and $[\cdot]$ is the operator concatenating multiple vectors. The clean diffusion target $a_t^{\tau_0}$ at timestep t is set to a single action $a_t^{\tau_0} = u_t \in \mathbb{R}^{d_a}$ or an action chunk $a_t^{\tau_0} = [u_t, \ldots, u_{t+n_c-1}] \in \mathbb{R}^{n_c d_a}$ where n_c is the chunk size (typically 8–16). Actions are normalized per dimension; we predict delta end-effector pose when applicable.

Diffusion parameterization. Following conditional diffusion over actions, we corrupt $a_t^{\tau_0}$ to $a_t^{\tau_K}$ via the forward process (Eq. (2)). The denoiser $\varepsilon_\theta(a^\tau, \tau, c_t)$ is trained with the noise-prediction objective:

$$\mathcal{L}_{\text{IL}}(\theta) = \mathbb{E}_{(a^{\tau_0}, c_t)\sim\mathcal{D},\, \tau,\, \varepsilon}\big[\, \|\varepsilon - \varepsilon_\theta(a^\tau, \tau, c_t)\|_2^2 \,\big], \tag{14}$$

where $\mathcal{D}$ is the demonstration dataset, $\tau \in \{\tau_K > \cdots > \tau_1\}$ indexes a K-step schedule, and $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$. The policy backbone is shared across control modes; only the output head differs: $\mathbb{R}^{d_a}$ (single-step) or $\mathbb{R}^{n_c d_a}$ (chunked).

Vision and proprioception encoders. For RGB input, we use pretrained ResNet/ViT backbones; for point clouds input, we adapt the 3D encoder of DP3 (Ze et al. 2024) with reconstruction regularization for stability during RL fine-tuning. Visual embeddings are projected and concatenated with proprioceptive features to form c_t. All encoders are trained end-to-end with Eq. (14).

When applicable, we add reconstruction (Recon) and variational information bottleneck (VIB) regularization:

$$\mathcal{L}_{\text{recon}} = \beta_{\text{recon}}\big(d_{\text{Chamfer}}(\hat{o}, o) + \|\hat{q} - q\|_2^2\big), \tag{15}$$

$$\mathcal{L}_{\text{KL}} = \beta_{\text{KL}}\, \text{KL}\big(\phi(z|o,s)\,\|\,\mathcal{N}(0, I)\big), \tag{16}$$

where o and q denote the observed point cloud and proprioceptive vector; $\hat{o}$ and $\hat{q}$ are the reconstructed observations given encoded embedding $\phi(o, q)$; d_{Chamfer} is Chamfer distance between two set of point clouds. The complete imitation learning objective becomes:

$$\mathcal{L}_{\text{total}}^{\text{IL}} = \mathcal{L}_{\text{IL}} + \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}}. \tag{17}$$

During RL fine-tuning, we reduce β_{recon} and β_{KL} by a factor of 10 to allow policy improvement while maintaining representation stability.

Inference and control. At deployment, K-step DDIM sampling (Eq. (5)) generates actions:

$$\hat{a}_t^{\tau_0} \leftarrow \text{DDIM}_K\big(\varepsilon_\theta(\cdot, \cdot, c_t)\big).$$

Single-step control executes $u_t \leftarrow \hat{a}_t^{\tau_0}$ immediately; chunked control executes $[u_t, \ldots, u_{t+n_c-1}] \leftarrow \hat{a}_t^{\tau_0}$ in the following n_c timesteps. Single-step control excels in reactive tasks (e.g., dynamic bowling), while action chunking reduces jitter in precision tasks (e.g., assembly). Both modes share the same architecture, enabling task-adaptive deployment.

Unified Offline and Online RL Fine-tuning

Handling single action vs. action chunk. Our framework supports both single-step and chunked action execution, which affects value computation and credit assignment:

- **Single action:** Standard MDP formulation with per-step rewards R_t and discount γ
- **Action chunk:** Each chunk of n_c actions is treated as a single decision. The chunk receives cumulative reward $R_{\text{chunk}} = R_{t:t+n_c-1} = \sum_{j=0}^{n_c-1} \gamma^j R_{t+j}$, and the equivalent discount factor between chunks is γ^{n_c}

For clarity, we present the single-action case below; the chunked case works similarly by replacing per-step quantities with their chunk equivalents.

Two-level MDP structure. Our approach operates on two temporal scales:

1. **Environment MDP:** Standard robot control with state s_t, action a_t, reward R_t
2. **Denoising MDP:** K-step diffusion process generating each $a_t^{\tau_0}$ through iterative refinement

The denoising MDP is embedded within each environment timestep, creating a hierarchical structure where K denoising steps produce one environment action.

Unified PPO objective over denoising steps. Given the two-level MDP structure, we optimize the PPO objective w.r.t. the K-step diffusion process at each timestep t in iteration i via the summation across all denoising steps k:

$$J_i(\pi) = \mathbb{E}_{s_t\sim\rho_\pi,\, a_t\sim\pi_i}\Bigg[\sum_{k=1}^{K}\min\big(r_k(\pi)\,A_t,\; \mathrm{clip}\big(r_k(\pi),\,1-\epsilon,\,1+\epsilon\big)\,A_t\big)\Bigg], \tag{18}$$

and the loss function is

$$\mathcal{L}_{\mathrm{RL}}^{\mathrm{off}} = -\,J_i(\pi), \tag{19}$$

where $r_k(\pi)$ is the per-denoising-step importance ratio and A_t is the task advantage tied to the environment timestep t. Here, π_i denotes the behavior policy at PPO iteration i, and ρ_π is the (discounted) state distribution under current policy π. The key insight is to share the same environment-level advantage A across all K denoising steps, providing dense learning signals throughout the denoising process while maintaining consistency with the environment reward structure.

Offline RL

Setting. Given an offline dataset $\mathcal{D}$, we initialize the behavior policy with a diffusion policy acquired from IL: $\pi_0 \coloneqq \pi^{\mathrm{IL}}$. No new data are collected in this pure offline stage.

Policy improvement on $\mathcal{D}$. At offline iteration i, we optimize the PPO-style surrogate (Eq. (18)) applied across K denoising steps, using the offline-policy ratio

$$r_k^{\mathrm{off}}(\pi) = \frac{\pi(a^{\tau_{k-1}} \mid s^k)}{\pi_i(a^{\tau_{k-1}} \mid s^k)},$$

with standard clipping. Offline advantages are computed as

$$A_t^{\mathrm{off}} = Q_\psi(s_t, a_t) - V_\psi(s_t),$$

where the critics (Q_ψ, V_ψ) are pre-trained on $\mathcal{D}$ following IQL (Kostrikov et al. 2022). This yields a candidate policy π from π_i by several epochs of gradient updates on $\mathcal{D}$.

OPE gate and iteration advancement. We use AM-Q (Lei et al. 2024) for offline policy evaluation (OPE) without further interaction with the environment to compare the candidate with current behavior policy:

$$\widehat{J}^{\text{AM-Q}}(\pi) = \mathbb{E}_{(s,a)\sim(\hat{T},\pi)}\Big[\sum_{t=0}^{H-1} Q_\psi(s_t, a_t)\Big],$$

where $\hat{T}$ is a learned transition model. We accept the candidate and advance the behavior-policy iteration only if

$$\widehat{J}^{\text{AM-Q}}(\pi) \;-\; \widehat{J}^{\text{AM-Q}}(\pi_i) \;\geq\; \delta, \tag{20}$$

by setting the updated policy as behavior policy: $\pi_{i+1} \coloneqq \pi$. Otherwise, we reject and keep the behavior policy unchanged ($\pi_{i+1} \coloneqq \pi_i$). In practice, we set $\delta = 0.05 \cdot |\widehat{J}^{\text{AM-Q}}(\pi_i)|$ for adaptive thresholding. This OPE-gated rule yields conservative, monotonic behavior-policy improvement on $\mathcal{D}$.

Shared and frozen encoders. To ensure stable representation learning and efficient computation, all components in our offline RL pipeline share the same fixed visual encoder ϕ pre-trained during imitation learning. During offline RL, we keep ϕ^{IL} fixed and only update the task-specific heads of each module.

Online RL The online stage uses on-policy components. We use an on-policy ratio for each diffusion step

$$r_k^{\mathrm{on}}(\pi) = \frac{\pi(a^{\tau_{k-1}} \mid s_k)}{\pi_i(a^{\tau_{k-1}} \mid s_k)},$$

and compute advantages using GAE:

$$A_t^{\mathrm{on}} = \mathrm{GAE}(\lambda, \gamma; r_t, V_\psi),$$

sharing the same A_t^{on} across all K denoising steps that produce the environment action at time t. We minimize the total loss:

$$\mathcal{L}_{\mathrm{RL}}^{\mathrm{on}} = -\,J_i(\pi) \;+\; \lambda_V\,\mathbb{E}\Big[(V_\psi(s_t) - \hat{V}_t)^2\Big], \tag{21}$$

where $\hat{V}_t = \sum_{l=0}^{\infty}\gamma^l r_{t+l}$ is the discounted return and λ_V weights the value function loss.

Distillation to One-step Consistency Policy

High-frequency control is crucial for robotics applications. While our diffusion policy achieves strong performance, the K-step denoising process introduces latency that can limit real-time deployment. To address this, we jointly train a consistency model c_w that learns to directly map noise to actions in a single step, distilling knowledge from the multi-step diffusion teacher π_θ.

During both offline and online RL training, we augment the policy optimization objective with the consistency distillation loss from Eq. (9):

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{RL}} + \lambda_{\mathrm{CD}} \cdot \mathcal{L}_{\mathrm{CD}}, \tag{22}$$

where $\mathcal{L}_{\mathrm{RL}}$ is either the offline objective (Eq. (19) with IQL-based advantages) or the online objective (Eq. (21) with GAE). The consistency loss $\mathcal{L}_{\mathrm{CD}}$ follows Eq. (9), with the teacher being our diffusion policy π_θ that performs K-step denoising conditioned on observation. The stop-gradient operator ensures the teacher policy continues to improve through RL objectives while simultaneously serving as a distillation target.

High-frequency control is essential for practical robot deployment in industrial settings. First, faster control loops directly translate to improved task completion efficiency—a robot operating at 20 Hz can execute the same trajectory in half the time compared to 10 Hz

operation, significantly increasing throughput in factory automation, where cycle time directly impacts productivity. Second, many manipulation tasks inherently require high-frequency feedback for reliable execution. Dynamic tasks such as catching moving objects, maintaining contact during sliding motions, or recovering from unexpected disturbances demand sub-50 ms response times that multi-step diffusion cannot provide. Furthermore, tasks involving compliance control, force feedback, or human-robot collaboration often fail catastrophically when control frequency drops below critical thresholds, as the system cannot react quickly enough to prevent excessive forces or maintain stable contact.

During inference, the consistency model generates actions in a single forward pass: $a^{\tau_0} = c_w(a^{\tau_K}, \tau_K | o)$, achieving $K\times$ speedup (e.g., from 100ms to 10ms latency) while preserving the performance of the diffusion policy. This order-of-magnitude improvement enables deployment in real-world manufacturing scenarios where robots must maintain consistent cycle times, respond to conveyor belt speeds, and safely operate alongside human workers—requirements that are infeasible with standard multi-step diffusion policies.

Overall training framework

While each component above can be used independently, we propose an iterative procedure that combines them for progressive improvement. Instead of applying offline RL once on fixed demonstrations, we alternate between training an IL policy on the current dataset, improving it via offline RL with conservative updates, collecting new data with the improved policy, and re-training IL on the expanded dataset, which is summarized in Algo. 1. This creates a virtuous cycle where better policies generate better data, which in turn enables learning even better policies.

Algorithm 1 RL-100 training pipeline

1: **Input:** Demonstrations $\mathcal{D}_0$, iterations M
2: **Initialize:** $\pi_0^{\text{IL}} \leftarrow \text{ImitationLearning}(\mathcal{D}_0)$
3: **for** iteration $m = 0$ to $M-1$ **do**
4: // Offline RL improvement
5: Train critics: $(Q_{\psi_m}, V_{\psi_m}) \leftarrow \text{IQL}(\mathcal{D}_m)$
6: Train transition: $T_{\theta_m}(s'|s,a)$
7: Optimize:
8: $\pi_m^{\text{ddim}}, \pi_m^{\text{cm}} \leftarrow \text{OfflineRL}(\pi_m^{\text{IL}}, Q_{\psi_m}, V_{\psi_m}, T_{\theta_m})$
9: // Data expansion
10: Deploy: $\mathcal{D}_{\text{new}} \leftarrow \text{Rollout}(\pi_m^{\text{ddim}} \text{ or } \pi_m^{\text{cm}})$
11: Merge: $\mathcal{D}_{m+1} \leftarrow \mathcal{D}_m \cup \mathcal{D}_{\text{new}}$
12: // IL re-training on expanded data
13: $\pi_{m+1}^{\text{IL}} \leftarrow \text{ImitationLearning}(\mathcal{D}_{m+1})$
14: **end for**
15: // Final online fine-tuning
16: $\pi_{\text{ddim}}^{\text{final}}, \pi_{\text{cm}}^{\text{final}} \leftarrow \text{OnlineRL}(\pi_{M-1}, V_{\psi_{M-1}})$
17: **Output:** $\pi_{\text{ddim}}^{\text{final}}, \pi_{\text{cm}}^{\text{final}}$

Why IL re-training matters. Re-training with IL on the expanded dataset (Algo. 1 line 13) is crucial for several reasons:

- **Distribution shift:** IL naturally adapts to the evolving data distribution as higher-quality trajectories are added
- **Stability:** Supervised learning is more stable than RL on mixed-quality data
- **Multimodality:** IL preserves the diffusion policy's ability to model multiple solution modes
- **Distillation:** IL effectively distills both human demonstrations and RL improvements into a unified policy

Final online fine-tuning. After the iterative offline procedure converges, we apply online RL for final performance optimization. This stage benefits from: (1) a strong initialization from iterative offline training, (2) pre-trained value functions that accelerate learning, and (3) a diverse dataset for replay and regularization.

