2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Robust Online Residual Refinement via Koopman-Guided Dynamics Modeling

Zhefei Gong[1], Shangke Lyu[2†], Pengxiang Ding[13], Wei Xiao[1], Donglin Wang[1]

***Abstract*—Imitation learning (IL) enables efficient skill acquisition from demonstrations but often struggles with long-horizon tasks and high-precision control due to compounding errors. Residual policy learning offers a promising, model-agnostic solution by refining a base policy through closed-loop corrections. However, existing approaches primarily focus on local corrections to the base policy, lacking a global understanding of state evolution, which limits robustness and generalization to unseen scenarios. To address this, we propose incorporating global dynamics modeling to guide residual policy updates. Specifically, we leverage Koopman operator theory to impose linear time-invariant structure in a learned latent space, enabling reliable state transitions and improved extrapolation for long-horizon prediction and unseen environments. We introduce KORR (Koopman-guided Online Residual Refinement), a simple yet effective framework that conditions residual corrections on Koopman-predicted latent states, enabling globally informed and stable action refinement. We evaluate KORR on long-horizon, fine-grained robotic furniture assembly tasks under various perturbations. Results demonstrate consistent gains in performance, robustness, and generalization over strong baselines. Our findings further highlight the potential of Koopman-based modeling to bridge modern learning methods with classical control theory.**

I. Introduction

Imitation learning (IL) offers an efficient framework for acquiring task-specific policies [1–3]. Recent progress in leveraging large-scale foundation models [4, 5] has further extended IL's potential toward generalizable robotic behaviors. However, pure IL methods remain brittle in long-horizon, high-precision tasks like furniture assembly—common in daily life yet inherently challenging [6]. In such settings, even minor deviations from expert demonstrations can accumulate, leading to catastrophic failure—a problem known as compounding error. For instance, misaligning a single peg during assembly may invalidate the entire construction. Similarly, fine-grained manipulation tasks require highly accurate control, leaving little tolerance for errors. Although methods like action chunking [3] can improve consistency and fluency, they introduce long decision latencies, compromising open-loop responsiveness and increasing sensitivity to environmental perturbations.

To address these challenges, residual policy learning [7–9] offers a compelling solution. By augmenting a pre-trained base policy with corrective actions through online reinforcement learning, it provides a model-agnostic, plug-and-play refinement mechanism applicable across diverse tasks, and has demonstrated strong empirical performance. Yet existing residual policies are typically limited to local corrections around the base policy's output, restricting their ability to address the core challenge of robotics—robustness and generalization—an aspect largely overlooked in residual learning, whereas recent IL methods scale for generalization. Conventional strategies [10] sample corrections only near base actions, resulting in limited global awareness and poor extrapolation to novel or unseen situations. Consequently, when base actions deviate substantially due to model uncertainty, residual policies often fail to recover, regardless of the base policy's quality.

We argue that achieving robust residual learning necessitates a principled modeling of global dynamics to support effective extrapolation. In particular, modeling dynamics in a time-invariant manner introduces an essential structural prior that promotes policy stability during online residual updates. Koopman operator theory [11] provides a compelling framework by lifting complex nonlinear dynamics into a linear latent space. In this lifted space, inherently nonlinear and coupled dynamics are represented as finite-dimensional, decoupled, and time-invariant linear transitions, which enables more reliable and globally consistent modeling of motion dynamics [12]. It also alleviates the exponential instabilities often encountered in nonlinear systems, facilitating more stable online training.

Building on these insights, we introduce **KORR** (**K**oopman-guided **O**nline **R**esidual **R**efinement), a novel and effective framework for residual policy learning. KORR first models system dynamics by lifting states into a linear latent space and propagating them with learned Koopman dynamics. During correction, the base policy generates an action, which KORR uses to extrapolate the next imagined state. The residual policy then conditions on this state to produce a corrective action, as shown in Fig. 1. This decoupled structure enables the residual policy to leverage global information for more stable and robust refinement. By combining Koopman modeling with residual learning, KORR bridges modern learning techniques with classical control theory, offering new opportunities for robust optimization in complex tasks. We evaluate KORR on challenging, long-horizon, and fine-grained robotic tasks under various perturbations, including external disturbances and randomized initial conditions. Experimental results show that KORR outperforms strong baselines in terms of performance, robustness, and generalization. Further comparisons with standard nonlinear dynamics models demonstrate the advantages of Koopman-guided modeling. Comprehensive ablation studies offer additional insights into the design choices behind Koopman operator learning and future study.

In summary, our contributions are as follows:

[1]MiLAB, Westlake University [2]Nanjing University [3]Zhejiang University
†Corresponding author: shangke_lyu@nju.edu.cn
[§]Project page: https://zhefeigong.github.io/korr-robot/
This work was supported by the National Science and Technology Innovation 2030-Major Projects (Grant No. 2022ZD0208800) and the Natural Science Foundation of China general program (Grant No. 62573362).

