Learning Object Manipulation from Scratch via Contrastive Interaction

Tongle Shen[1], **Caleb Chuck**[2,†], **Fan Feng**[1,†], **Biwei Huang**[1,†]

Keywords: Object-Centric Manipulation, Contrastive RL, Unsupervised Learning

Summary

Contrastive Reinforcement Learning (CRL) has seen recent success in a wide variety of goal-conditioned robotics tasks by learning structured representations of the dynamics. However, despite its success in locomotion and similar domains, CRL often struggles in interaction-rich manipulation. We argue that a key source of this difficulty is object-centric *interaction*, such as contact or grasping, that induces distinct changes in the underlying dynamic modes. In this work, we prove that manipulation can only be approximated by piecewise affine transformation and show that interaction-induced mode changes create piecewise nonlinear reachability structures that are difficult for standard CRL energy functions to represent and plan over. Based on this analysis, we introduce Interaction-weighted Resampling (IWR). IWR performs interaction-aware resampling around phases before, during, and after interactions, encouraging the learned representations to better capture the dynamic complexity around interactions. Across interaction-centric environments, including 2D dynamic control, robotic manipulation, and robot air hockey, IWR improves both sample efficiency and overall performance over prior CRL methods, with around 19.8% average improvement in simulation. Finally, using a sim-to-real pipeline with policies trained by IWR, we demonstrate the first real-world goal-conditioned robot air hockey agent capable of hitting goals, improving success rate from 25% to 60%.

Contribution(s)

1. This paper provides a principled analysis of why interaction-rich manipulation poses a challenge for contrastive reinforcement learning, showing that interaction-induced mode changes create piecewise nonlinear reachability structures that are difficult to represent and plan over.
   Context: Prior work [22] has analyzed contrastive RL through smooth temporal interpolation and Gaussian-like representation geometry; our analysis focuses on how this distribution breaks down in manipulation due to object interactions fundamentally.
2. This paper introduces Interaction-Weighted Resampling (IWR), a sampling-based method that shifts contrastive updates toward interaction-relevant transitions while preserving the standard CRL objective.
   Context: Prior CRL methods use uniform or trajectory-context resampling [88]; IWR uses interaction structure to target sparse mode-changing transitions that are critical for manipulation.
3. This paper provides theoretical guarantees for the proposed interaction weighting scheme and empirically validates it across dynamic maze domains, robotic manipulation tasks, and robot air hockey.
   Context: The theoretical results characterize how interaction-centered samples improve learning of interaction-relevant representations; the empirical evaluation demonstrates improved sample efficiency and performance in interaction-centric environments.

Learning Object Manipulation from Scratch via Contrastive Interaction

Tongle Shen[1], **Caleb Chuck**[2,†], **Fan Feng**[1,†], **Biwei Huang**[1,†]
{t7shen, f2feng, bih007}@ucsd.edu; calebc@cs.utexas.edu

[1]**University of California, San Diego** [2]**University of Texas, Austin**

[†] Equal advising

Abstract

Contrastive Reinforcement Learning (CRL) has seen recent success in a wide variety of goal-conditioned robotics tasks by learning structured representations of the dynamics. However, despite its success in locomotion and similar domains, CRL often struggles in interaction-rich manipulation. We argue that a key source of this difficulty is object-centric *interaction*, such as contact or grasping, that induces distinct changes in the underlying dynamic modes. In this work, we formulate manipulation dynamics as a piecewise-smooth Markov process and show that interaction-induced mode changes create piecewise nonlinear reachability structures that are difficult for standard CRL energy functions to represent and plan over. Based on this analysis, we introduce Interaction-weighted Resampling (IWR). IWR performs interaction-aware resampling around phases before, during, and after interactions, encouraging the learned representations to better capture the dynamic complexity around interactions. Across interaction-centric environments, including 2D dynamic control, robotic manipulation, and robot air hockey, IWR improves both sample efficiency and overall performance over prior CRL methods, with around 19.8% average improvement in simulation. Finally, using a sim-to-real pipeline with policies trained by IWR, we demonstrate the first real-world goal-conditioned robot air hockey agent capable of hitting goals, improving success rate from 25% to 60%.