Variance clipping for stable exploration. To ensure stable learning during RL fine-tuning, we introduce variance clipping in the stochastic DDIM sampling process. Specifically, we constrain the standard deviation at each denoising step:

$$\tilde{\sigma}_k = \text{clip}(\sigma_k, \sigma_{\text{min}}, \sigma_{\text{max}}), \tag{23}$$

where σ_k is the original DDIM variance parameter from Eq. (5b), and $[\sigma_{\text{min}}, \sigma_{\text{max}}]$ defines the permissible range. This modification effectively bounds the stochasticity of the behavior policy $\pi_\theta(a^{\tau_{k-1}}|a^{\tau_k}, \tau_k) = \mathcal{N}(\mu_\theta(a^{\tau_k}, \tau_k), \tilde{\sigma}_k^2 \mathbf{I})$, preventing both:

- **Excessive exploration** when σ_t is too large, which can lead to out-of-distribution actions that destabilize training or cause safety violations in physical systems
- **Premature convergence** when σ_t approaches zero, which eliminates exploration and prevents the policy from discovering better modes

In practice, we set $\sigma_{\text{min}} = 0.01$ to maintain minimal exploration even in late denoising steps, and $\sigma_{\text{max}} = 0.8$ to prevent destructive exploration in early steps. This bounded variance ensures that the importance ratio $r_k(\pi) = \frac{\pi(a^{\tau_{k-1}}|s^k)}{\pi_i(a^{\tau_{k-1}}|s^k)}$ remain well-behaved during PPO updates, as extreme variance differences between the current and behavior policies are avoided. We will empirically demonstrate that this simple modification is crucial for achieving stable fine-tuning performance.

Related Work

Reinforcement Learning with Generative Diffusion Models

The integration of generative diffusion models with RL represents a paradigm shift in policy representation and optimization. Building upon foundational work in diffusion models (Ho et al. 2020b; Song et al. 2020) and flow matching (Lipman et al. 2023), recent advances have demonstrated the power of these generative frameworks in capturing complex, multimodal action distributions inherent in RL problems. Diffusion Q-Learning (DQL) (Wang et al. 2023b) pioneered this integration by replacing traditional Gaussian policies with conditional diffusion models in offline reinforcement learning, addressing fundamental limitations of parametric policies in modeling multimodal behaviors. This approach has evolved through multiple directions: weighted regression

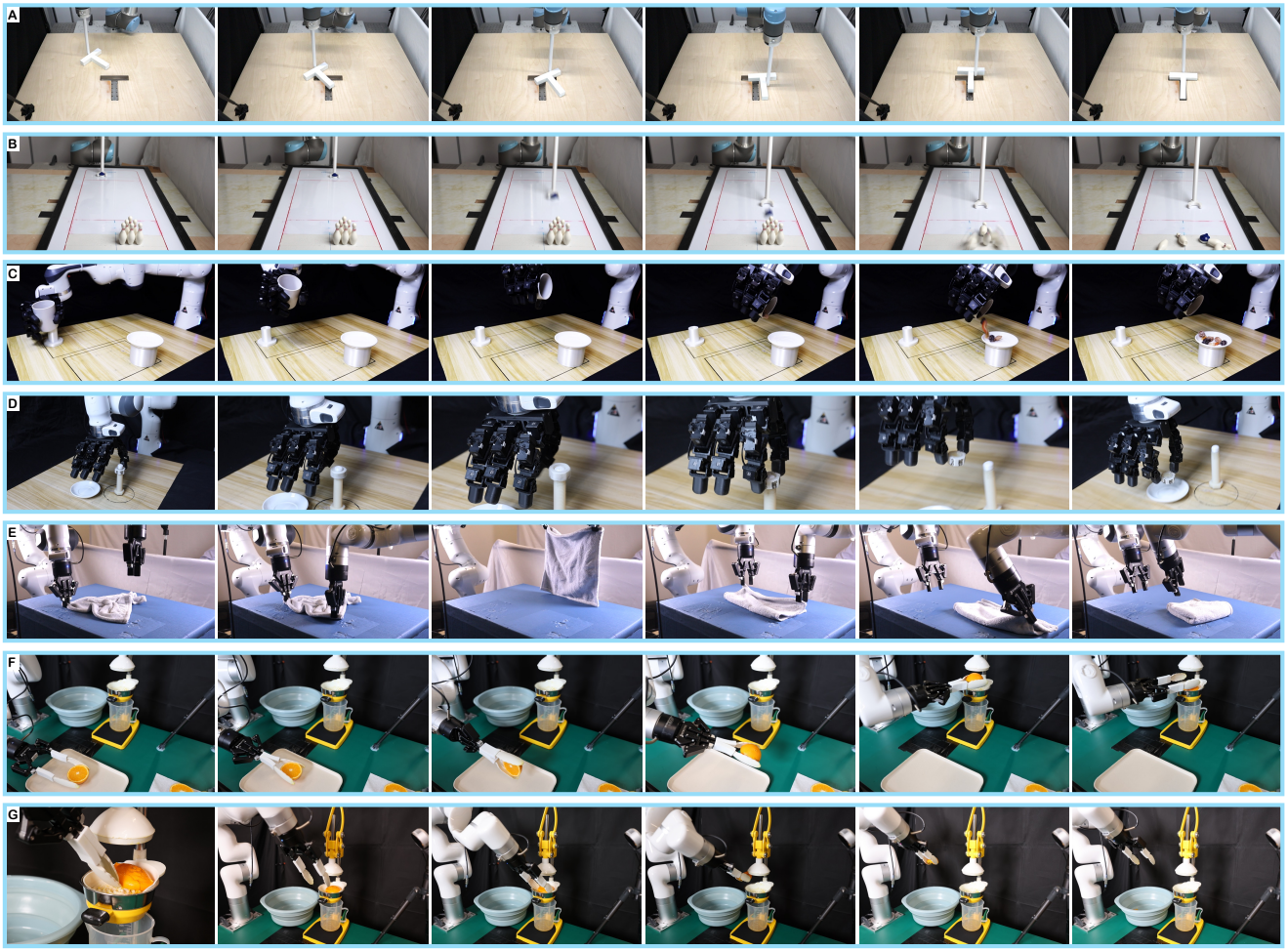

Figure 3. Illustrations from rollouts on seven real-world tasks. Each row shows one task, and columns are time-ordered frames (left→right) subsampled from a single trajectory. From top to bottom: (a) Dynamic Push-T, (b) Agile bowling, (c) Pouring, (d) Dynamic Unscrewing, (e) Soft-towel Folding, (f) Orange Juicing – Placing, (f) Orange Juicing – Removal. The suite spans fluids or particles, tool-using, deformable objects manipulation, dynamic non-prehensile manipulation, and precise insertion, highlighting the diversity and dynamics of our benchmark.

methods (Kang et al. 2023; Lu et al. 2023; Ding et al. 2024a) train diffusion policies through importance-weighted objectives to maximize learned Q-functions; reparameterization gradient approaches (Psenka et al. 2024; He et al. 2023; Ding et al. 2024b) utilize gradient-based optimization despite temporal backpropagation challenges; and sampling-based methods (Chen et al. 2023; Hansen-Estruch et al. 2024) provide effective but computationally expensive solutions. More recently, consistency-based extensions(Li et al. 2024) have further generalized diffusion and consistency policies to visual RL.

Recent works have also explored using RL to directly optimize diffusion models beyond traditional likelihood-based training. Black et al. (2023) demonstrate fine-tuning diffusion models with RL to maximize arbitrary reward functions, while Fan et al. (2024) apply similar techniques specifically to text-to-image generation using human feedback. Ren et al. (2024) introduce policy gradient methods tailored for diffusion-based policies, enabling effective online optimization while maintaining the expressiveness benefits of diffusion models.

To address computational bottlenecks inherent in diffusion models, Park et al. (2025) present Flow Q-Learning (FQL), which leverages flow-matching policies to model complex action distributions while avoiding recursive backpropagation through denoising processes. By training an independent single-step policy that matches the flow model's output, FQL achieves computational efficiency without sacrificing expressiveness, demonstrating state-of-the-art performance on offline RL benchmarks with significant computational advantages, particularly in offline-to-online fine-tuning settings. Similarly, One-Step Flow Q-Learning (OFQL) (Nguyen and Yoo 2025) extends this framework for even faster single-step action generation.

Generative Diffusion Models in Robotics

The application of diffusion models to robotics has yielded transformative advances in visuomotor control, trajectory planning, and real-world deployment. Diffusion Policy (Chi et al. 2023) demonstrated breakthrough performance in visuomotor control by modeling action distributions as conditional diffusion processes, excelling at handling multimodal behaviors through visual conditioning and receding horizon control. This approach has inspired extensions including DiffClone (Sabatelli et al. 2024) and Diffusion Model-Augmented Behavioral Cloning (Wang et al. 2023a), which further improve upon traditional behavioral cloning by leveraging the expressiveness of

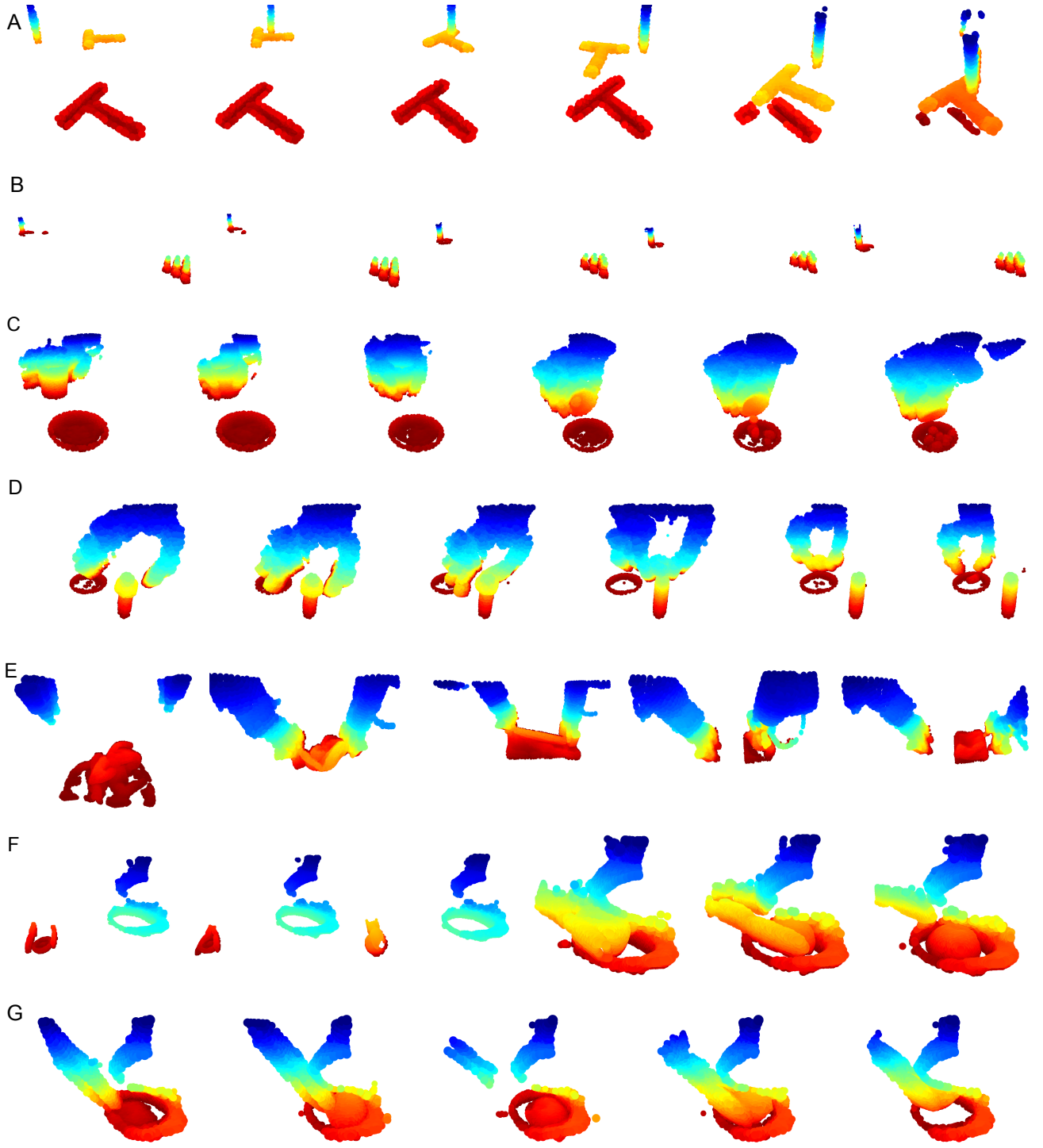

Figure 4. Point clouds trajectories for seven tasks (ordered top-to-bottom): (A) Push-T, (B) Bowling, (C) Pouring, (D) Unscrewing, (E) Folding, (F) Orange Juicing – Placing, (G) Orange Juicing – Removal.

diffusion models. The integration with pretrained visual representations, including R3M (Nair et al. 2022), VC-1 (Majumdar et al. 2023), and MoCo (He et al. 2020), has proven particularly effective for generalization across visual variations. Recent work on 3D Diffusion Policy (Ze et al. 2024) extends these capabilities to point cloud observations, while FlowPolicy (Chen et al. 2024) introduces consistency flow matching for robust 3D manipulation tasks. Lu et al. (2025) propose a triply-hierarchical diffusion policy that decomposes complex visuomotor tasks into multiple levels of abstraction, improving both learning efficiency and generalization.