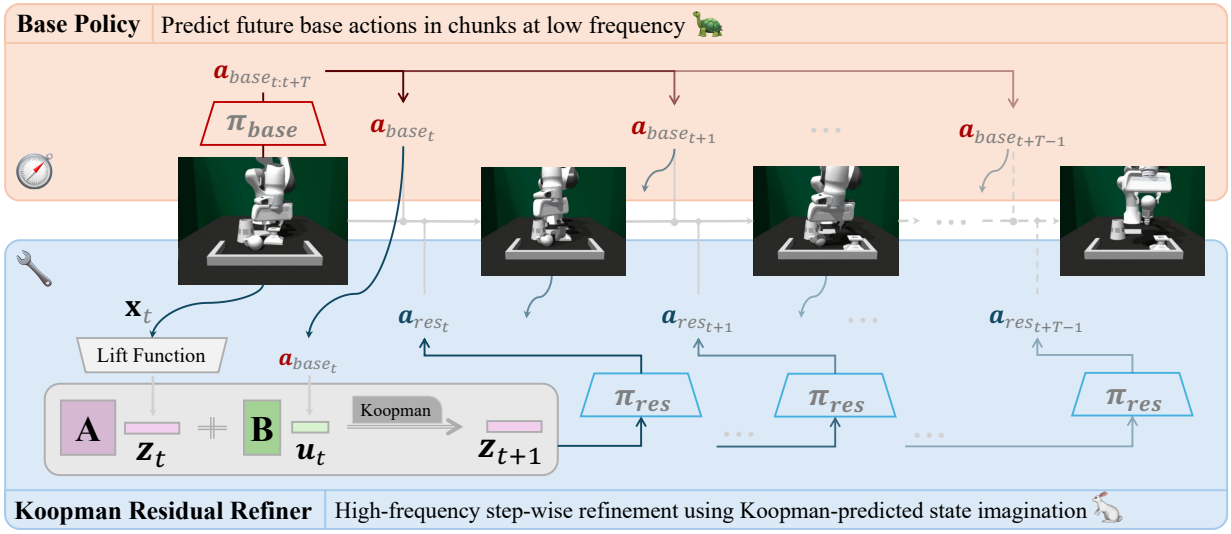


Fig. 1: **Overview of KORR**. The base policy predicts a chunk of base actions at a lower frequency, while KORR refines these actions step-by-step at a higher control rate. For each base action a_{base_t}, KORR extrapolates the next possible state $\mathbf{z}_{t+1}$ using Koopman dynamics as an imagined future state, and then conditions the residual policy on this state to generate a corrective residual action a_{res_t}. The final executed action, obtained by combining the base and residual actions, forms a closed-loop refinement that enhances robustness and generalization.

- Highlight limitations of existing residual policy learning, particularly its reliance on local corrections and lack of global awareness, which limits robustness and generalization.
- Introduce KORR, a Koopman-guided residual refinement framework to enhance robustness and generalization via the extrapolation capabilities of linear time-invariant dynamics.
- Conduct extensive experiments and ablation studies to evaluate the effectiveness of our dynamics modeling approach, offering insights for future research.

II. Related Work

Koopman Operator in Robotics: The Koopman operator [11] offers a linear framework to analyze nonlinear dynamics by lifting them into a high-dimensional *observable* space. Efficient approximations such as Dynamic Mode Decomposition (DMD) [13] and Extended DMD (EDMD) [14] have enabled practical estimation from time-series data. More recently, data-driven encoding approaches have been introduced [15]. Koopman theory has been integrated into classical control paradigms, such as LQR [16, 17] and MPC [18], and later enhanced by deep learning techniques [19]. For example, an LQR structure can be embedded into a learned Koopman representation [20]. In robotics, Koopman-based modeling enables long-horizon control in interactive settings [12], and has been applied to pixel-based tasks using reinforcement learning and contrastive learning [21, 22], though primarily in simplified environments. Recent work extends Koopman to dexterous manipulation in both state and pixel spaces [23, 24], but these tasks are typically short-horizon and designed for specific setups. Koopman has also been leveraged in imitation learning to improve sample efficiency via its linear constraints [8]. Overall, Koopman theory has emerged as a promising modeling paradigm, facilitating learning and control across diverse robotic domains [25].

Residual Learning in Robotics: Residual learning has been widely adopted in deep learning to address vanishing gradients [26] and to enable efficient fine-tuning via delta updates [27]. In robotics, residual reinforcement learning (Residual RL) has shown promise in refining predefined controllers, particularly in challenging industrial settings [28]. Residual policies have also demonstrated improved responsiveness and performance in deformable object manipulation tasks [29]. Model-agnostic residual structures are increasingly used to refine base policies, including vision-language-action agents [4, 30]. Transic [31] trains residual policies via supervised learning from human preferences for sim-to-real transfer, though this requires extensive human involvement. More recent works apply online RL to train residual policies in sparse reward settings [9, 32], showing improved task performance. Despite these advances, residual policies are typically constrained to operate within a narrow correction range around the base policy. While this design accelerates learning and convergence, it limits exploration and confines optimization to a local neighborhood—impeding the capture of global features and thereby restricting robustness and generalization.

III. Preliminary

Koopman Operator Theory provides a linear representation of nonlinear dynamical systems by lifting the original state space into an infinite-dimensional Hilbert space [11]. Consider a discrete-time autonomous nonlinear system:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) \tag{1}$$

where $\mathbf{x}_t \in X \subset \mathbb{R}^d$ is the system state at time t, and $f : \mathbb{R}^d \to \mathbb{R}^d$ denotes a nonlinear transition function. To enable linear analysis, the Koopman framework introduces a lift function $g : X \to O$ that maps states into a higher-dimensional space of *observables*. The system dynamics in the lifted space evolve

linearly under the action of the Koopman operator K:

$$K \circ g(\mathbf{x}_t) = g(\mathbf{x}_{t+1}) = g \circ f(\mathbf{x}_t) \tag{2}$$

where K is a linear operator acting on the *observables*.