1 Introduction

Object manipulation remains a core challenge for robot learning, but the factors that make it difficult to learn generalist manipulation policies remain unclear. Most existing approaches rely heavily on either expert demonstrations [75, 89, 43, 9, 58] or carefully designed reward functions [52, 72]. While these paradigms have impressive progress, they also introduce strong supervision requirements: demonstrations can be expensive to collect and may cover only a narrow range of behaviors, while task-specific rewards often require substantial engineering and do not easily scale to open-ended multitask settings. These limitations motivate unsupervised approaches that can acquire reusable manipulation skills and representations directly from interaction, without assuming demonstrations or manually specified rewards [19, 74, 45, 59, 2]. Among them, goal-conditioned Reinforcement Learning (GCRL) offers a promising direction for learning a family of policies to handle the inherent multitask nature of manipulation [51]. Successful GCRL requires learning strong representations to capture the shared properties between different control behaviors. While the strategies for learning these representations remain an open challenge, recent work in Contrastive Reinforcement Learning (CRL) offers a promising direction forward [21]. CRL has shown impressive capabilities to learn complex control from scratch, and even limited success in manipulation tasks [50]. However, despite the clear fit of GCRL algorithms for manipulation, their performance remains limited.

In this work, we ask a central question: *when and why does CRL fail for manipulation*, and *what is needed to make it work*? We identify *interactions* as a core property of manipulation that makes effective policy representation difficult. Specifically, interactions are relational events between entities that induce distinct changes in dynamic modes. Since CRL relies on the learned representation to define goal-reaching energies for planning, these interaction-induced mode changes can substantially distort the geometry over which policies are learned and planned.

To formalize this challenge, we view manipulation as a mode-switching dynamical system, where the active mode changes across phases before, during, and after an interaction. These interaction-dependent modes induce piecewise nonlinear dynamics and reachability structures that are substantially harder to represent than the smoother structures often observed in locomotion tasks, which can be well approximated by linear-Gaussian dynamics [22]. This provides a concrete mechanism for why CRL representations can become poorly shaped in manipulation: they may smooth over critical interaction boundaries and fail to capture the mode changes that determine future reachability. We empirically observe this failure mode and show how it leads to poor policy performance in Fig. 1.

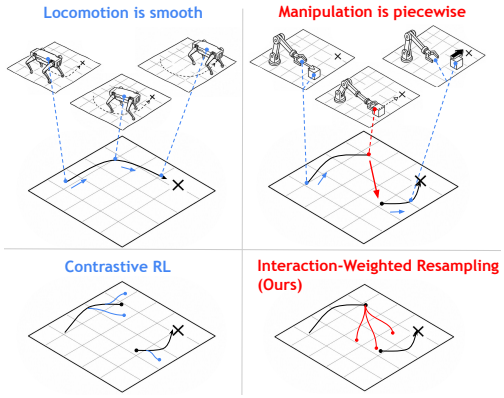


Figure 1: **Motivation**. In locomotion-like domains, future reachability often follows a smooth temporal structure, making standard CRL effective. In manipulation, object interactions induce piecewise mode changes, causing standard CRL to miss critical interaction transitions. IWR emphasizes these interaction-relevant points to better learn the goal-reaching energy.