Beyond direct policy learning, diffusion models have revolutionized trajectory planning in robotics. Janner et al. (2022) and Ajay et al. (2023) reformulate planning as conditional generation, producing complete state-action trajectories conditioned on rewards and constraints, excelling in long-horizon tasks requiring complex coordination. Shang et al. (2024) introduces an alternative paradigm by generating policy parameters rather than trajectories

in latent spaces, offering computational advantages. Real-world deployment challenges have been addressed through methods like One-Step Diffusion Policy (Wang et al. 2024), which uses distillation to achieve real-time performance suitable for robotic control. Successful applications now span deformable object manipulation with PinchBot (Liu et al. 2024), multi-task learning, and navigation scenarios. The availability of large-scale datasets like BridgeData V2 (Walke et al. 2023) and foundation models such as π_0 (Black et al. 2024b) enables broader generalization across robotic platforms and tasks, accelerating the transition from laboratory demonstrations to practical robotic systems.

Offline-to-Online Reinforcement Learning

The transition from offline pretraining to online fine-tuning presents unique challenges in managing distribution shift and preventing catastrophic forgetting while enabling continuous improvement. Conservative Q-Learning (CQL) (Kumar et al. 2020) established the foundational framework for safe offline RL through pessimistic value estimation, preventing overestimation for out-of-distribution actions. Advantage-Weighted Regression (AWR) (Peng et al. 2019) provides a scalable framework that has influenced numerous subsequent works, though its restriction to Gaussian policies limits expressiveness for complex behavioral patterns. Calibrated Q-Learning (Cal-QL) (Nakamoto et al. 2023) addresses initialization challenges by ensuring conservative Q-values are appropriately scaled for effective online fine-tuning, providing crucial insights for successful offline-to-online transfer. Uni-O4 (Lei et al. 2024) directly applies the PPO (Schulman et al. 2017) objective to unify offline and online learning, eliminating the need for extra regularization.

Recent advances focus on efficiently leveraging both offline and online data through hybrid strategies. RLPD (Ball et al. 2023) achieves sample-efficient online learning by mixing offline and online experiences, demonstrating that careful data mixture strategies can accelerate learning significantly. Nair et al. (2020) established fundamental frameworks for combining offline pretraining with online fine-tuning, showing that hybrid approaches achieve superior sample efficiency compared to pure online or offline methods. A paradigm shift in offline-to-online adaptation is represented by Wagenmaker et al. (2025), which introduces DSRL (Diffusion Steering with Reinforcement Learning) that operates RL entirely in the latent noise space of pretrained diffusion policies rather than modifying base policy weights.

Real-world RL

Real-world RL trains directly on real robot dynamics, optimizing deployment metrics (reliability, speed, safety) and yielding robust performance that continually adapts to disturbances—without sim-to-real gaps. Critical requirements include *sample efficiency*, stability under high-dimensional perception, safe continuous operation, and automated reward/reset mechanisms. While early work demonstrated end-to-end visuomotor learning (Levine et al. 2016) and large-scale grasping with off-policy methods (Kalashnikov et al. 2018), subsequent advances established key algorithmic foundations: off-policy actor-critic methods (SAC, TD3) for data efficiency (Haarnoja et al. 2018; Fujimoto et al. 2018), model-based approaches for sample acceleration (Chua et al. 2018; Janner et al. 2019), reset-free learning for autonomous operation (Eysenbach et al. 2018; Gupta et al. 2021), and learned reward specifications from visual classifiers or human feedback (Singh et al. 2019; Christiano et al. 2017). Despite these advances, most systems required extensive engineering, task-specific tuning, or long training times to achieve reliable performance.

These scattered advances converge in SERL (Luo et al. 2024b), a comprehensive framework that integrates high update-to-data ratio off-policy learning, automated resets, and visual reward specification to achieve several manipulation tasks. However, SERL relies solely on demonstrations and struggles with tasks requiring precision or recovery from failures. HIL-SERL (Luo et al. 2024c) addresses these limitations by incorporating real-time human corrections during training, enabling the policy to learn from mistakes and achieve perfect success rates across diverse tasks, including dual-arm coordination and dynamic manipulation.

While SERL and HIL-SERL report impressive on-robot learning efficiency and reliability on well-scoped tabletop skills, their evaluations typically employ action–space shaping (e.g., limiting wrist rotations and encouraging near-planar end-effector motion) and focus on short-horizon regimes with relatively low-dimensional control (Luo et al. 2024b,c). Such constraints are pragmatic for safety and sample efficiency, but they reduce policy expressivity and can cap performance on orientation-critical, contact-rich, or compositionally complex tasks. In everyday home and factory scenarios, many skills inherently require full SE(3) control and substantial reorientation, including: deformable manipulation with twist and regrasp (e.g., towel folding), insertion/ejection in confined cavities with large tilt changes (e.g., orange juicing), fluid and granular control that hinges on container tilt (e.g., controlled pouring), dynamic release and trajectory shaping (e.g., agile bowling), cable routing or cloth placement with out-of-plane rotations, and bimanual reorientation.* In contrast, our system retains full 6-DoF control without hard rotation constraints and targets these under-explored regimes by (i) using a diffusion/consistency visuomotor policy to capture diverse human strategies, (ii) unifying offline-to-online improvement with an OPE-gated PPO-style objective for nearly-monotonic improvement, and (iii) enabling high-frequency control via one-step consistency distillation, across dual-arm, deformable, and dynamic tasks with larger cross-object generalization.

Real-world Experiments

In this section, we detail the experimental setup and report results on reliability (success rate), efficiency (time-to-completion), and robustness (generalization across objects and initial conditions). We compare policies trained with RL-100 to human teleoperators and strong baselines. Across

*We do not claim these systems cannot be extended to such settings; rather, the *reported* experiments emphasize rapid learning on well-scoped tasks, leaving broader generalization, long-horizon composition, and orientation-sensitive control less explored.

tasks, RL-100 achieves **higher precision and greater efficiency than human teleoperation**. We then present ablations to analyze the contribution of each component of RL-100. Collectively, these results demonstrate the **practical viability of RL-100 for real-world deployment**.

Overview

We evaluate RL-100 on seven real-world manipulation tasks that jointly cover dynamic rigid-body control, deformable-object handling, and precision assembly (Tab. 1, Fig. 1) to comprehensively evaluate the versatility of our framework. The suite spans single- and dual-arm embodiments (UR5, Franka with LeapHand, xArm–Franka) and two control modes (single-step actions vs. chunked action sequences). The trajectory of each task visualized in images and point clouds can be found in Fig. 3 and 4, respectively.

Key specifications for observation and action spaces of all tasks are summarized in Tab. 2. We randomize initial object placements for each trial to encourage policy generalization, with specific ranges for each task depicted in Fig. 5. Two types of reward structure are used for our tasks. For five tasks except Dynamic Push-T, we use a sparse reward function where the agent receives a reward of +1 upon successful task completion and 0 at all other timesteps, labeled by a human supervisor pressing a keyboard. The Dynamic Push-T task, which requires continuous, high-precision control, utilizes a dense shaped reward. The total reward at each timestep t is $r_t = r_{\text{pose}} + r_{\text{static}} + r_{\text{smooth}}$, where $r_{\text{pose}} = \exp(-3e) - 1$ punishes the SE(3) discrepancy e between the T-block and the desired pose; $r_{\text{static}} = -1$ if the movement of T-block at one timestep is below a given threshold and otherwise 0; $r_{\text{smooth}} = -5\|a_t - a_{t-1}\|_2^2$ punishes jerky actions. More details about our setup (calibration, point-cloud processing, low-level control) are provided in supplementary materials.

Across tasks, RL-100 achieves higher success rates and shorter time-to-completion than baselines and human operators, with especially large gains in settings that require fast online corrections (dynamic contacts, narrow clearances) or stable dexterous grasping under pose variability. A comprehensive robustness protocol evaluates either zero-shot or few-shot transfer across challenging variations (e.g., changed objects, visual distractors, external perturbations). Results show consistent performance retention under these shifts. We defer task-specific objectives and success criteria to the following section, and present aggregate reliability/robustness/efficiency results afterwards.

Description of Tasks

Following the overview, we describe the objective, challenges and further evaluation protocols beyond normal ones of each task individually.

Dynamic Push-T The Dynamic Push-T task requires the robot arm, equipped with a 3D-printed stick-like end-effector, to push a T-shaped block from various initial locations into a wooden slot that is only slightly larger (3mm each side) than the block. The T-block's initial position and orientation are fully randomized, uniformly distributed within a 55cm×60cm reachable workspace of the arm. The robot's initial position and the target slot's position are fixed. The task is considered successful only when the T-block is fully and correctly inserted into the slot.

As a 3D extension of the 2D Push-T task, this task inherits the challenge of requiring high-frequency and precise dynamic adjustments but introduces additional complexities. For instance, due to the slot's geometry, the friction coefficient at the slot's edges varies when the T-block is partially suspended, leading to unpredictable behaviors such as significant rotations of the block. Additionally, the robot must avoid pushing the T-block into the slot in an incorrect orientation, which could lead to failure and end-effector fracture. The policy must therefore exhibit robust control to handle these dynamic and frictional variations while ensuring precise alignment with the slot.

To evaluate the generalization capabilities of the framework, we further perform the task under multiple variations: (i) different initial positions and orientations of the T-block, (ii) a wooden surface with a desktop sticker to reduce friction, (iii) the presence of additional objects on the workspace to introduce visual input distractions, and (iv) external disturbances applied to the T-block's during pushing. These variations challenge the policy to dynamically adjust with high precision, ignore environmental distractions, and avoid incorrect paths, demonstrating its robustness in maintaining high-frequency dynamic control while accurately completing the task.

Agile Bowling The Agile Bowling task requires the robot arm, equipped with a 3D-printed semi-circular end effector, to push a curling stone to knock down six bowling pins positioned 60cm away. The robot arm starts from a fixed initial position, while the curling stone and bowling pins are uniformly distributed along an 18cm starting line and a 20cm target line, respectively. The task is considered successful only when more than five pins are knocked down.

This task presents several challenges due to the highly sensitive dynamics of the setup. The curling surface is extremely smooth, and the end effector is 4mm larger in diameter than the curling stone, meaning even minor variations in the robot's movements can significantly alter the stone's trajectory. The robot must precisely initiate the stone's motion from its starting position, align it toward the bowling pins, and execute a controlled push while making fine adjustments to account for potential trajectory deviations. Furthermore, knocking down almost all pins requires applying sufficient force to accurately strike the first pin, as slight misalignments can result in some pins remaining upright, leading to failure. The small size of both the curling stone and bowling pins provides limited visual input, adding to the challenge of achieving precise control.

To evaluate the generalization capabilities of the framework, we test the task under varying initial positions of the curling stone and bowling pins along their respective lines. We further test (1) zero-shot transferability on a coarser trail surface and (2) adaptation by further finetuning on an inversed pins placement of the policy. These variations challenge the policy to adapt its pushing strategy and trajectory corrections to different configurations while maintaining accuracy.

Pouring This task requires the robot to grasp a cup containing mixed nuts and snacks of varying sizes and

Table 2. State and action components of each task. We abbreviate dimension as 'dim.', proprioception as 'prop.'.