Koopman for Control Systems requires one more control input $\mathbf{u}_t$ in the nonlinear discrete controlled system formulation as $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$. A common approach is to lift only the state $\mathbf{x}_t$ into the higher-dimensional *observable* space, while leaving the control input $\mathbf{u}_t$ unchanged. Under this setting [33, 34], the system can be represented linearly as:

$$g(\mathbf{x}_t, \mathbf{u}_t) = \begin{bmatrix} g(\mathbf{x}_t) \\ \mathbf{u}_t \end{bmatrix}^\top ; \; K \circ g(\mathbf{x}_t, \mathbf{u}_t) = g(\mathbf{x}_{t+1}) \tag{3}$$

Operating in an infinite-dimensional space is impractical for real-world applications. Thus, Koopman analysis typically relies on finite-dimensional approximations of the operator. Specifically, the Koopman operator K can be approximated by a finite matrix $K \in \mathbb{R}^{p \times p}$ as follows:

$$K = \begin{bmatrix} A & B \\ \cdot & \cdot \end{bmatrix} \tag{4}$$

where $A \in \mathbb{R}^{m \times m}$ represents the evolution of the lifted state, and $B \in \mathbb{R}^{m \times n}$ models the effect of the control inputs on the state evolution. Here, $p = m + n$, and the lower block (denoted as $(\cdot)$) represents the evolution associated with the control input $\mathbf{u}_t$, which is not explicitly modeled in this work. The resulting Koopman-based discrete dynamics can then be expressed as:

$$K \cdot g(\mathbf{x}_t, \mathbf{u}_t) = A \cdot g(\mathbf{x}_t) + B \cdot \mathbf{u}_t \approx g(\mathbf{x}_{t+1}) \tag{5}$$

Koopman Operator Approximation aims to identify the optimal Koopman operator and lift function. The choice of g is typically made using heuristics (e.g., polynomial lifting or kernel methods) or a neural network learned from data. To avoid tedious parameter tuning, we adopt the neural network approach. Specifically, given a dataset $D = \{\mathbf{x}_k, \mathbf{u}_k\}_{k=0}^{M}$ with M total data pairs, the Koopman operator K can be obtained by minimizing the following objective function:

$$K = \arg\min_K \frac{1}{2} \sum_{k=0}^{M-1} ||g(\mathbf{x}_{k+1}) - K \cdot g(\mathbf{x}_k, \mathbf{u}_k)||^2 \tag{6}$$

where the optimization can be performed via least squares or data-driven approaches, such as minimizing the MSE loss using stochastic gradient descent.

IV. Method

To address the limited robustness and generalization of recent conventional residual policies—and to leverage the long-horizon modeling capabilities of Koopman operator theory—we propose **KORR** (**K**oopman-guided **O**nline **R**esidual **R**efinement), a model-agnostic framework that augments any base policy with globally informed, reliable corrections. KORR benefits from the interpretability and extrapolation properties of Koopman-based linear dynamics, as illustrated in Fig. 1. Building on our problem formulation in Sec. IV-A, we develop KORR through two key components: (1) modeling system dynamics as linear time-invariant transitions in a latent space via Koopman theory to capture structured, global transitions (Sec. IV-B); and (2) guiding residual policy refinement using Koopman-predicted future latent states to enable stable and globally coherent corrections (Sec. IV-C).

A. Problem Formulation

We formulate the task as a sequential decision-making problem: given a base policy that may suffer from imprecision or perturbations, the objective is to learn corrective residual actions to enhance execution. We assume the base policy uses action chunking, a strategy shown to improve performance in recent work [1–3]. Notably, KORR makes no structural assumptions about the base policy, allowing it to operate with any offline-pretrained model. At each timestep t, given the environment state $\mathbf{x}_t$ (or observation $\mathbf{o}_t$) and the base action $\mathbf{a}_{\text{base}_t}$, KORR predicts a residual correction $\mathbf{a}_{\text{res}_t}$. Training is conducted via online reinforcement learning under sparse rewards, highlighting practical applicability [9, 32]. The action space includes end-effector poses in $SE(3)$ and a binary gripper state, while the state space comprises the robot configuration (pose and velocity) and object poses.

B. Koopman-Guided Dynamics Modeling

KORR adopts a simple yet effective strategy by modeling forward dynamics as linear time-invariant evolution in a lifted latent space, where structured embeddings capture complex nonlinear behaviors. This design allows the residual policy to more accurately model state transitions, system structure, and residual corrections, leading to more effective refinement. Specifically, we leverage Deep Koopman embeddings to lift system states into a latent space, while assuming that a linear control term (without lifting the input) suffices for effective modeling [16, 35]. We construct a neural network g_θ that maps the nonlinear state $\mathbf{x}_t \in \mathbb{X}$ into a linear Koopman *observable* space $\mathbf{z}_t \in \mathbb{O}$, as shown in Eq. (3). (Alternatively, polynomial lifting up to a finite degree can also be employed.)

The Koopman operator, represented by a matrix $\mathbf{K}$, captures the linear evolution of system dynamics in the lifted latent space. Following Eq. (5), we decompose $\mathbf{K}$ into two matrices, $\mathbf{A}$ and $\mathbf{B}$, corresponding to the contributions from the current state $\mathbf{z}_t$ and the control input $\mathbf{a}_t$, respectively.

$$\mathbf{A} \cdot \mathbf{z}_t + \mathbf{B} \cdot \mathbf{a}_t = \mathbf{z}_{t+1} \tag{7}$$

To learn the Koopman operators $\mathbf{A}$, $\mathbf{B}$, and the lift function parameters θ, we optimize them via gradient backpropagation to minimize the model prediction loss $\mathscr{L}$ (see Eq. (6)). Specifically, given a dataset $\mathscr{D} = (\mathbf{x}_0, \mathbf{a}_0), \ldots, (\mathbf{x}_M, \mathbf{a}_M)$ of state-action trajectories, where $\mathbf{a}$ (equivalently $\mathbf{a}_{\text{exe}}$) denotes the action executed in the environment, the objective is to minimize the MSE loss as follows:

$$\mathscr{L}_{\text{kpm}} = \mathbb{E}_{t \sim \mathscr{D}} ||\mathbf{z}_{t+1} - (\mathbf{A} \cdot \mathbf{z}_t + \mathbf{B} \cdot \mathbf{a}_t)||^2 \; ; \; \mathbf{z}_{t+1} = g_\theta(\mathbf{x}_{t+1}) \tag{8}$$

Note that $\mathbf{x}_{t+1}$ represents the next state following $\mathbf{x}_t$ in the same trajectory. This dynamical system is time-invariant, which allows it to naturally capture manipulation skills that are

more robust to intermittent perturbations compared to systems that explicitly depend on time [36].