Guided by our analysis of how interactions induce piecewise nonlinear structure in manipulation, we propose a principled yet simple scheme to improve CRL in these settings. Building on insights from interaction modeling [12, 37, 15, 47] and combinatorial resampling [42, 88], we introduce Interaction-weighted Resampling (IWR), a contrastive resampling principle that reshapes the training distribution toward interaction-induced mode transitions. Instead of treating all transitions uniformly, IWR constructs informative state-goal comparisons around interaction-relevant points, encouraging the representation to preserve the mode boundaries that govern future reachability. Through evaluation on robotic manipulation domains with complex object interactions, including Meta-World, simulated dynamic maze environments, and robot air hockey, we show that IWR provides a principled way to address the core challenges of CRL by improving both its training distribution and representation capacity. IWR achieves significantly better sample efficiency and overall performance than prior GCRL methods. Notably, we demonstrate the first goal-conditioned robot air hockey agent capable of hitting multiple goals from a dynamically moving puck.

Our contributions are: ***(i)*** a principled analysis of why interaction-rich manipulation poses a fundamental challenge for CRL ***(ii)*** Introducing Interaction-weighted Resampling (IWR), inspired by this principled analysis, which reshapes the effective contrastive training distribution toward interaction-relevant transitions. ***(iii)*** Empirical validation of IWR across dynamic maze domains, robotic manipulation, and substantial improvement in robot air hockey.

2 Related Work

Our work lies at the intersection of CRL, unsupervised goal-conditioned control, and robot manipulation with interactions. An extended discussion on these works is given in Appendix B.

Contrastive Reinforcement Learning (CRL) applies contrastive learning [57] to goal-conditioned reinforcement learning (GCRL), through both classification [20] and regression objectives [21]. Recent work has extended CRL toward metric learning [84, 56], planning [22], language align-

ment [54, 55], and combinatorial reasoning and search [77, 88]. Interaction-weighted Resampling is most closely related to Ziarko et al. [88], which also uses resampling. However, IWR targets resampling using interaction structure to focus on difficult-to-model dynamics, whereas Ziarko et al. [88] uses different contexts for in-trajectory sampling. We also build on recent successes of CRL in simulated manipulation [76], including emergent exploration [50, 5], which enables limited but sample-efficient manipulation performance compared to prior GCRL methods [63, 38, 51, 3, 4]. Beyond CRL, IWR draws on broader contrastive learning ideas, including the use of privileged information such as labels or structured metadata [24, 17, 41, 31], and debiasing strategies that carefully select negatives [10, 36, 18]. Our view of the tension between non-smooth dynamics and smooth contrastive representations is partly inspired by Betser et al. [6], which observes that learned representations tend toward a thin-shell Gaussian. Furthermore, the use of factorization and entity interactions in contrastive learning remains limited, which is the focus of IWR.

Manipulation Learning with Interactions Robot manipulation covers tasks such as grasping, moving, and throwing objects [29]. Interactions [14] have been used in manipulation learning through controllability [68], data augmentation [61, 62], hierarchical RL [11, 12, 79], skill learning [33, 65, 32], causal modeling [46, 7], and exploration [78]. In contrast, our work formalizes manipulation difficulty as interaction-induced mode-changing dynamics and introduces IWR as an interaction-aware algorithm for training CRL. IWR improves sample efficiency and, to our knowledge, provides the first CRL application to robot air hockey [13].

3 Background

In this section, we set up the basic modeling framework. Specifically, we model manipulation as a Markov Decision Process with factored structures that describe interactions, and then learn goal-conditioned policies using CRL within the GCRL framework.

Factorized Markov Decision Process We consider a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with a factored state space [27]. Specifically, the state is decomposed into n components, $s = (s^1, s^2, \ldots, s^n) \in \mathcal{S}, \quad \mathcal{S} = \mathcal{S}^1 \times \mathcal{S}^2 \times \cdots \times \mathcal{S}^n$. Each factor s^i represents the state of an entity, such as the robot, an object, and its properties. This factorization gives a natural way to describe manipulation domains with multiple interacting entities.