Task	Observation space			Action space	
	Point cloud dim.	Prop. dim.	Note	Dim.	Note
Dynamic Push-T	(512, 3)	6	UR5 joints	2	Normalized $\Delta(x, y)$ of end-effector
Agile Bowling	(512, 3)	6	Same as above	2	Same as above
Pouring	(512, 3)	23 = 7 + 16	Franka joints, Leap hand joints	22 = 6 + 16	Target arm pose, target hand joints
Dynamic Unscrewing	(1024, 3)	23	Same as above	23 = 22 + 1	Same as above, 1: enable arm motion
Soft-towel Folding	(1024, 3)	16 = 2×(7+1)	Dual arm joints + grippers	16 = 2×(7+1)	Dual arms: target pose (pos+quat), target gripper
Orange Juicing – Placing	(1024, 3)	7 = 6 + 1	xArm joints, gripper state	8 = 6 + 1 + 1	Target arm pose, target gripper, task done
Orange Juicing – Removal	(1024, 3)	7	Same as above	8	Same as above

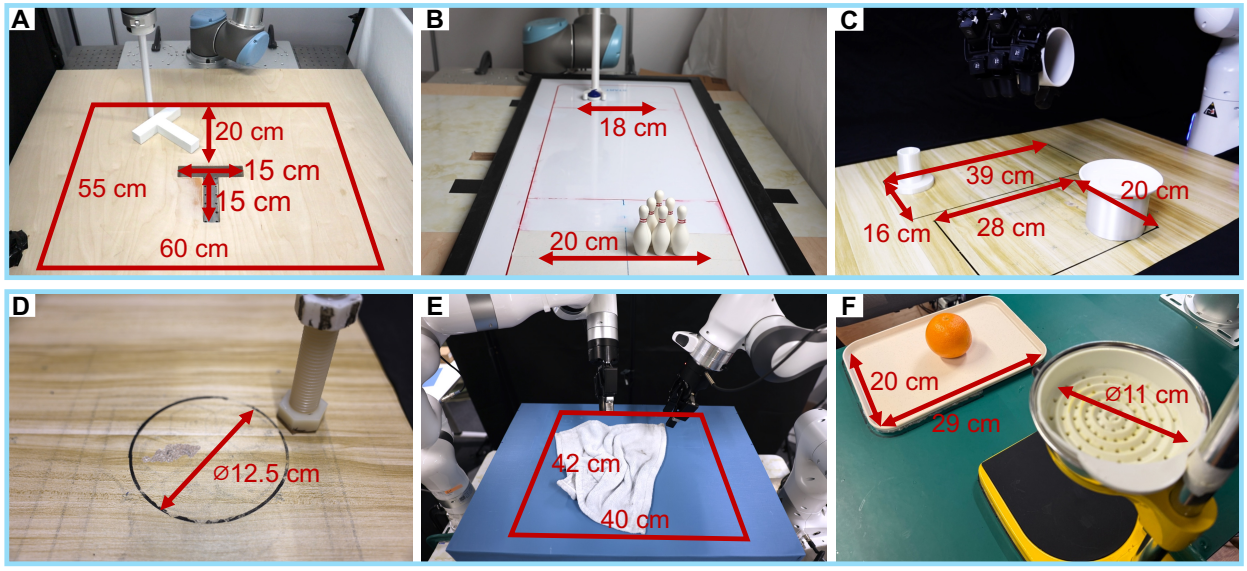

Figure 5. The object initialization workspaces for the seven real-world tasks. At the beginning of each episode, objects-of-interest are randomly positioned within the areas denoted by the red boundaries and annotations.

textures with a LeapHand dexterous hand, then precisely pour the contents into a target plate through controlled wrist rotation. The cup's initial position is randomized within 39cm×16cm on the table surface, while the plate position varies within 28cm×20cm. The robot begins from a fixed initial pose, grasps the cup, rotates the wrist to invert the container, and pours the contents into the plate.

The task is considered successful if the robot maintains a stable cup grasp and accurate alignment during pouring, with no spillage attributable to misalignment; incidental bounce-outs from unavoidable rebounds are disregarded. This task presents multiple challenges that demand sophisticated sensorimotor coordination. First, the cup's smooth surface makes it inherently difficult to grasp securely, requiring the policy to learn appropriate finger configurations that balance grip stability with the ability to maintain a suitable pouring orientation. Second, the substantial randomization in both cup and plate positions necessitates robust spatial generalization. The policy must adapt its grasping strategy and pouring trajectory to varying relative positions while maintaining precision. Third, the mixed contents with different physical properties (varying nut sizes and weights) create unpredictable dynamics during pouring, requiring adaptive control to ensure complete and accurate transfer without overshooting.

To evaluate the generalization capabilities of our framework, we extend this task to three additional variations with distinct physical properties: (i) pouring small soft candies that exhibit different flow characteristics due to their lighter weight and higher elasticity, (ii) pouring water from a cup, which introduces liquid dynamics requiring smooth, continuous motion control to prevent splashing, and (iii) pouring water using a long-spout teapot, which demands adaptation to a different container geometry and requires precise control of the extended spout for accurate targeting. These variations test the policy's ability to transfer learned manipulation skills across different materials (granular solids to liquids) and container morphologies while maintaining the core competency of controlled pouring.

Dynamic Unscrewing This task simulates industrial assembly operations by requiring the robot to unscrew a nut from a vertically-mounted bolt with a LeapHand dexterous hand. The bolt position is randomized within a circle of 12.5cm diameter on the work surface, demanding robust adaptation to varying target locations. The task involves a multi-phase manipulation sequence: the robot first approaches and uses

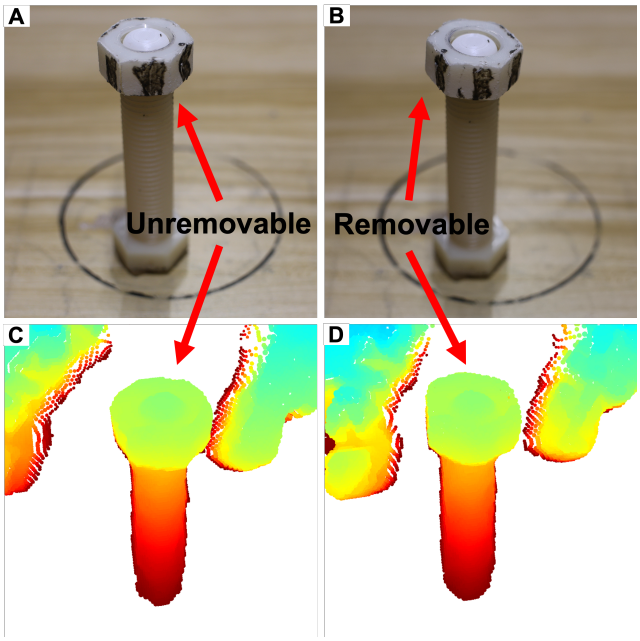

Figure 6. Comparison between images of (A) an unremovable nut and (B) a removable one, as well as their corresponding point-cloud observations (C-D). Robot policies must be accurate enough to recognize the removable state by the tiny tilt shown in (B) and (D) to guarantee success grasps.

its thumb to engage the nut with rotational motions, gradually lifting it along the threaded shaft. As the nut ascends, the robot must continuously adjust its hand pose to maintain effective contact and torque transmission. Once the nut clears the threads, the robot should precisely grasp it with a pinch grip between the index finger and thumb, then transports and places it on the plate.

Success is evaluated based on three criteria: (i) complete unscrewing of the nut, (ii) stable grasping without dropping, and (iii) accurate placement at the target position. This task presents several interconnected challenges that test the limits of dexterous manipulation. First, the initial approach requires precise positioning to establish an effective contact configuration—the thumb must reach a pose that enables sufficient rotational leverage while maintaining clearance for continuous motion. Second, the dynamic nature of the unscrewing process demands adaptive control: as the nut rises along the threads, the optimal contact points and force directions continuously change, requiring the policy to learn time-varying manipulation strategies that maintain consistent rotational motion despite the evolving kinematic constraints. Third, the transition from unscrewing to grasping requires accurate state estimation, as shown in Fig. 6 – the policy must recognize when the nut has cleared the threads and swiftly reconfigure from a rotating contact to a stable pinch grip. Finally, precise grasping of a small object tests fine motor control and hand-eye coordination, as even minor positioning errors can result in unstable grasps or dropped nuts.

Soft-towel Folding This task requires dual robot arms to collaboratively fold a randomly crumpled towel, uniformly distributed in the 42cm×40cm central region of a table, into a neatly folded state. The task is considered successful only when the arms lift and flatten the towel, then perform two precise folds, ensuring no wrinkles or misalignments occur.

This task is challenging due to the need for precise dual-arm coordination and handling a deformable object with significant dynamic variability. The towel's initial position and crumpled configuration vary greatly, leading to substantial differences in point cloud observations. The task is further complicated by potential issues such as uneven folding, unflipped towel corners, or failed grasps, requiring the policy to perform robust failure recovery. Arms must dynamically adjust their motions to flatten and fold the towel accurately while responding to unexpected deformations or missteps.

To evaluate the generalization capabilities of the framework, we test the policy under conditions with external human-induced disturbances during the folding process, assessing its ability to recover from failures and adapt to disruptions. This task challenges the model's capacity for dual-arm coordination, generalization across diverse initial towel configurations, manipulation of deformable objects, and effective error correction.

Orange juicing The orange juicing task consists of three subtasks: placing a halved orange on the juicer, pressing down the lever to extract juice, and removing the discarded orange pulp. The robot's end effector is replaced with two 3D-printed sleeves that fix two rubber grippers, enabling robust grasping irregular fruit surfaces. For each subtask, the initial and final poses of the robot arm are fixed. Note that these poses are defined individually for each subtask and therefore differ across the three subtasks.

Orange placing. In this subtask, a randomly sized half-orange is placed on the inside 29cm×20cm area of a tray at a random position and orientation. The robot first grasps the orange with the gripper vevertically aligned to the surface of the trayhe tray , then rotates its wrist joint by 180° to flip the orange, and finally places it onto the filter section of a lever-style juicer before retracting the gripper. The task is considered successful if the orange is properly placed on top of the filter. This step is challenging due to the high variability in orange size, inclination, height, and placement as shown in Fig. 7A and 7B. The policy must therefore exhibit strong robustness to spatial randomness to achieve reliable placement.

Lever pressing. The pressing process follows a relatively fixed trajectory. To ensure consistent execution, we use kinesthetic teaching to record the downward and upward motions of the lever in advance. During deployment, the robot replays this trajectory to lower the hammer-like press, squeeze the juice out of the orange, and then reset the lever. The task is considered successful if the lever is fully pressed down and restored.

Discard removal. After pressing, the flattened orange half is randomly embedded within the 11cm diameter juicer filter. The robot begins with the gripper closed and aligned parallel to the filter surface, pushing against the flattened fruit to move it from the tightly embedded position. The wrist then rotates by 90°, the gripper opens to grasp the orange, and the discarded pulp is placed into a disposal container on the left side of the juicer. The task is considered successful if the flattened orange is removed from the filter

(A) Various appearances of halves.

(B) Different sizes of oranges.

(C) Proper force is needed for removal.

Figure 7. Challenging parts of the orange juicing task: (A)-(B) spatial robustness w.r.t. orange appearances and positions; (C) force-sensitive manipulation of deformable orange discards.

Table 3. Success rate (%) across tasks. RL–100 (ours) groups an iterative offline fine-tuning stage followed by online RL with either DDIM or one-step CM policy. Averages are unweighted across tasks.

Task	Imitation baselines		RL-100 (ours)		
	DP-2D	DP3	Iterative Offline RL	Online RL (DDIM)	Online RL (CM)
Dynamic Push-T	40 (20/50)	64 (32/50)	90 (45/50)	100 (50/50)	100 (50/50)
Agile Bowling	14 (7/50)	80 (40/50)	88 (44/50)	100 (50/50)	100 (50/50)
Pouring	42 (21/50)	48 (24/50)	92 (46/50)	100 (50/50)	100 (50/50)
Soft-towel Folding	46 (23/50)	68 (34/50)	94 (47/50)	100 (50/50)	100 (250/250)
Dynamic Unscrewing	82 (41/50)	70 (35/50)	94 (47/50)	100 (50/50)	100 (50/50)
Orange Juicing – Placing	78 (39/50)	88 (44/50)	94 (47/50)	100 (100/100)	100 (50/50)
Orange Juicing – Removal	48 (24/50)	76 (38/50)	86 (43/50)	100 (50/50)	—
Mean (unweighted)	**50.0**	**70.6**	**91.1**	**100.0**	**100.0**†

† Mean over the six tasks evaluated with CM.

and dropped into the container. This subtask is particularly challenging because it involves force-sensitive manipulation of a deformable object in a confined space. Fig. 7C indicates that sufficient force is needed to move the orange, but excessive force can cause deformation and hinder grasping. Furthermore, juice on the surface makes the orange slippery, and inappropriate grasping poses may result in failed pickups.

Results on Real Robots

Main results

Tab. 3 traces performance from imitation-only policies to our RL-100 method. The imitation baselines produce only moderate results. The DP-2D baseline attains an average success of 50.0% while DP3 improves to 70.6%. Both baselines are especially challenged by tasks with dynamic or deformable objects, as shown by Agile Bowling with success rates of 14% and 80% respectively, and by Pouring with success rates of 42% and 48% respectively.

The iterative offline RL phase of RL-100 yields a large improvement and raises average success to 91.1%. This stage produces the largest absolute gains on the hardest tasks, namely an increase of 8 points over DP3 on Agile Bowling with 88% versus 80%, an increase of 44 points on Pouring with 92% versus 48%, and an increase of 24 points on Dynamic Unscrewing with 94% versus 70%. These results indicate that conservative model-guided policy refinement reliably improves performance across diverse manipulation regimes while avoiding the instability associated with naive online updates.

The full RL-100 pipeline, consisting of iterative offline RL followed by brief on-policy online refinement, attains perfect success across all evaluated trials. With the DDIM policy, we observe 400 successful episodes out of 400 across seven tasks. The consistency-model variant evaluated on six tasks achieves the same perfect success rate with 500 successful episodes out of 500 and enables single-step high-frequency control, including 250 successful trials out of 250 on the challenging dual-arm soft-towel folding task. Overall, the results highlight the contribution of each component, namely that OPE-gated iterative learning delivers substantial and reliable improvements under a conservative regime and that short on-policy fine-tuning together with consistency distillation removes remaining failure modes and produces computationally efficient deployment-ready policies.