C. Residual Policy through Koopman Imagination

Given the state $\mathbf{x}_t$ or observation $\mathbf{o}_t$ from the environment, the base policy π_{base} generates a base action $\mathbf{a}_{\text{base}_t}$. Unlike traditional residual policies that generate the residual action directly based on the state and base action, we integrate an additional layer of imagination through Koopman dynamics to enhance both robustness and interpretability. Using Koopman dynamics, we first project the next imagined state $\mathbf{z}_{t+1}^{\text{base}}$ based on executing $\mathbf{a}_{\text{base}_t}$. Then, the residual policy π_{res} conditions on this imagined state to generate the residual action $\mathbf{a}_{\text{res}_t}$(see Eq. (9)). Finally, the executable action $\mathbf{a}_{\text{exe}_t}$ is computed by summing the base action $\mathbf{a}_{\text{base}_t}$ and the residual action $\mathbf{a}_{\text{res}_t}$.

$$\mathbf{A} \cdot g_\theta(\mathbf{x}_t) + \mathbf{B} \cdot \mathbf{a}_{\text{base}_t} = \mathbf{z}_{t+1}^{\text{base}} \;;\; \mathbf{a}_{\text{res}_t} = \pi_{\text{res}}(\mathbf{z}_{t+1}^{\text{base}}) \tag{9}$$

We train the efficient residual Multi-Layer Perceptron (MLP) policy π_{res} with the base policy frozen, using online RL to fine-tune and address issues of imprecision, slow reactivity, and limited robustness and generalization. We apply PPO [37]. With these straightforward updates (highlighted in light-blue in Alg. 1), we leverage the robustness of Koopman dynamics' extrapolation for a holistic system view. Additionally, by conditioning on the predicted next state rather than the base action and current state, we decouple the residual policy's understanding, mirroring human decision-making, which often relies on anticipated outcomes over direct sensory inputs [38].

Algorithm 1 KORR pseudo-code with PPO

Initialize base policy π_{base} (pretrained and frozen here), residual policy π_{res} with parameters ϕ.
Initialize Koopman Operator $\mathbf{A}$ and $\mathbf{B}$, and lift function $g(\cdot)$ with parameters θ.
for iteration = 1 to N **do**
  for each environment rollout step **do**
    Observe current state $\mathbf{x}_t$
    Compute base action: $\mathbf{a}_{\text{base}_t} = \pi_{\text{base}}(\mathbf{x}_t)$
    Koopman extrapolate: $\mathbf{A} \cdot g_\theta(\mathbf{x}_t) + \mathbf{B} \cdot \mathbf{a}_{\text{base}_t} = \mathbf{z}_{t+1}^{\text{base}}$ ▷ See description in Eq. (9)
    Compute residual action based the imaginary next state: $\mathbf{a}_{\text{res}_t} = \pi_{\text{res}}(\mathbf{z}_{t+1}^{\text{base}})$
    Execute $\mathbf{a}_{\text{exe}_t} = \mathbf{a}_{\text{base}_t} + \mathbf{a}_{\text{res}_t}$, observe reward r, next state $\mathbf{x}_{t+1}$
    Store $(\mathbf{x}_t, \mathbf{a}_{\text{exe}_t}, r, \mathbf{x}_{t+1})$ into buffer $\mathscr{D}$
  end for
  Compute advantage estimates A using GAE
  for update step **do**
    Compute Koopman loss $\mathscr{L}_{\text{kpm}}$ to update θ, $\mathbf{A}$, and $\mathbf{B}$ ▷ MSE loss depicted in Eq. (8)
    Compute PPO loss $\mathscr{L}_{\text{ppo}}$ to update all of the parameters including the ϕ
  end for
end for

V. Experiments

With the above relatively simple modifications, our method, **KORR**, demonstrates stronger robustness, generalization, and higher performance compared to conventional residual policies under online reinforcement learning. In this section, we present experimental results designed to address the following research questions (**RQ**s):

- **RQ1**: Does KORR improve robustness and performance over traditional residual policies?
- **RQ2**: Does Koopman modeling offer advantages over non-linear dynamics for residual learning?
- **RQ3**: Which design choices in KORR are critical for stable performance?

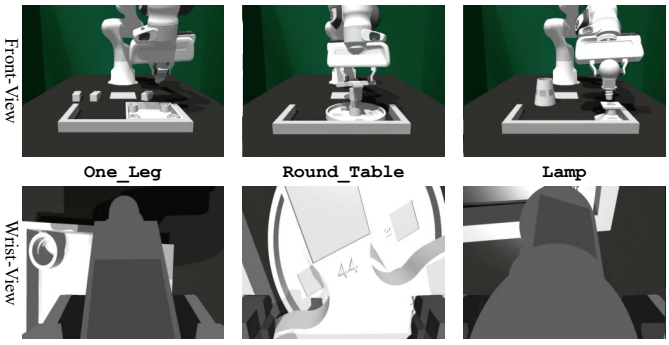


Fig. 2: **Task visualizations.** Snapshots of the IKEA *Furniture Assembly* tasks used for evaluation throughout our experiments.