Goal-Conditioned Reinforcement Learning In Goal-Conditioned Reinforcement Learning (GCRL) [38], an agent learns the policy that can solve a family of tasks specified by different goals. Let $\mathcal{S}$ denote the state space, $\mathcal{A}$ the action space, and $\mathcal{G}$ the goal space. At the beginning of each episode, a goal $g \in \mathcal{G}$ is sampled from a goal distribution $p(g)$. The agent then acts according to a goal-conditioned policy $\pi(a \mid s, g)$, which maps the current state and desired goal to a distribution over actions. Given transition dynamics $p(s_{t+1} \mid s_t, a_t)$ and a goal-conditioned reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$, the objective is to maximize the expected discounted return over both trajectories and goals: $J(\pi) = \mathbb{E}_{g \sim p(g),\, \tau \sim p_\pi(\tau|g)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, g) \right]$, where $\tau = (s_0, a_0, \ldots, s_T)$ is a trajectory induced by $\pi(\cdot \mid s, g)$ and $\gamma \in [0, 1)$ is the discount factor.

Contrastive Reinforcement Learning We consider goals as future states or achieved goal states. For a state-action pair (s, a), we define the discounted future occupancy under policy π as

$$\rho^\pi(g \mid s, a) = (1 - \gamma) \sum_{k=1}^{\infty} \gamma^{k-1} p^\pi(s_{t+k} = g \mid s_t = s, a_t = a).$$

This essentially measures how likely the policy is to reach goal g in the future after taking action a at state s. Thus, learning a goal-conditioned policy can be viewed as learning a representation of state-goal reachability.

Given this goal-conditioned setting, contrastive reinforcement learning (CRL) [21] learns a goal-reaching energy that measures how likely a future goal g is reachable from a state-action pair (s, a). CRL can be viewed as estimating the density ratio between this conditional future occupancy

$\rho^\pi(g \mid s, a)$ and a replay-marginal future distribution $\bar{\rho}^\pi_{\mathcal{B}}(g)$:

$$E^\star(s, a, g) = \log \rho^\pi(g \mid s, a) - \log \bar{\rho}^\pi_{\mathcal{B}}(g). \tag{1}$$

In practice, CRL parameterizes this energy with an inner product between a state-action encoder and a goal encoder [21, 50]. Let $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and $\psi : \mathcal{S} \to \mathbb{R}^d$. The learned energy is

$$E_{\phi,\psi}(s, a, g) = \phi(s, a)^\top \psi(g). \tag{2}$$

For each state-action pair $(s, a) \sim d_{\mathcal{B}}(s, a)$, CRL samples one positive future $g^{(1)} \sim \rho^\pi(\cdot \mid s, a)$ and $N - 1$ negative futures $g^{(2)}, \ldots, g^{(N)} \sim \bar{\rho}^\pi_{\mathcal{B}}(\cdot)$. The critic is trained with an InfoNCE-style objective, together with a LogSumExp regularizer and a constant hyperparameter β_{lse}:

$$\max_{\phi,\psi} \mathbb{E}\left[\log \frac{\exp(\phi(s, a)^\top \psi(g^{(1)}))}{\sum_{j=1}^N \exp(\phi(s, a)^\top \psi(g^{(j)}))} - \beta_{\text{lse}} \left(\log \sum_{j=1}^N \exp(\phi(s, a)^\top \psi(g^{(j)})) \right)^2 \right], \tag{3}$$

where the expectation is over $(s, a) \sim d_{\mathcal{B}}(s, a)$, $g^{(1)} \sim \rho^\pi(\cdot \mid s, a)$, and $g^{(2)}, \ldots, g^{(N)} \sim \bar{\rho}^\pi_{\mathcal{B}}(\cdot)$. The learned energy is then used as an implicit goal-reaching reward for actor learning.

4 Analysis: Why does CRL Struggle with Interactions in Manipulation?

The success of CRL depends on whether the inner-product energy $E_{\phi,\psi}(s, a, g)$ can faithfully represent the future-density ratio $E^\star(s, a, g)$. This motivates our analysis: what class of future distributions is naturally represented by the inner-product CRL energy function, and when does this representation become insufficient? Building on the Gaussian idealization behind temporal interpolation in CRL [22], we first revisit why contrastive representations are well-suited to smooth, single-mode dynamics, and then show why this structure breaks down in manipulation, where interactions induce latent mode changes and piecewise nonlinear future distributions.