Zero-shot adaptation to new dynamics

To assess the robustness of our learned policies, we evaluate zero-shot transfer to environmental variations unseen during training, listed in Tab. 4 and Fig. 8. On Dynamic Push-T, as shown in Fig. 8B-C, the policy maintains perfect 100% success when the surface friction is substantially altered, and achieves 80% success even when visual and physical clutter from interference objects of various shapes are introduced into the workspace. Similarly, Agile Bowling in Fig. 8D achieves 100% success on modified surface properties that change ball rolling dynamics. For Pouring in Fig. 8A, replacing granular nuts with fluid water—a significant shift in material properties and flow dynamics—still yields 90% success, demonstrating that the policy has learned sufficiently stable and robust manipulation strategies rather than overfitting to specific physical parameters. Across

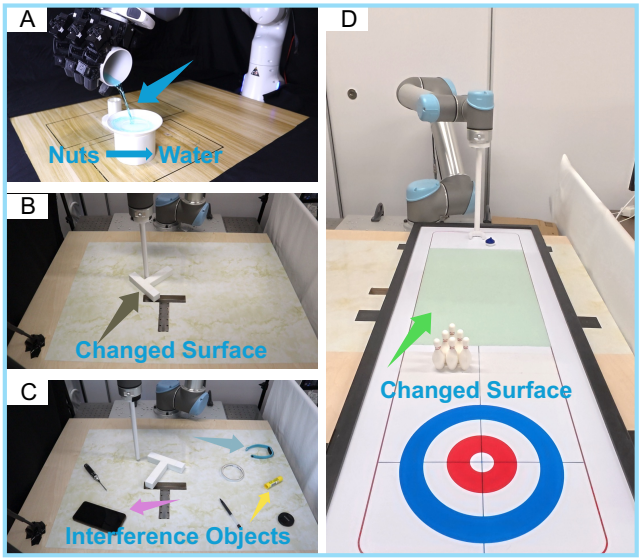

Figure 8. Zero-shot adaptation to novel dynamics. Our method generalizes to unseen physical variations without retraining. **(A)** Pouring with different granular/fluid materials. **(B)** Dynamic Push-T on altered surface friction. **(C)** Dynamic Push-T with visual and physical interference objects. **(D)** Agile Bowling on modified surface properties with a target-based scoring system. Red arrows highlight the changed environmental conditions.

Table 4. Zero-shot generalization success rates (%) on novel dynamics and environmental variations. All policies are trained on nominal conditions and evaluated without any retraining or fine-tuning.

Task Variation	Success Rate (%)
Pouring (Water)	90
Push-T (Changed surface)	100
Push-T (Interference Objects)	80
Bowling (Changed Surface)	100
Average	**92.5**

all four variations, our method achieves 92.5% average success without any retraining or fine-tuning, indicating strong generalization to distribution shifts in environmental dynamics.

Few-shot adaptation

Beyond zero-shot generalization, we evaluate the sample efficiency of adapting to more substantial task modifications through brief fine-tuning, as shown in Tab. 5 and Fig. 9. After only 1–3 hours of additional training on each variation, our policies achieve high success rates across diverse changes. The Soft-towel Folding policy adapts perfectly to a different towel material with altered deformable properties, achieving 100% success rate as shown in Fig. 9A, demonstrating effective transfer despite significant changes in cloth dynamics. Similarly, Agile Bowling achieves 100% success on an inverted pin arrangement listed in Fig. 9C, requiring the policy to adapt its trajectory planning and aiming strategy. The Pouring task with a modified container geometry, as shown in Fig. 9B, achieves 60% success, showing that while geometric changes require more adaptation, the policy can still leverage its learned

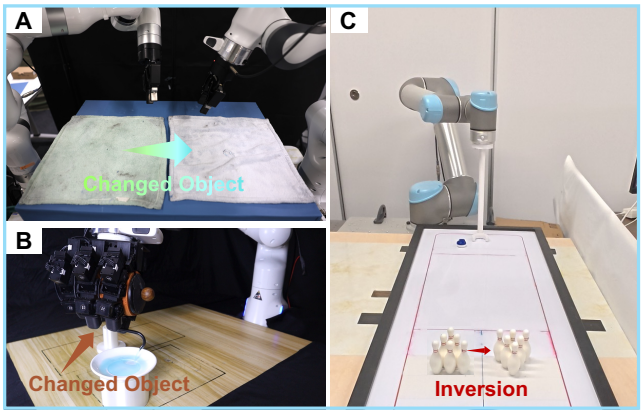

Figure 9. Few-shot adaptation to novel task variations. Our method rapidly adapts to unseen scenarios with minimal additional training data. **(A)** Soft-towel Folding with a different towel material exhibiting altered deformable properties. **(B)** Pouring with a modified container shape and size. **(C)** Agile Bowling with inverted pin arrangement requiring adapted trajectory planning.

Table 5. Few-shot adaptation success rates (%) after brief fine-tuning (1–3 hours) on novel task variations.

Task Variation	Success Rate (%)
Pour (New Container)	60
Folding (Changed Object)	100
Bowling (Inverted pin)	100
Average	**86.7**

manipulation primitives. Averaging 86.7% success across these variations with minimal retraining demonstrates the sample efficiency and adaptability of our approach, highlighting that the learned representations and control strategies effectively transfer to modified task configurations.

Robustness to physical disturbances

We evaluate the resilience of RL-100 policies to real-world physical perturbations by introducing human-applied disturbances during task execution (Tab. 6, Fig. 10). For Soft-towel Folding, we apply external forces at two critical stages: during initial grasping (Stage 1, Fig. 10A) and during pre-folding manipulation (Stage 2, Fig. 10B), achieving 90% success at both stages despite the perturbations. On the Dynamic Unscrewing task (Fig. 10C), we apply sustained counter-rotational **interference for up to 4 seconds** during both the unscrewing motion and at the critical zero-boundary point where visual judgment is required to grasp the nut, a particularly challenging moment as described in our task specifications. Remarkably, the policy achieves 100% success, demonstrating the ability to recover from prolonged disturbances even during precision-critical manipulation phases. Similarly, Dynamic Push-T (Fig. 10D) achieves 100% success despite **multiple dragging perturbations** applied throughout the pushing motion. Across all tested scenarios, our policies maintain an average 95.0% success rate, validating that the closed-loop control learned through RL-100 enables robust recovery and task completion in the presence of external perturbations typical of unstructured real-world environments.

Figure 10. Robust recovery from real-world disturbances. RL-100 policies demonstrate resilience to external perturbations across diverse manipulation tasks. **(A–B)** Soft-towel Folding recovers from external pulling and lateral dragging at different manipulation stages. **(C)** Dynamic Unscrewing withstands counter-rotational interference from human hand. **(D)** Dynamic Push-T handles dragging perturbations to the target object. Orange arrows indicate disturbance directions and timing. The closed-loop policies successfully recover and complete all tasks, demonstrating robustness critical for unstructured real-world environments.

Table 6. Task completion success rates (%) after recovering from physical disturbances.

Task & Disturbance Stage	Success Rate (%)
Folding (Stage 1: Grasping)	90
Folding (Stage 2: Pre-folding)	90
Unscrewing	100
Push-T (Whole stage)	100
Average	**95.0**

Execution Efficiency

We evaluate the execution efficiency of RL-100's policies from four complementary perspectives, with results presented in Fig. 11.

Episode length in successful trajectories. We first compare the number of environment steps needed to complete a task successfully (Fig. 11A). RL-finetuned policies (RL-100) consistently outperform imitation baselines across all three tasks. This advantage stems from reward-driven policy optimization, which steers the agent toward more efficient—and potentially optimal—trajectories for earlier task completion. Specifically, RL-100 achieves substantial step reductions: in *Soft-towel Folding*, 390 steps (DP-2D) $\rightarrow$ 312 steps (RL-100 CM; $1.25\times$ fewer), and in *Dynamics Unscrew*, 361 steps (DP-2D) $\rightarrow$ 280 steps (RL-100 DDIM; $1.29\times$ fewer). These gains reflect the value-driven objective's tendency to produce more decisive state transitions.

Wall-clock time per episode. Beyond step count, wall-clock time is the most intuitive metric for real-world deployment, especially in dynamic tasks requiring real-time action prediction. Fig. 11B reveals that inference latency depends on two factors: (i) *Input modality*—point clouds (DP3, 1024 points) incur lower computational overhead than 128×128 RGB images (DP-2D), yielding 1.02–$1.31\times$ speedup despite similar step counts; (ii) *Inference method*—RL-100 (CM)'s one-step generation per action reduces control-loop latency compared to the 10-step DDIM sampler, providing an additional 1.05–$1.16\times$ wall-clock speedup over RL-100 (DDIM). For example, in *Orange Juicing–Placing*, RL-100

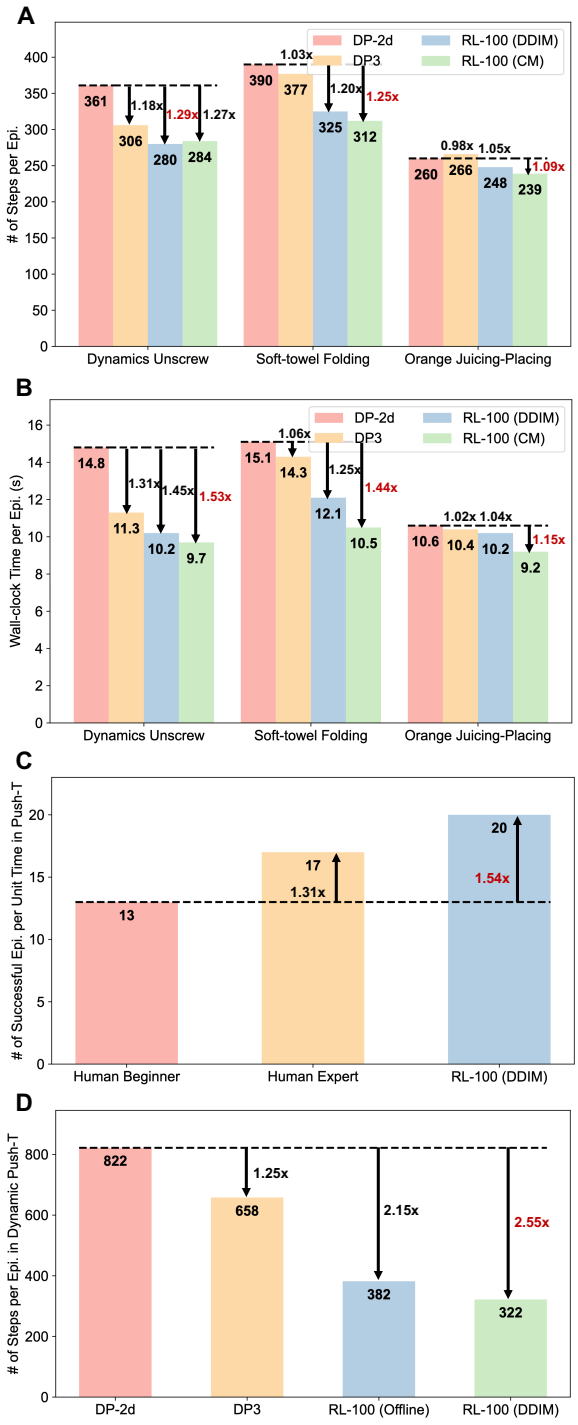

Figure 11. Comparison of RL-100 against algorithmic baselines or human operators in terms of task completion efficiency. (A) Episode length (# of environmental steps) across three tasks; lower is better. (B) Wall-clock time (s) to finish one episode; lower is better. (C) Dynamic Push-T throughput (# of successful episodes per unit time); higher is better. (D) Episode length in Dynamic Push-T; lower is better. Bars show absolute values; annotations indicate speedup factors relative to the baseline (DP-2D for A, B, and D; Human Beginner for C). We abbreviate 'episode' as 'epi.'.

(CM) completes episodes in 9.2s versus 10.2s for RL-100 (DDIM) (1.11× faster) and 10.6s for DP-2D (1.15× faster).

RL-100 vs. human operators. A critical question for practical deployment is whether robotic policies can match or exceed human efficiency. To address this, we compare RL-100 against human teleoperators on *Dynamic Push-T* (Fig. 11C). RL-100 (DDIM) achieves 20 successful episodes per unit time, surpassing the human expert who collected the demonstrations (17 episodes; 1.18× improvement) and substantially exceeding human beginners (13 episodes; 1.54× improvement). This demonstrates that our learned policy not only matches expert-level task completion quality but also executes more efficiently in repetitive scenarios.

Episode length including failures. The previous comparisons focused solely on successful trajectories. However, a comprehensive efficiency analysis must account for all attempts, including failures. Fig. 11D shows that when considering all trials on *Dynamic Push-T*, RL-100 maintains significantly shorter average horizons: 822 steps (DP-2D) → 658 steps (DP3) → 382 steps (RL-100 Offline) → 322 steps (RL-100 DDIM), representing up to 2.55× fewer steps than DP-2D. This indicates that RL-100 produces policies that both succeed more reliably and terminate unpromising attempts quickly, avoiding wasted execution time.

Summary. Overall, the efficiency gains of RL-100 arise from three synergistic sources: (1) *efficient encoding* (point clouds over RGB images), (2) *reward-driven policy optimization* (shorter, more decisive rollouts via the discount factor $\gamma < 1$), and (3) *low-latency action generation* (CM's one-step versus DDIM's 10-step inference). The monotonic improvement—DP-2D → DP3 → RL-100 (DDIM) → RL-100 (CM)—validates each design choice's contribution. **Crucially, RL-100's execution efficiency surpasses both algorithmic baselines and human operators, demonstrating readiness for practical deployment.**

Training efficiency

Fig. 12 shows the online RL training progression for Agile Bowling. The policy achieves consistent 100% success after approximately 120 episodes of on-policy rollouts, with a moving average demonstrating stable learning without performance drop from offline. The final 100+ episodes maintain perfect success, validating robust policy convergence. *These success rates reflect training rollout episodes with exploration noise, not formal evaluations. Since rollouts involve noise for exploration, while evaluation uses deterministic actions, the reported rates underestimate the true policy performance. However, frequent evaluation on physical hardware is prohibitively costly. We therefore track learning progress through rollout success rates as an efficient proxy, providing real-time visibility into policy improvement. We present this curve for Agile Bowling as it exemplifies the rapid convergence and stability characteristic of our online RL finetuning phase.*

Simulation Experiments

To complement our real-robot evaluations, we benchmark RL-100 in simulation and compare it against state-of-the-art diffusion/flow–based offline-to-online RL methods, including DPPO (Ren et al. 2024), ReinFlow (Zhang et al. 2025), and DSRL (Wagenmaker et al. 2025). We also conduct ablations to verify the contribution of each component in RL-100.