A. Experimental Design

Benchmarks and Baselines: We evaluate KORR on the challenging 6-Degree-of-Freedom (DoF) long-horizon *Furniture Assembly* benchmark [6], built with IsaacGym [39]. This benchmark has been widely studied in recent works [31, 32, 40, 41] for its real-world complexity and fine-grained manipulation skills. For base policies, we adopt the official implementations of (1) *Diffusion Policy (DP)* [2], which learns via denoising diffusion probabilistic models [42], and (2) *CARP* [1], which employs coarse-to-fine autoregressive prediction. To evaluate residual policy designs, we compare KORR against: (1) *ResiP* [32], a basic MLP-based residual policy; and (2) *ResiP + Non-Linear Dynamics*, a variant where residuals are predicted via learned non-linear dynamics (MLP). Residual policies are separately trained for each base policy–task pair.

Tasks and Metrics: We evaluate on three challenging tasks from the IKEA Furniture Assembly benchmark [6]: `One_Leg`, `Round_Table`, and `Lamp` (visualized in Fig. 2). Each task requires executing long-horizon dexterous skills such as picking, placing, inserting, screwing, and flipping, over episodes lasting up to 1000 steps (`One_Leg` is capped at 700 steps). To systematically assess robustness and generalization, we introduce (1) *initial randomness* in object placements, categorized into `Low`, `Med`, and `High` levels, and (2) *external disturbances*, where objects are perturbed after each action (`w/o` vs. `w/` disturbance). These settings simulate real-world uncertainties. Policies are evaluated over 1024 episodes, and success rates are reported based on the best-performing ckpt.

B. General Robustness and Performance Study

To thoroughly evaluate residual policy performance, we introduce two levels of initialization randomness—`Low` and

Methods	One_Leg						Round_Table				Lamp			
	Low		Med		High		Low		Med		Low		Med	
DP [2]	47.97		23.93		4.39		5.76		1.56		5.57		1.46	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
ResiP [32]	98.14	85.35	84.99	46.09	30.03	3.20	96.19	80.27	52.73	22.66	86.58	60.55	51.66	29.98
KORR (ours)	98.73	**90.38**	**87.21**	**52.31**	**44.04**	**8.30**	96.78	81.35	**67.19**	**27.25**	**89.65**	**63.48**	**74.47**	**39.16**

Tab. I: **Performance across tasks and difficulty levels.** We report the success rate (%) based on 1024 rollouts for each task. The residual policies (ResiP and KORR) are built upon the same base policy (DP), with the entire training process conducted without disturbances. Here, w/o denotes evaluation without disturbances, while w/ refers to evaluation with the corresponding level of disturbance. By seamlessly incorporating simple Koopman dynamics, KORR consistently surpasses conventional residual policies in both task performance and robustness.

Med—with an additional High level used exclusively for the One_Leg task. Each level is associated with a disturbance scale, where scene components are randomly perturbed after each interaction with the environment to simulate the robustness required in real-world tasks. All residual policies are fine-tuned via reinforcement learning on a fixed, disturbance-free base policy, with disturbances applied only during the evaluation phase to fully test the model's robustness. Here, we denote evaluation without disturbances as w/o, and with level-specific disturbances as w/.

As shown in Tab. I, KORR not only consistently improves success rates under normal conditions but also exhibits remarkable robustness across all tasks and randomness levels when disturbances are applied. The impact of Koopman dynamics plays a key role in this improvement. By modeling the global dynamics, KORR enables the policy to generalize effectively, even under unseen conditions. This extrapolation capability is especially advantageous when disturbances create novel situations. These results strongly support **RQ1**, demonstrating that KORR achieves substantial performance and robustness improvements with minimal changes to the implementation, as illustrated in Alg. 1.

Methods	One_Leg		Round_Table		Lamp	
	→Med	↓Δ	→Med	↓Δ	→Med	↓Δ
ResiP [32]	16.41	83.28	7.32	92.39	17.28	80.04
KORR(ours)	**22.27**	77.44	**10.54**	89.11	**18.75**	79.09

Tab. II: **Generalization performance.** Direct evaluation of Low-trained policies under Med initialization randomness.

We further assess the generalization capabilities of KORR enabled by Koopman dynamics. In this experiment, residual policies trained under Low initialization randomness are directly deployed at the Med level, evaluating their ability to adapt to previously unseen scenarios, while the base policy remains frozen throughout. As shown in Tab. II, KORR consistently achieves stronger generalization, exhibiting higher success rates at Med (measured by → Med) and smaller relative performance drops (measured by ↓ Δ, computed as $\frac{\Delta x}{x_{\text{original}}} \times 100\%$), compared to ResiP. These results further validate KORR's robust extrapolation to unseen scenarios, thus strongly supporting **RQ1**.

We further evaluate the universality of KORR by applying

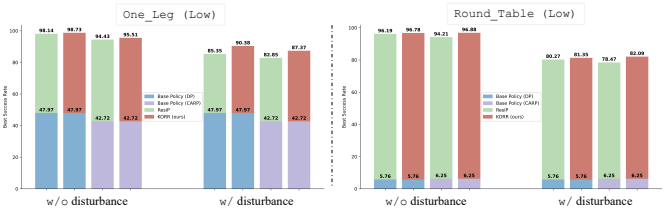


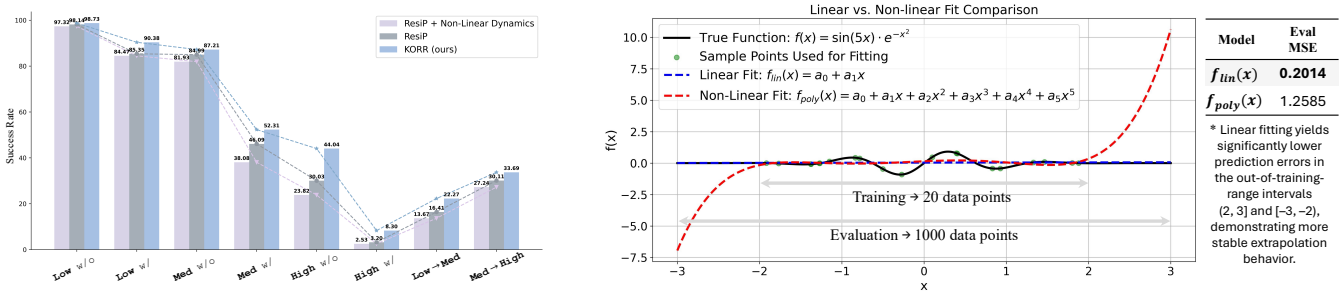
Fig. 3: **Universality across base policies.** KORR consistently enhances performance and robustness with an alternative base policy.