4.1 CRL as a Linear-Gaussian Temporal Model

We begin from the Gaussian idealization of temporal contrastive representations. Eysenbach et al. [22] show that CRL induces a simple linear-Gaussian model of discounted future occupancy in representation space. In particular, when the marginal representation distribution is approximately isotropic Gaussian and the contrastive density ratio is represented by a quadratic score, the conditional future representation admits a Gaussian form,

$$p(\psi_{t+n} \mid \psi_t) = \mathcal{N}(\mu_n(\psi_t), \Sigma_n).$$

Naturally, for a single-mode MDP, the temporal mean can be parameterized by a single linear operator A, yielding

$$\mu_n(\psi_t) = A^n \psi_t, \qquad p(\psi_{t+n} \mid \psi_t) = \mathcal{N}(A^n \psi_t, \Sigma_n). \tag{4}$$

All temporal structure is generated by powers of the same operator A. The geometrically sampled future states used by CRL provide dense supervision for estimating this temporal operator across different horizons, which explains the strength of CRL in navigation and locomotion domains. Essentially, CRL can be understood as learning a smooth Gaussian-like geometry for future reachability.

4.2 Manipulation Induces Mode-Conditioned Gaussian Futures

However, the single-operator Gaussian view becomes insufficient in manipulation. Unlike locomotion domains, where the action persistently affects the same state variable used as the future goal, manipulation often separates the actuated factor from the target factor. Let the state be factored as $s_t = (u_t, y_t)$, where u_t denotes the actuated factor, such as the robot end-effector or paddle, and y_t denotes the target factor, such as an object or puck, and the state-action anchor is $x_t = (u_t, y_t, a_t)$.

In the average case, actions do not always affect the target factor. Instead, their effects are activated only through interaction events. We represent this using an interaction indicator $\omega_t = m(x_t) \in \{0, 1\}$, where $\omega_t = 0$ denotes *passive* target dynamics and $\omega_t = 1$ denotes an interaction mode. The target dynamics can then be written as a piecewise transition:

$$y_{t+1} = \begin{cases} f_0(y_t), & \omega_t = 0, \\ f_1(u_t, y_t, a_t), & \omega_t = 1. \end{cases} \tag{5}$$

In the passive mode, the target dynamics are without direct dependence on the robot action; in the interaction mode, the target dynamics depend on the actuated factor and action.

Applying the Gaussian temporal view mode-wise yields a different representation-level future model for each mode. Let $\psi_t = \psi(y_t)$ be the target-factor representation. In the *passive mode*, the future representation can be approximated by a target-only temporal operator, $\psi_{t+1} = A_0\psi_t + \epsilon_t$. In the *interaction mode*, however, the target update includes an additional displacement induced by the actuated factor and action: $\psi_{t+1} = A_1\psi_t + b_t + \epsilon_t$, $b_t = B(u_t, y_t, a_t)$.

Combining the two cases, manipulation induces a switched affine Gaussian model:

$$\psi_{t+1} = A_{\omega_t}\psi_t + \omega_t b_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, \Sigma_{\omega_t}). \tag{6}$$

This indicates the *structural mismatch* between manipulation and the single-mode CRL geometry. Interactions change the active mode of the representation dynamics and introduce an interaction residual b_t that depends on the robot-object relation and action. As a result, the future representation is no longer governed by powers of a single operator A, but by a sequence of mode-dependent operators and interaction residuals. An illustration is in Fig. A1.

4.3 Two Challenges for CRL in Interaction-Rich Manipulation

The mode-conditioned Gaussian reveals two distinct challenges for applying CRL to manipulation.