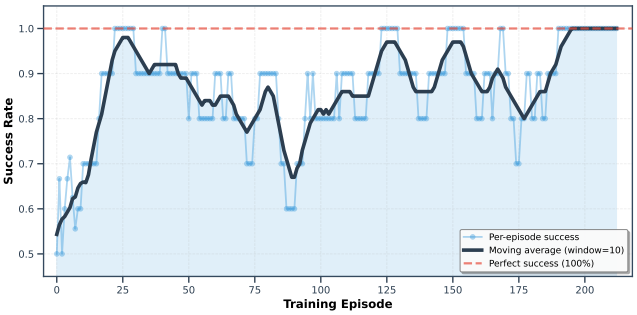

Figure 12. Online RL training curve for Agile Bowling. Per-episode success rate during online RL finetuning phase, showing rapid convergence from the iterative offline baseline (approximately 80+%) to perfect performance (100%).

Main Results

We evaluate on three standard continuous-control suites that cover both dexterous manipulation and locomotion: **Adroit** (Rajeswaran et al. 2018), **Meta-World** (Yu et al. 2020), and **MuJoCo locomotion** (Fu et al. 2020; Todorov et al. 2012). Unless noted otherwise, observations include robot proprioception together with visual inputs (RGB frames or point clouds), and actions are continuous joint/EE commands. We report normalized returns and task success rates (mean ± std over multiple seeds) and compare RL-100 with baselines finetuning diffusion/flow policies as well as their offline/online training protocols.

As shown in Fig. 13, RL-100, shown in red curves, consistently achieves the highest or near-highest performance across all ten tasks. On MuJoCo locomotion, RL-100 reaches 10,000 return on halfcheetah-medium-v2, which is 2.2× higher than DPPO's 4,500 and 3.3× higher than DSRL's 3,000, and substantially outperforms all baselines on hopper and walker2d. On Adroit dexterous tasks, RL-100 attains near-perfect success rates, about 100%, on door and hammer, while DPPO plateaus at 0.9 and ReinFlow struggles below 0.6 on hammer. For Meta-World precision tasks like peg-insert-side, RL-100 maintains a stable 1.0 success rate after a few steps, where ReinFlow fails to exceed 0.2.

From a sample efficiency and stability perspective, RL-100 demonstrates faster convergence—achieving peak performance within 1-2M steps on most tasks versus 3-5M for baselines—and exhibits notably lower variance (narrower confidence intervals) across random seeds. On challenging coordination tasks (pen, disassemble), RL-100's advantage is most pronounced, showing both higher asymptotic performance and more stable learning trajectories. The consistent gains across diverse modalities—from high-dimensional locomotion to contact-rich manipulation and precision insertion—validate RL-100's generality as a unified framework for diffusion policy fine-tuning.

Policy Inference Frequency Fig. 14 presents a comparative analysis of our proposed methods, RL-100 (CM), against representative baselines in terms of model size (number of parameters) and policy inference speed (frequency in

Table 7. Summary of numbers of episodes and corresponding collection time (h) across all tasks during the three training stages - human demonstration, iterative offline RL and online RL. Note that in offline RL stage, the number is the sum of all iterations.

Task	Human Demonstration		Iterative Offline RL		Online RL	
	# of epi.	Collection time (h)	# of epi.	Collection time (h)	# of epi.	Collection time (h)
Dynamic Push-T	100	2	821	8	763	7.5
Agile Bowling	100	2	249	2	213	2.5
Pouring	64	1	741	6.8	129	1.5
Soft-towel Folding	400	5	896	11	654	8.5
Dynamic Unscrew	31	0.5	467	4.5	288	3
Orange Juicing – Placing	80	1.5	642	10.5	750	12.5
Orange Juicing – Removal	29	0.5	149	2.5	240	4

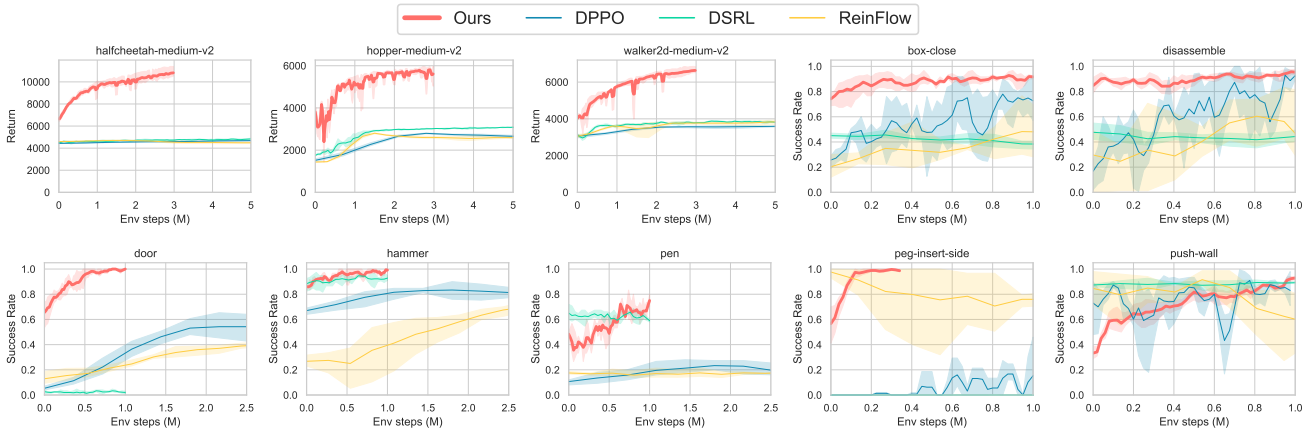

Figure 13. Learning curves of various methods on MuJoCo locomotion, Adroit, Meta-World tasks are presented across three random seeds. Solid lines indicate the mean performance, while the shaded regions represent the corresponding 95% confidence interval. We use robot proprioception as the policy input for MuJoCo locomotion, and both 3D visual observations and proprioceptive state for Adroit (Door, Hammer, Pen) and Meta-World (Box-Close, Disassemble, Peg-Insert-Side, Push-Wall); baseline methods follow the input modalities specified in their original papers.

Hz). The results demonstrate the superior efficiency of our approach. Specifically, RL-100 (CM) with a Skip-Net architecture achieves an exceptional inference frequency of 378 Hz, outperforming DSRL (35 Hz) and DPPO (30 Hz) by over an order of magnitude, while maintaining a compact model size of just 3.9M parameters w.r.t. DSRL (52.3M) . Even a larger U-Net architecture (39.2M) enjoys an inference speed boost up to 133 Hz via CM. Reinflow benefits from a relatively high frequency due to its smaller network size, but it is still notably lower than ours, even with a smaller model.

This significant performance gain is attributed to our core methodology: we distill a diffusion policy into a Consistency Model (CM) capable of single-step inference alongside training/finetuning the original policy. Unlike diffusion-based policies that inherently rely on a slow, multi-step iterative sampling process, CM collapses policy generation into a single, efficient forward pass. Consequently, we achieve near real-time policy inference, effectively removing the decision-making model as a bottleneck. The overall system frequency becomes primarily limited by the speed of the perception system (e.g., camera frame rate at 30Hz), enabling truly reactive control and timely responses. This low latency is critical for robust performance in dynamic environments, where the system delay introduced by iterative samplers can severely hinder task success, which has been shown in our real-world experiments.

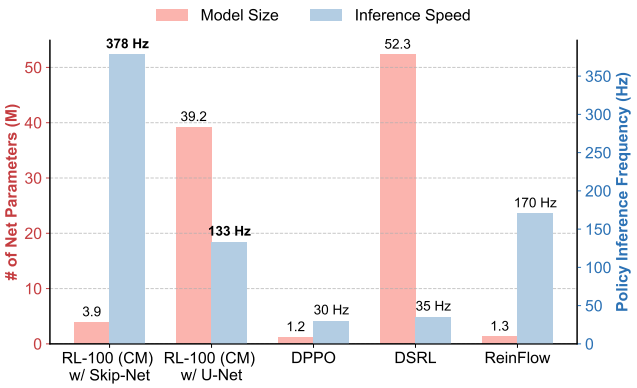

Figure 14. Comparison of two RL-100 variations with different action heads against other baselines in simulation in terms of model size and inference speed. Model size is defined as the number of parameters in a policy neural net, while inference speed is the frequency of a policy outputting an action given observations at each timestep.

Ablation Studies

We conduct ablation studies on five components: (i) visual modality (2D vs. 3D), (ii) diffusion-noise clipping, (iii) sampler/model (CM vs. DDIM), (iv) representation learning during fine-tuning (ReconVIB / no ReconVIB / frozen encoder), and (v) diffusion parameterization (ϵ- vs. x_0-prediction). The comparison is listed in Fig. 15 and the

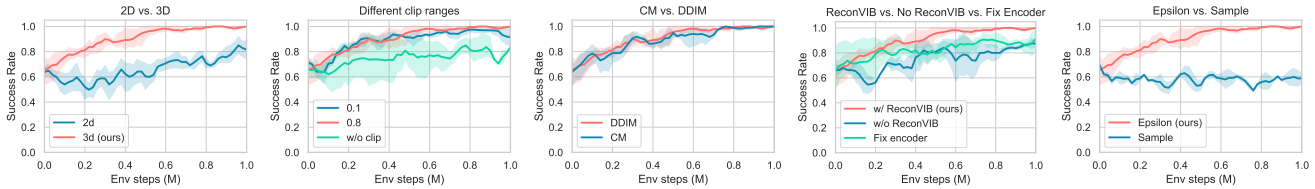

Figure 15. Learning curves of various ablation studies on the door task in Adroit are presented across three random seeds. Solid lines indicate the mean performance, while the shaded regions represent the corresponding 95% confidence interval. From left to right: (A) 2D vision modality (i.e., images) vs. 3D point clouds used by RL-100. (B) Different clip ranges leveraged to bound the standard deviation of stochastic DDIM sampling steps. (C) Different models in RL-100. (D) Whether or not including reconstruction and VIB loss during finetuning as well as fixing encoder during finetuning. (E) Different diffusion parameterizations (i.e., the output of neural net; epsilon: predicting noises, sample: predicting clean samples.

detailed configurations and results are provided below.

Visual modality: 2D vs. 3D. Our framework is *modality-agnostic* and accepts either 2D RGB images or 3D point clouds as visual inputs. On the Adroit *door* task—a relatively clean scene—the 3D variant learns faster and attains a higher final success rate. We attribute this to the ability of 3D inputs to enable precise geometric filtering (e.g., ROI/radius "clipping") that cleanly isolates task-relevant structures such as the handle and contact surfaces, yielding a higher signal-to-noise ratio and easier credit assignment than 2D images. This ablation thus demonstrates both the flexibility of our framework (2D/3D optional) and the practical advantage of 3D in clean, contact-rich settings.

Clipping the diffusion noise scale. Bounding the standard deviation of stochastic DDIM steps with a *moderate* clip (e.g., 0.8) yields the best stability–performance trade-off. An overly tight bound (0.1) under-explores and converges lower, whereas no clipping induces pronounced oscillations and wider intervals. We use 0.8 for Adroit, Mujoco Locomotion tasks, and real-world single-action mode tasks, and 0.1 for Meta-World and real-world chunk-action mode tasks.

Consistency model (CM) vs. DDIM. One-step CM matches multi-step DDIM performance with $K\times$ speedup. The distilled consistency model (CM, blue) achieves nearly identical learning curves and final success rates ($\sim$1.0) compared to the K-step DDIM policy (red), validating that our lightweight distillation successfully preserves task performance while enabling single-forward-pass inference. Both policies converge at similar rates and maintain comparable stability (overlapping confidence intervals), demonstrating that the CM effectively compresses the iterative denoising process without sacrificing control quality. This enables high-frequency deployment (100+ Hz) critical for dynamic real-world tasks.

Representation learning during fine-tuning. Including reconstruction plus VIB losses while updating the encoder achieves the highest and most stable success. Removing these losses degrades stability and final performance, and freezing the encoder further limits gains; joint policy–representation adaptation mitigates representational drift and improves sample efficiency.

Diffusion parameterization: ϵ vs. x_0. Noise prediction (ϵ-prediction) substantially outperforms clean-sample prediction (x_0-prediction), achieving $\sim$40% higher final success ($\sim$1.0 vs. $\sim$0.6) with faster learning and reduced variance—suggesting better-conditioned gradients and closer alignment with the PPO-style objective used during RL post-training.