it to different base policies, retrained for each scenario. In Fig. 3, KORR, when applied to an autoregressive base policy, CARP [1], consistently outperforms legacy residual methods, achieving significant improvements in both success rate and robustness. These findings provide further support for **RQ1** and demonstrate KORR's flexibility in integrating with various base policies. Additionally, we observe that the initial quality of the base policy influences the effectiveness of residual learning, as detailed in Fig. 6. The reported success rates of the base policies reflect their frozen performance under standard, disturbance-free conditions.

C. Linear Benefit Study

Compared to traditional residual policies [32], KORR incorporates a simple yet effective linear dynamics model during online RL fine-tuning. In contrast, conventional designs typically favor non-linear dynamics, often parameterized by MLPs. To better isolate the benefit of linear structure, we introduce an additional baseline, *ResiP + Non-Linear Dynamics* , where a non-linear transition dynamics model predicts future states conditioned on the current state and base action, and the residual policy is trained accordingly on these imagined outcomes as well.

As shown in Fig. 4(a), we evaluate performance across all levels of randomness in the One_Leg task, under both w/o and w/ disturbance conditions, as well as in generalization settings (Low/Med training to Med/High evaluation). The results show that KORR consistently outperforms the ResiP baseline, whereas the addition of non-linear dynamics leads to performance degradation. Compared to non-linear extrapolation, KORR enforces a linear time-invariant structure through Koopman modeling, encouraging the learning of globally consistent dynamics. This inductive bias improves long-horizon extrapolation, mitigates overfitting to spurious correla-



(a) Success rate in `One_Leg` task.

(b) Extrapolation behavior of linear vs. non-linear models.

Fig. 4: **Comparison of non-linear and Koopman-guided dynamics.** (a) shows that non-linear dynamics hurt performance, while Koopman-based dynamics consistently improve robustness and generalization. (b) shows a simple analytical experiment on polynomials demonstrates that linear models extrapolate conservatively and stably, while non-linear models suffer from large errors outside the training range.

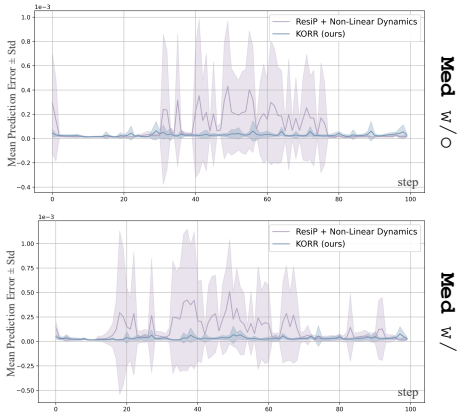


Fig. 5: **Dynamics prediction error.** Linear Koopman constraints improve stability, producing more consistent and accurate long-term extrapolations, measured by MSE (mean ± std) over 10 rollouts.

Task	ResiP	NL (2-layer)	NL (4-layer)	KORR (Ours)
`Low`	98.14	97.32	97.47	**98.73** (↑)
`Med`	84.99	81.93	80.76	**87.21** (↑)

Tab. III: **Evaluation with stronger non-linear dynamics.** Results on the `One_Leg` task without disturbances using DP [2] as the base. Deeper non-linear (NL) models fail to improve and may even reduce performance, whereas Koopman dynamics consistently enhances results, echoing recent findings on linear modeling [12, 43].

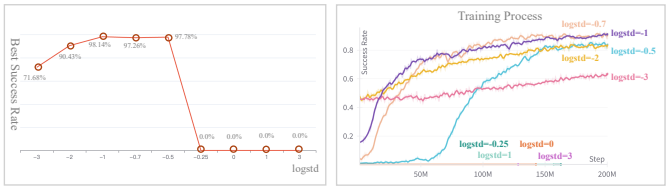


Fig. 8: **Ablation on exploration rate.** Properly balancing exploration is crucial: excessive exploration destabilizes early learning, while insufficient exploration limits policy improvement. A moderate exploration rate enables stable and efficient residual policy training.

tions, and reduces cumulative prediction errors—ultimately enhancing robustness and generalization, thus supporting **RQ2**. In Fig. 5, KORR achieves significantly lower prediction drift over time, whereas non-linear dynamics often exhibit instability and oscillation. This linear prior provides stable inductive regularization and helps avoid the divergence commonly observed in unconstrained non-linear models.

To provide intuitive insights into these findings, we combine empirical comparisons with deeper non-linear architectures and theoretical analyses with formal proofs of their extrapolation behavior. As reported in Tab. III, non-linear dynamics modeled by MLPs show no improvement and even cause a notable performance drop, despite using deeper, more expressive architectures. This is consistent with the findings in recent work on the benefits brought from linear modeling [12, 43].

One explanation lies in extrapolation. When fitting complex non-linear targets with limited and range-restricted data, linear models often generalize better due to their conservative extrapolation behavior [43]. This advantage becomes more pronounced in long-horizon settings where errors accumulate over time. As shown in Fig. 4(b), we fit both linear and non-linear models to a complex polynomial target function using limited training data. Outside the training range (e.g., $(2,3]$), the non-linear model exhibits large prediction errors that worsen with increased extrapolation distance. In contrast, the linear model remains stable and achieves lower overall MSE, reflecting its robustness and superior generalization due to the conservative nature of linear extrapolation.