Sampling Challenge: Standard CRL learns from endpoint pairs (x_t, y_f), where y_f is sampled as a discounted future. However, the interaction event that changes the target dynamics may occur only at a sparse and latent time τ_t. If the sampled future lies before the interaction, the tuple only observes passive target evolution and provides no direct supervision for the interaction-induced shift. If the sampled future lies far after the interaction, the effect of the interaction is propagated through subsequent passive dynamics, which can attenuate or mix the signal. As a result, ordinary contrastive sampling can be dominated by smooth pre-interaction or post-interaction transitions, while the critical bridge transition that changes the dynamic mode remains underrepresented.

Error Propagation: Even when an interaction-crossing future is sampled, the interaction information may be observed only after it has propagated through subsequent target dynamics. This makes the learned energy fragile around contact. We formalize this issue in the one-reset case, where the target first evolves passively, then experiences one interaction, and then evolves passively again.

Proposition 1 (Error Propagation)**.** *Let e denote the local interaction-bridge estimation error, and let k denote the number of passive steps after the interaction. In the one-reset case, the endpoint energy error scales as*

$$\sup |\widehat{E}_k - E_k| \propto \|A_0^k\|\|e\| + \frac{1}{2}\|A_0^k\|^2\|e\|^2. \tag{7}$$

Proposition 1 shows that the endpoint energy error depends on two factors: the local bridge estimation error $\|e\|$ and the post-interaction propagation factor $\|A_0^k\|$. In manipulation, after contact the target often evolves passively and cannot be corrected by the robot. Therefore, an inaccurate interaction shift is carried forward by the passive dynamics, producing fragile energy estimates and inaccurate actor gradients near the interaction. In tasks with repeated or branched interactions, such as pick-place, this issue becomes more severe because the future distribution becomes a mixture over multiple interaction times and interaction residuals. More detailed theorems, analysis, and the full proof can be found in Appendix A.

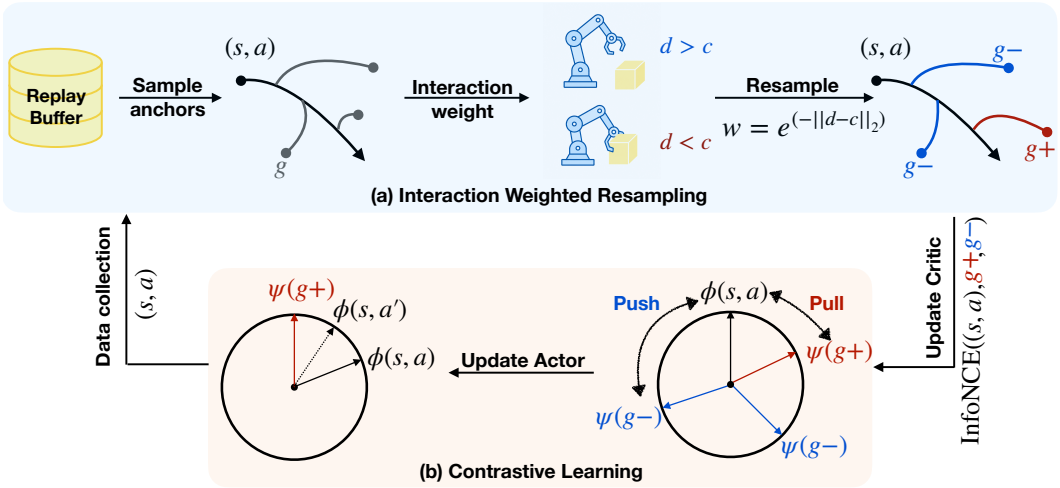


Figure 2: **Overview of the learning framework**. (a) IWR samples anchor state-action pairs from the replay buffer and reweights candidate future goals by their proximity to the interaction threshold, increasing the chance of selecting interaction-relevant positives. (b) The resampled positives and replay-marginal negatives are used in the standard CRL critic update. The learned energy then updates the actor, whose new rollouts are added back to the replay buffer.