Prediction parameterization analysis Prior work on diffusion models for robot learning (e.g., DP3) predominantly adopts the x_0–prediction parameterization rather than ϵ–prediction. The two parameterizations recover the clean action (or action chunk) $\hat{x}_0$ as

$$\hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\, f_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} \qquad (\epsilon\text{–prediction}), \tag{24}$$

$$\hat{x}_0 = f_\theta(x_t, t) \qquad (x_0\text{–prediction}). \tag{25}$$

Analysis. A practical reason is variance amplification in (24): early in the reverse process ($t \to T$), the factor $\frac{1}{\sqrt{\bar{\alpha}_t}}$ becomes large, which magnifies estimation noise in ε_θ and yields a higher-variance $\hat{x}_0$. In supervised imitation settings (or when actions are executed open-loop), this extra variance can destabilize training and control, making the direct x_0–prediction attractive.

Variance measurements. We pretrain identical policies under the two parameterizations and, during environment rollouts, perform 100 forward passes per time step to estimate predictive variance. On a locomotion task (Hopper), the empirical variance of $\hat{x}_0$ is 0.1316 for ϵ–prediction versus 0.0870 for x_0–prediction; on a manipulation task (Adroit–Door) it is 0.0589 **for ϵ–prediction versus** 0.0290 **for x_0–prediction**, respectively.

Implication for online RL. In our *two-level MDP*—where each action is produced by a K-step denoising trajectory—the higher stochasticity induced by ϵ–prediction acts as structured exploration *within* the denoising chain, improving coverage of latent action modes and helping avoid local optima during policy-gradient fine-tuning. With all other components fixed, online RL with ϵ–prediction achieves faster and more reliable improvement on Adroit–Door (see Fig. 15), whereas x_0–prediction exhibits slower improvement and more frequent premature convergence.

Takeaway. While x_0–prediction can be preferable for low-variance imitation and open-loop execution, the variance amplification inherent to ϵ–prediction is beneficial for *exploration* in diffusion-based online RL. Consequently, we adopt ϵ–prediction for post-training, leveraging its exploratory behavior inside the denoising steps.

Conclusion and Future Work

This paper focuses on the core challenges of real-world deployment of robot learning, which demands systems with

high reliability, efficiency, and robustness. To this end, we present RL-100, a three-stage real-world RL training framework built atop a diffusion-based visuomotor policy, designed to "start from human priors" and then "go beyond human performance" in robotic manipulation. By unifying imitation learning and reinforcement learning under a single policy optimization objective, RL-100 preserves the strengths of human teleoperation demonstrations while continuously self-improving through autonomous exploration. In particular, a shared PPO-style policy-gradient loss is applied across both offline and online RL phases to fine-tune the diffusion policy's denoising process, providing stable, near-monotonic performance gains. Moreover, the framework introduces a consistency distillation technique that compresses the multi-step diffusion policy (requiring $K = 5$–10 denoising steps) into a single-step consistency model, enabling high-frequency control at deployment without sacrificing performance. This unification of IL and RL under one objective, coupled with the one-step distilled policy, yields a controller that is both efficient to fine-tune and fast to execute in real-world settings.

RL-100 's efficacy was demonstrated across a diverse suite of real-world tasks, encompassing dynamic object pushing, dual-arm deformable cloth folding, agile high-speed bowling, precision pouring of granular and fluid materials, unscrewing, and a multi-stage orange juicing task. Deployment-level results show that the fully-trained RL-100 policy achieved 100% success rates on every task, attaining virtually perfect reliability in extensive evaluations. Notably, RL-100 also improved task efficiency, often matching or even surpassing skilled human operators in time-to-completion on multiple tasks. The trained policies proved robust over long durations, maintaining stable performance over 2+ hours of continuous operation, which attests to the framework's long-horizon reliability. These results collectively indicate that RL-100 satisfies the stringent reliability, efficiency, and robustness requirements needed for real-world deployment.

Beyond excelling under nominal conditions, RL-100 exhibited strong generalization and robustness in the face of new variations and disturbances. Without any retraining, a single policy maintained high performance under significant environmental shifts. Across several such unseen scenario variations, RL-100 attained an average 92.5% success rate without any fine-tuning, underscoring the policy's robustness to modest domain shifts. Furthermore, the framework's consistency-model variant (the one-step distilled policy) demonstrated that high-frequency control is possible without loss of accuracy, as it too reached a 100% success rate on all evaluated tasks, including 250/250 successes in the challenging dual-arm towel folding. This blend of reliability and adaptability indicates that RL-100 can handle the kind of dynamic perturbations and variability inherent in real home or factory environments, rather than overfitting to narrow training conditions.

In summary, RL-100 integrates imitation learning and reinforcement learning into a unified paradigm that yields deployment-grade performance on real robots. It leverages human demonstration data as a foundation and then aligns to human-grounded objectives while exceeding human-level reliability through iterative self-improvement. The framework's near-perfect success rates, broad task generality, and sustained operational robustness represent a significant step toward deploying autonomous manipulators in unstructured environments. These outcomes "offer a credible path to deployment in homes and factories", where robots must perform daily manipulation tasks with super-human consistency and safety.

Future Work. There are several promising directions to extend and enhance this work. A priority is to extend evaluation to more complex, cluttered, and partially observable scenes that better mirror the variability of homes and factories. This includes dynamic multi-object settings, occlusions, specular/transparent materials, changing illumination, and non-stationary layouts. Stress-testing the policy beyond the current benchmark—where we already observe near-perfect success, strong long-horizon stability, and human teleoperation-level time-to-completion on multiple tasks—will help quantify limits and failure modes under realistic deployment conditions.

Our results indicate that small diffusion policies can attain high reliability and efficiency after modest fine-tuning. Building on this, we plan to scale post-training to larger, multi-task, multi-robot Vision-Language-Action (VLA) models. Key directions include: (i) scaling laws for data/model size vs. real-robot sample efficiency; (ii) cross-embodiment and cross-task transfer in a single policy; and (iii) aligning large VLA priors with RL-100 's unified objective to retain zero-shot semantic generalization while preserving the high success rates we report.

Although our pipeline already incorporates conservative operation and stable fine-tuning, reset and recovery remain practical bottlenecks. We will investigate autonomous reset mechanisms—learned reset policies, scripted recovery behaviors, task-aware environment fixtures, and failure-aware action chunking—to minimize on-robot human effort during training and evaluation. A principled reset strategy should reduce downtime, smooth data collection, and further stabilize online improvement, complementing RL-100 's OPE-gated iterative updates.

Contribution List

Kun: Project lead, Core, responsible for the original idea and codebase, overall design and training of simulation and real-world experiments, and paper writing.
Huanyu: Core, optimized training infrastructure, led the unscrew, pour and juicing tasks, contributed to paper writing, and implemented part of the simulation baseline.
Dongjie: Core, led the juicing task, contributed to paper writing, most of the real-world baselines, and implemented part of the simulation baseline.
Zhenyu: Core, responsible for the real-world robot setup, design, and data collection, and contributed to paper writing.
Lingxiao: Core, led data collection for the folding task, managed its offline training process, and explored dual-arm embodiments in the early stages.
Zhennan: Core, contributed to part of the algorithm design, improved consistency policy distillation, debugged and explored settings in simulation, and developed most of the simulation baselines.
Ziyu: Managed the Metaworld domain tasks.

Shiyu: Writing refinement
Huazhe: Principal Investigator (PI), Core, responsible for project direction and guidance, and contributed to paper writing.

Acknowledgement

We would like to thank Tianhai Liang for his contributions to the hardware design for this project, and Jiacheng You and Zhecheng Yuan for their valuable discussions and insights.

References

Ajay A, Du Y, Gupta A, Tenenbaum J, Jaakkola T and Barbu A (2023) Is conditional generative modeling all you need for decision making? *arXiv preprint arXiv:2211.15657* .

Ball PJ, Smith L, Kostrikov I and Levine S (2023) Efficient online reinforcement learning with offline data. In: *International Conference on Machine Learning*.

Black K, Brown N, Driess D, Esmail A, Equi M, Finn C, Fusai N, Groom L, Hausman K, Ichter B, Jakubczak S, Jones T, Ke L, Levine S, Li-Bell A, Mothukuri M, Nair S, Pertsch K, Shi LX, Tanner J, Vuong Q, Walling A, Wang H and Zhilinsky U (2024a) π_0: A vision-language-action flow model for general robot control. URL https://arxiv.org/abs/2410.24164.

Black K, Brown N, Driess D, Ko AE, Hu K, Lee A, Nakayama M, Singh A, Xu Q, Yang Q et al. (2024b) π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164* .

Black K, Janner M, Du Y, Kostrikov I and Levine S (2023) Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301* .

Chen H, Lu C, Ying C, Su H and Zhu J (2023) Score regularized policy optimization through diffusion behavior. *arXiv preprint arXiv:2310.07297* .

Chen Q, Deng Z, Wang Y, Bao J, Lu Y, Zeng W, Niebles JC, Gao R, Li FF and Wu J (2024) Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. *arXiv preprint arXiv:2412.04987* .

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

Christiano PF, Leike J, Brown TB et al. (2017) Deep reinforcement learning from human preferences. In: *NeurIPS*.

Chua K, Calandra R, McAllister R and Levine S (2018) Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In: *NeurIPS*.

Cui J and Trinkle J (2021) Toward next-generation learned robot manipulation. *Science Robotics* 6(54): eabd9461. DOI:10.1126/scirobotics.abd9461. URL https://www.science.org/doi/abs/10.1126/scirobotics.abd9461.

Ding S, Hu C, Liu Z, Zhang C and Liang Y (2024a) Diffusion-based reinforcement learning via q-weighted variational policy optimization. In: *Advances in Neural Information Processing Systems*.

Ding Z, Zhang C, Dong J and Jia J (2024b) Consistency models as a rich and efficient policy class for reinforcement learning. *arXiv preprint arXiv:2309.16984* .

Eysenbach B, Gu S, Ibarz J and Levine S (2018) Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In: *International Conference on Learning Representations*.

Fan Y, Watkins O, Du Y, Liu H, Ryu M, Goodman C, Jain A, Vafa K, Kim D, Yu J, Tang H, Hao T, Tenenbaum J, Garg S, Jesse A, Abbeel P and Singh AK (2024) DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems* .

Fu J, Kumar A, Nachum O, Tucker G and Levine S (2020) D4rl: Datasets for deep data-driven reinforcement learning.

Fujimoto S, van Hoof H and Meger D (2018) Addressing function approximation error in actor-critic methods. In: *International Conference on Machine Learning (ICML)*.

Guo L, Xue Z, Xu Z and Xu H (2025) Demospeedup: Accelerating visuomotor policies via entropy-guided demonstration acceleration. URL https://arxiv.org/abs/2506.05064.

Gupta A, Mendonca R, Zhao Y, Emmert-Streib F and Levine S (2021) Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In: *IEEE International Conference on Robotics and Automation*.

Haarnoja T, Zhou A, Abbeel P and Levine S (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: *International Conference on Machine Learning (ICML)*.

Hansen-Estruch P, Kostrikov I, Janner M, Kuba JG and Levine S (2024) Revisiting diffusion q-learning: From iterative denoising to one-step action generation. In: *International Conference on Machine Learning*.

He H, Bai C, Xu K, Yang Z, Zhang W, Wang D, Zhao B and Li X (2023) Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *arXiv preprint arXiv:2305.18459* .

He K, Fan H, Wu Y, Xie S and Girshick R (2020) Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9729–9738.

Ho J, Jain A and Abbeel P (2020a) Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33: 6840–6851.

Ho J, Jain A and Abbeel P (2020b) Denoising diffusion probabilistic models. In: *Advances in Neural Information Processing Systems*, volume 33. pp. 6840–6851.

Intelligence P, Black K, Brown N, Darpinian J, Dhabalia K, Driess D, Esmail A, Equi M, Finn C, Fusai N, Galliker MY, Ghosh D, Groom L, Hausman K, Ichter B, Jakubczak S, Jones T, Ke L, LeBlanc D, Levine S, Li-Bell A, Mothukuri M, Nair S, Pertsch K, Ren AZ, Shi LX, Smith L, Springenberg JT, Stachowicz K, Tanner J, Vuong Q, Walke H, Walling A, Wang H, Yu L and Zhilinsky U (2025) $\pi_{0.5}$: a vision-language-action model with open-world generalization. URL https://arxiv.org/abs/2504.16054.

Janner M, Du Y, Tenenbaum JB and Levine S (2022) Planning with diffusion for flexible behavior synthesis. In: *International Conference on Machine Learning*. pp. 9902–9915.

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

Kalashnikov D, Irpan A, Pastor P, Ibarz J et al. (2018) Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In: *Conference on Robot Learning (CoRL)*.

Kang B, Jie X, Du D, Li J, Chen X, Zhang Y and Tian Y (2023) Efficient diffusion policies for offline reinforcement learning. *arXiv preprint arXiv:2305.20081* .

Kostrikov I, Nair A and Levine S (2022) Offline reinforcement learning with implicit q-learning. In: *ICLR*.

Kumar A, Zhou A, Tucker G and Levine S (2020) Conservative q-learning for offline reinforcement learning. In: *Advances in Neural Information Processing Systems*, volume 33. pp. 1179–1191.

Lei K, He Z, Lu C, Hu K, Gao Y and Xu H (2024) Uni-o4: Unifying online and offline deep reinforcement learning with multi-step on-policy optimization. In: *ICLR*.

Levine S, Finn C, Darrell T and Abbeel P (2016) End-to-end training of deep visuomotor policies. In: *Robotics: Science and Systems (RSS)*.