To further support the empirical results, we compare the extrapolation behavior of a linear model $\hat{f}_{\text{lin}}(x) = a_0 + a_1 x$ and a non-linear polynomial $\hat{f}_{\text{poly}}(x) = \sum_{i=0}^{5} a_i x^i$ over $x \in [x_0, x_0 + \delta]$, where x_0 is at the edge of the training region (with limited dataset). The linear model has a constant derivative and is globally Lipschitz:

$$|\hat{f}_{\text{lin}}(x) - \hat{f}_{\text{lin}}(x_0)| = |a_1| \cdot |x - x_0|$$

Extrapolation grows linearly, predictably bounded by $|x - x_0|$:

$$|f(x) - \hat{f}_{\text{lin}}(x)| \leq |f(x) - f(x_0)| + |f(x_0) - \hat{f}_{\text{lin}}(x_0)| + |a_1| \cdot |x - x_0|$$

Taylor's theorem bounds the non-linear polynomial error as:

$$|f(x) - \hat{f}_{\text{poly}}(x)| \leq \frac{|f^{(d+1)}(\xi)|}{(d+1)!} |x - x_0|^{d+1} \quad \text{for some } \xi \in [x_0, x]$$

Methods	One_Leg						Round_Table	
	Low		Med		High		Low	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/
KORR	**98.73**	**90.38**	**87.21**	**52.31**	**44.04**	**8.30**	**96.78**	**81.35**
KORR w/o bkp (↓)	95.99	83.59	73.73	35.16	19.24	2.83	64.94	28.91

Tab. IV: **Effect of disabling RL loss backpropagation to Koopman dynamics.** In KORR, RL losses backpropagate into the Koopman dynamics module alongside a standalone prediction loss (see Alg. 1). We ablate this by blocking RL gradients and evaluating One_Leg and Round_Table tasks across randomization levels, with and without disturbances. Removing RL feedback severely degrades performance in all settings, indicating that minimizing the Koopman prediction loss alone is insufficient; task rewards are necessary to guide the lifted representation in capturing reward-relevant features.

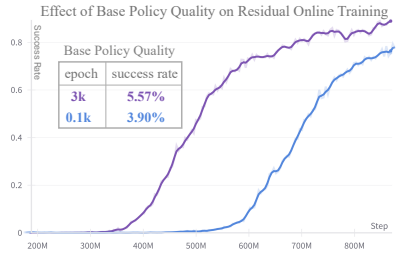


Fig. 6: **Effect of base policy quality.** The same residual policy is trained on base diffusion policies of varying training durations. Better-trained base policies accelerate residual learning and improve final performance.

Methods	One_Leg				Lamp			
	Low		Med		Low		Med	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/
ResiP	**98.14**	**85.35**	**84.99**	**46.09**	**86.58**	**60.55**	**51.66**	**29.98**
ResiP + Goal (↓)	96.78	81.15	82.62	39.55	44.53	26.27	48.54	24.68
KORR	98.73	**90.38**	87.21	52.31	89.65	63.48	74.47	39.16
KORR + Goal (↑)	**99.02**	90.27	**87.60**	**53.81**	**90.02**	**65.57**	**75.10**	**40.04**

Tab. V: **Effect of goal-conditioning on residual policies.** We incorporate goal-conditioning into the residual policy by providing the final desired state, constructed from the last frame of a successful rollout, as an additional input. Goal information slightly improves KORR but degrades ResiP, highlighting KORR's potential advantage in goal-conditioned tasks.

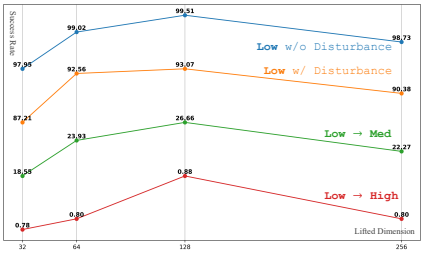


Fig. 7: **Effect of lifted dimension.** A 128-d lifted space performs best, while larger dimensions add unnecessary complexity, highlighting the need to match the lifted dimension to task complexity.

The extrapolation error scales as $\mathcal{O}(|x-x_0|^{d+1})$, leading to rapid error amplification even with moderate d and small extrapolation beyond the training region. This highlights the instability of high-degree polynomials in extrapolation, whereas linear models provide more stable and conservative predictions, mitigating error amplification in long-horizon prediction.

D. Additional Ablation Study

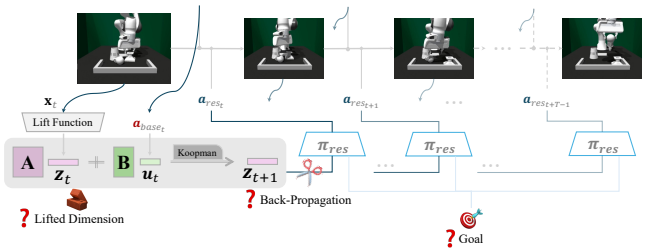


Fig. 9: **Visualization of key ablation studies in KORR.** Highlighted 'question' emojis indicate the specific ablation topics discussed below, such as lifted dimension and others.

To further investigate the design choices of KORR and address **RQ3**, we conduct a series of further studies focusing on the use of Koopman-guided dynamics. We analyze the impact of the exploration rate during residual policy training (Fig. 8) and examine how the quality of the base policy influences the learning of residuals (Fig. 6). Regarding the Koopman dynamics itself, we perform extensive ablation studies (Fig. 9), including evaluations of the lifted space dimension (Fig. 7), the importance of backpropagating the RL loss into the Koopman module (Tab. IV), and the feasibility of incorporating goal-conditioning (Tab. V). These comprehensive ablations offer strong guidance for the future application and refinement of the Koopman-guided framework.