Li H, Jiang Z, CHEN Y and Zhao D (2024) Generalizing consistency policy to visual RL with prioritized proximal experience regularization. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. URL https://openreview.net/forum?id=MOFwt8OeXr.

Lin T, Sachdev K, Fan L, Malik J and Zhu Y (2025) Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396* .

Lipman Y, Chen RT, Ben-Hamu H, Nickel M and Le M (2023) Flow matching for generative modeling. In: *International Conference on Learning Representations*.

Liu Y, Bao J and Li KH (2024) Pinchbot: Long-horizon deformable manipulation with guided diffusion policy. *arXiv preprint arXiv:2507.17846* .

Lu C, Chen H, Chen J, Su H, Li C and Zhu J (2023) Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. *arXiv preprint arXiv:2304.08349* .

Lu Y, Tian Y, Yuan Z, Wang X, Hua P, Xue Z and Xu H (2025) H3dp: Triply-hierarchical diffusion policy for visuomotor learning. *arXiv preprint arXiv:2505.07819* .

Luo J, Hu Z, Xu C, Tan YL, Berg J, Sharma A, Schaal S, Finn C, Gupta A and Levine S (2024a) Serl: A software suite for sample-efficient robotic reinforcement learning. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 16961–16969. DOI:10.1109/ICRA57147.2024.10610040.

Luo J, Hu Z, Xu C, You S, Darrell T and Levine S (2024b) SERL: A software suite for sample-efficient robotic reinforcement learning. In: *IEEE International Conference on Robotics and Automation*.

Luo J, Xu C, Liu F, Trevor D and Levine S (2024c) Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics* 9(96): eads5033.

Luo J, Xu C, Wu J and Levine S (2025) Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics* 10(105): eads5033. DOI:10.1126/scirobotics.ads5033. URL https://www.science.org/doi/abs/10.1126/scirobotics.ads5033.

Majumdar A, Yadav K, Arnaud S, Ma YJ, Chen C, Silwal S, Jain A, Berges VP, Abbeel P, Malik J et al. (2023) Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240* .

Nair A, Dalal M, Gupta A and Levine S (2020) AWAC: Accelerating online reinforcement learning with offline datasets. In: *International Conference on Machine Learning*.

Nair S, Rajeswaran A, Kumar V, Finn C and Gupta A (2022) R3m: A universal visual representation for robot manipulation. In: *Conference on Robot Learning*.

Nakamoto M, Zhai Y, Singh A, Mark MS, Ma Y, Finn C, Kumar A and Levine S (2023) Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In: *Advances in Neural Information Processing Systems*.

Nguyen T and Yoo CD (2025) Revisiting diffusion q-learning: From iterative denoising to one-step action generation. *arXiv preprint arXiv:2508.13904* .

Park S, Kang K and Levine S (2025) Flow q-learning. *arXiv preprint arXiv:2502.02538* .

Peng XB, Kumar A, Zhang G and Levine S (2019) Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. In: *International Conference on Learning Representations*.

Psenka M, Escontrela A, Abbeel P and Ma Y (2024) Learning a diffusion model policy from rewards via q-score matching. *arXiv preprint arXiv:2312.11752* .

Rajeswaran A, Kumar V, Gupta A, Vezzani G, Schulman J, Todorov E and Levine S (2018) Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In: *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania. DOI:10.15607/RSS.2018.XIV.049.

Ren AZ, Dixit A, Bodrova AS, Singh S, Tu S, Brown N, Xu P, Takayama L, Xia F, Varley J, Xu Z, Sadigh D, Zeng A and Majumdar A (2024) Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588* .

Sabatelli M, Suresh S and Zhang B (2024) Diffclone: Enhanced behaviour cloning in robotics with diffusion-driven policy learning. *arXiv preprint arXiv:2401.09243* .

Schulman J, Moritz P, Levine S, Jordan MI and Abbeel P (2016) High-dimensional continuous control using generalized advantage estimation. In: *ICLR*.

Schulman J, Wolski F, Dhariwal P, Radford A and Klimov O (2017) Proximal Policy Optimization Algorithms. *ArXiv preprint* abs/1707.06347.

Shang S, Zheng Y, Yin Z and Duan Y (2024) Latent weight diffusion: Generating policies from trajectories. *arXiv preprint arXiv:2410.14040* .

Singh A, Yang L, Hartikainen K, Finn C and Levine S (2019) End-to-end robotic reinforcement learning without reward engineering. In: *Robotics: Science and Systems*.

Song J, Meng C and Ermon S (2021) Denoising diffusion implicit models. In: *International Conference on Learning Representations*. URL https://openreview.net/forum?id=St1giarCHLP.

Song Y, Dhariwal P, Chen M and Sutskever I (2023) Consistency models. In: *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S and Poole B (2020) Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* .

Todorov E, Erez T and Tassa Y (2012) Mujoco: A physics engine for model-based control. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 5026–5033. DOI:10.1109/IROS.2012.6386109.

Wagenmaker A, Beeson A, Ajay A and Gupta A (2025) Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2501.xxxxx* .

Walke H, Black K, Lee A, Kim MJ, Du M, Zheng C, Zhao T, Hansen-Estruch P, Xu Q, Dragan A et al. (2023) BridgeData V2: A dataset for robot learning at scale. *arXiv preprint arXiv:2308.12952* .

Wang SY, Kubota Y and Shah D (2023a) Diffusion model-augmented behavioral cloning. *arXiv preprint arXiv:2302.13335* .

Wang Z, Hunt JJ and Zhou M (2023b) Diffusion policies as an expressive policy class for offline reinforcement learning. In: *International Conference on Learning Representations*.

Wang Z, Zhu Y, Liu S and Song S (2024) One-step diffusion policy: Fast visuomotor policies via diffusion distillation. *arXiv preprint arXiv:2410.21257* .

Yu T, Quillen D, He Z, Julian R, Hausman K, Finn C and Levine S (2020) Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: *Conference on robot learning*. PMLR, pp. 1094–1100.

Yuan Z, Wei T, Gu L, Hua P, Liang T, Chen Y and Xu H (2025) Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation. URL https://arxiv.org/abs/2508.20085.

Ze Y, Zhang G, Zhang K, Hu C, Wang M and Xu H (2024) 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In: *Proceedings of Robotics: Science and Systems*. Delft, Netherlands. DOI:10.15607/RSS.2024.XX.067.

Zhang T, Yu C, Su S and Wang Y (2025) Reinflow: Fine-tuning flow matching policy with online reinforcement learning. *arXiv preprint arXiv:2505.22094* .

Zhao TZ, Kumar V, Levine S and Finn C (2023) Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In: *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea. DOI:10.15607/RSS.2023.XIX.016.

Appendix

Experiment Setting

This section presents the setup and methodology for conducting real-world robot manipulation experiments. The experimental framework is designed to integrate perception, control, and demonstration collection in a coherent pipeline. We employ the Intel RealSense L515 camera to capture depth images, which are subsequently processed into 3D point clouds and transformed into the robot's root frame for consistent spatial representation. Intrinsic and extrinsic calibrations ensure accurate mapping between image coordinates and the robot workspace, while spatial filtering and downsampling produce compact yet informative point cloud observations suitable for high-frequency control.

The robotic platform consists of three manipulators—UR5, xArm, and Franka Emika Panda—equipped with dexterous or gripper-type end-effectors, as well as a passive fixed effector. Asynchronous control schemes are implemented across all manipulators to achieve low-latency, high-frequency execution, with task-specific interpolation strategies ensuring smooth and responsive motion.

Demonstrations for seven manipulation tasks are collected using teleoperation interfaces matched to task complexity. For dexterous 3D tasks, the Apple Vision Pro provides natural hand tracking, which is converted into robot joint or Cartesian commands via inverse kinematics for the LeapHand. For planar or low-dimensional tasks, a standard game joystick is used to provide delta-action control of the end-effector in the plane. Each task includes approximately 100 demonstration trajectories, efficiently recorded within a few hours per task.

Overall, the experimental setup provides a consistent, flexible, and practical framework for acquiring high-quality real-world data and evaluating robot manipulation policies across a diverse set of tasks.

Camera Calibration In the real-world experiments, we employ the Intel RealSense L515 camera, chosen for its ability to provide high-fidelity 3D point cloud observations owing to its superior depth accuracy. This ensures reliable perception for robot manipulation tasks across diverse environments.

The intrinsic calibration is performed using a Charuco board, which serves as the reference pattern for estimating the camera's internal parameters. Multiple images of the board are captured from varying angles and distances. The board corners are then detected and used to compute the intrinsic matrix and distortion coefficients through an optimization process that minimizes the reprojection error. This procedure guarantees an accurate mapping between 2D image coordinates and their corresponding 3D spatial points.

The extrinsic calibration aligns the camera frame with the robot's root frame using an AprilTag placed on the tabletop. First, a manual measurement establishes the transformation T_{tag2root}, which encodes the pose of the tag relative to the robot base. Subsequently, the AprilTag is detected in the camera's RGB channel, yielding the transformation $T_{\text{tag2camera}}$ from the tag to the camera frame. The camera-to-root transformation is then obtained as:

$$T_{\text{camera2root}} = T_{\text{tag2root}} \cdot (T_{\text{tag2camera}})^{-1}$$

This calibrated extrinsic transformation is stored and utilized throughout the experiments to maintain consistent spatial alignment between the camera observations and the robot workspace.

Point Cloud Processing The raw depth maps captured by the RealSense L515 are first back-projected into 3D point clouds using the calibrated intrinsic parameters, thereby establishing a metric reconstruction of the scene in the camera frame. These point clouds are then transformed into the robot's root frame using the calibrated extrinsic transformation $T_{\text{camera2root}}$, ensuring spatial consistency between visual observations and the robot's operational workspace.

To suppress environmental noise and spurious reflections, a spatial bounding box is applied in the root frame, filtering out irrelevant regions while preserving only the volume above the tabletop. This step enhances the semantic

relevance of the retained points, focusing on task-relevant structures.

Finally, to improve computational efficiency while maintaining geometric fidelity, farthest point sampling (FPS) is applied. The processed point cloud is uniformly downsampled to 512 points, yielding a compact yet informative representation suitable for robot perception and policy learning. During deployment, the camera operates in a dedicated thread with a buffered pipeline, enabling asynchronous acquisition of observations. This design minimizes latency and allows the effective point cloud update rate to exceed 25 Hz, ensuring timely and reliable perception for closed-loop control.

Robotic Arm & End-effector Control The experimental platform incorporates three robotic manipulators—UR5, xArm, and Franka Emika Panda—equipped with different types of end-effectors, including the LeapHand dexterous hand, the Robotiq 2F-85 gripper, and a custom 3D-printed fixed effector. All control modules are designed to operate asynchronously, ensuring low-latency communication and high-frequency execution for responsive and stable manipulation.

For the UR5, relative action commands are transmitted at 30 Hz. These commands are handled through the Real-Time Data Exchange (RTDE) interface, where they are interpolated to 125 Hz for execution. A lookahead time of $1/30$ s and a control gain of 200 are used, providing a balance between smooth motion trajectories and timely responsiveness.

For the xArm, control is performed via absolute Cartesian position commands at 30 Hz. Linear interpolation is applied to the position trajectories, while orientation is represented in quaternion form and interpolated via spherical linear interpolation (slerp). The resulting trajectory is executed at 200 Hz, yielding continuous and smooth motion in task space.

For the Franka Emika Panda, absolute Cartesian position commands are also issued at 30 Hz. The robot's low-level controller performs trajectory interpolation, converting the input into control signals at 1000 Hz, thereby ensuring precise and compliant execution.

Regarding the end-effector control, both the LeapHand and the Robotiq 2F-85 are driven asynchronously at 30 Hz. The policy performs end-to-end joint control for the LeapHand, while the gripper uses binary control for the Soft-towel Folding task and continuous control for the Orange Juicing task. The custom 3D-printed fixed effector, by contrast, is purely passive and requires no active control, serving as a simple but reliable interface for specific experimental settings like Dynamic Push-T or Agile Bowling.

Data Collection

Data for seven real-world manipulation tasks are collected using teleoperation interfaces selected to match the complexity and dimensionality of each task. These tasks include both dexterous three-dimensional manipulations—such as folding, pouring, unscrewing, and juicing—and planar or constrained motions, such as Dynamic Push-T and Agile Bowling. The data collection strategy aims to provide sufficient coverage of the task space while maintaining practical efficiency in demonstration acquisition.

For tasks involving complex 3D hand motions, the Apple Vision Pro is employed. Its built-in spatial hand-tracking captures natural hand and finger movements, which are mapped to robot end-effector motions. Before starting each demonstration, the operator's hand pose is aligned with the robot end-effector through an initialization procedure, ensuring a consistent reference frame. During teleoperation, wrist and finger motions are streamed in real time and converted into robot joint or Cartesian commands, while gripper grasping actions are triggered via pinch gestures. For the LeapHand dexterous hand, inverse kinematics is solved in PyBullet to produce feasible joint configurations from the tracked hand poses. This setup allows operators to efficiently provide demonstrations for tasks requiring dexterous, spatially rich manipulations.

For planar or low-dimensional manipulation tasks—such as Dynamic Push-T and Agile Bowling—a standard game joystick is used. The joystick's XY-axis values are mapped to delta actions of the robot end-effector in the plane. This approach provides an intuitive and efficient means for recording demonstrations where precise two-dimensional control is sufficient.

Our dataset is intentionally varied. Some tasks feature a large number of demonstrations to test performance limits, while others use smaller datasets to evaluate sample efficiency. Tab. 7 provides a complete summary.