VI. Conclusion

In this work, we presented KORR, a Koopman-guided residual refinement framework that enhances robustness and generalization by modeling global dynamics through linear time-invariant structures induced by Koopman operator theory in a lifted latent space. By constraining residual learning within this linearized space, KORR improves stability and mitigates the sensitivity of non-linear dynamics to disturbances. Comprehensive experiments and ablation studies further validate the design choices and offer guidance for future extensions. Moreover, by leveraging Koopman-based linearization, KORR naturally connects learning-based approaches with classical control techniques, such as LQR [19] and ESO [44], which are inherently effective in linear regimes. KORR represents a promising step toward integrating control-theoretic insights into learning frameworks, paving the way for more robust and principled learning in future practical robotics applications.

References

[1] Z. Gong, P. Ding, S. Lyu, S. Huang, M. Sun, W. Zhao, Z. Fan, and D. Wang, "Carp: Visuomotor policy learning via coarse-to-fine autoregressive prediction," *arXiv preprint arXiv:2412.06782*, 2024. [Online]. Available: https://arxiv.org/abs/2412.06782

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

[3] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

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

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

[6] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," *The International Journal of Robotics Research*, p. 02783649241304789, 2023.

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

[8] J. Bi, K. Lim, K. Chen, Y. Huang, and H. Soh, "Imitation learning with limited actions via diffusion planners and deep koopman controllers," *arXiv preprint arXiv:2410.07584*, 2024.

[9] X. Yuan, T. Mu, S. Tao, Y. Fang, M. Zhang, and H. Su, "Policy decorator: Model-agnostic online refinement for large policy model," *arXiv preprint arXiv:2412.13630*, 2024.

[10] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6023–6029.

[11] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[12] A. K. Mondal, S. S. Panigrahi, S. Rajeswar, K. Siddiqi, and S. Ravanbakhsh, "Efficient dynamics modeling in interactive environments with koopman theory," *arXiv preprint arXiv:2306.11941*, 2023.

[13] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[14] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[15] J. Ng and H. H. Asada, "Data-driven encoding: A new numerical method for computation of the koopman operator," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3940–3947, 2023.

[16] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, p. e0150171, 2016.

[17] S. Lyu, X. Lang, and D. Wang, "Koopman-based robust learning control with extended state observer," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2303–2310, 2025.

[18] I. Abraham, G. De La Torre, and T. D. Murphey, "Model-based control using koopman operators," *arXiv preprint arXiv:1709.01568*, 2017.

[19] H. Shi and M. Q.-H. Meng, "Deep koopman operator with control for nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7700–7707, 2022.

[20] H. Yin, M. C. Welle, and D. Kragic, "Embedding koopman optimal control in robot policy learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 392–13 399.

[21] X. Lyu, H. Hu, S. Siriya, Y. Pu, and M. Chen, "Task-oriented koopman-based control with contrastive encoder," in *Conference on Robot Learning*. PMLR, 2023, pp. 93–105.

[22] H. Kumawat, B. Chakraborty, and S. Mukhopadhyay, "Robokoop: Efficient control conditioned representations from visual input in robotics using koopman operator," *arXiv preprint arXiv:2409.03107*, 2024.

[23] Y. Han, M. Xie, Y. Zhao, and H. Ravichandar, "On the utility of koopman operator theory in learning dexterous manipulation skills," in *Conference on Robot Learning*. PMLR, 2023, pp. 106–126.

[24] H. Chen, A. Abuduweili, A. Agrawal, Y. Han, H. Ravichandar, C. Liu, and J. Ichnowski, "Korol: Learning visualizable object feature with koopman operator rollout for manipulation," *arXiv preprint arXiv:2407.00548*, 2024.

[25] L. Shi, M. Haseli, G. Mamakoukas, D. Bruder, I. Abraham, T. Murphey, J. Cortes, and K. Karydis, "Koopman operators in robot learning," *arXiv preprint arXiv:2408.04200*, 2024.

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

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[28] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5548–5555.

[29] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 389–404, 2024.

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

[31] Y. Jiang, C. Wang, R. Zhang, J. Wu, and L. Fei-Fei, "Transic: Sim-to-real policy transfer by learning from online correction," *arXiv preprint arXiv:2405.10315*, 2024.

[32] L. Ankile, A. Simeonov, I. Shenfeld, M. Torne, and P. Agrawal, "From imitation to refinement–residual rl for precise assembly," *arXiv preprint arXiv:2407.16677*, 2024.

[33] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[34] E. Caldarelli, A. Chatalic, A. Colomé, C. Molinari, C. Ocampo-Martinez, C. Torras, and L. Rosasco, "Linear quadratic control of nonlinear systems with koopman operator learning and the nyström method," *Automatica*, vol. 177, p. 112302, 2025.

[35] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the koopman operator and model predictive control," *arXiv preprint arXiv:1902.02827*, 2019.

[36] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, no. 1, pp. 297–330, 2020.

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

[38] C. Summerfield and F. P. De Lange, "Expectation in perceptual decision making: neural and computational mechanisms," *Nature Reviews Neuroscience*, vol. 15, no. 11, pp. 745–756, 2014.

[39] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[40] L. Ankile, A. Simeonov, I. Shenfeld, and P. Agrawal, "Juicer: Data-efficient imitation learning for robotic assembly," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 5096–5103.

[41] H. Lin, R. Corcodel, and D. Zhao, "Generalize by touching: Tactile ensemble skill transfer for robotic furniture assembly," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9227–9233.

[42] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[43] S. Fujimoto, P. D'Oro, A. Zhang, Y. Tian, and M. Rabbat, "Towards general-purpose model-free reinforcement learning," *arXiv preprint arXiv:2501.16142*, 2025.

[44] J. Han, "From pid to active disturbance rejection control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.