5 Method

To address the weak identification of interactions under standard CRL sampling, we propose *Interaction-Weighted Resampling* (IWR). The goal of IWR is to shift the effective contrastive training distribution toward interaction-centered futures, while leaving the CRL critic objective unchanged.

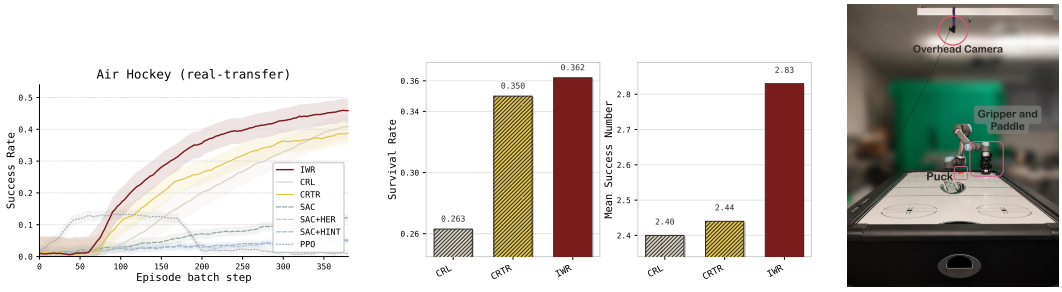
For an anchor transition $x_t = (s_t, a_t)$, we consider a set of candidate future goals $g_k = s_{t+k}$ from the same trajectory. Let $u_{p,t+k}$ and $y_{p,t+k}$ denote the positions of the actuated factor and target factor at the candidate future time $t + k$, respectively. We define their distance as

$$d_{t+k} = \|u_{p,t+k} - y_{p,t+k}\|_2.$$

Let c be a loose interaction threshold, encoding prior knowledge of where interaction is likely to occur, e.g., contact distance between the gripper and object. IWR assigns each candidate a future soft interaction weight:

$$w_k = \epsilon + \exp\left(-\frac{||d_{t+k} - c||_2}{2\sigma^2}\right), \tag{8}$$

where $\epsilon > 0$ preserves global coverage and σ controls the width of the interaction window. Figure 2 summarizes the learning pipeline. We first sample anchors (s, a) from the replay buffer and collect candidate future goals from the same trajectory. IWR computes the interaction weights in Eq. 8 and resamples positive futures toward likely interaction regions. The resampled positives are then used in the standard CRL critic objective in Eq. 3. The actor is updated by maximizing the learned goal-reaching energy, and the updated policy collects new transitions into the replay buffer, forming a self-improving loop. Then, the method follows the standard Contrastive RL pipeline: it trains the critic with Eq. 3 and updates the actor policy by $\pi = \text{argmax}_a E(s, a, g)$. Finally, the updated policy collects new transitions in the replay buffer. Thus, IWR focuses the sampling distribution of contrastive updates towards mode-switching regions to improve the modeling of this region. We describe the tighter link between IWR and Proposition 1 in Appendix C.



(a) Training curve. (b) Consecutive evaluation. (c) Real robot setup.

Figure 3: **Air hockey real-transfer**. (a) Training performance in simulation under the real-transfer setting. Shaded area indicates the confidence interval. (b) Consecutive evaluation after training in simulator. (c) Real robot air hockey setup.

6 Experiments

6.1 Simulation Experiment

Experimental Settings We evaluate IWR across three groups of interaction-centric environments: (i) **2D Box2D manipulation.** We test IWR with state inputs on five 2D manipulation tasks in Box2D [8] (Fig. A2), where the agent controls the blue ball to move the green ball into the red goal region. We report *success ticks*, defined as the number of ticks for which the green ball remains inside the red circle, normalized by 100 ticks, with each episode lasting 200 ticks. (ii) **Robotic manipulation.** We then evaluate IWR on four manipulation tasks: PUSH, PICK-PLACE, SWEEP-INTO, and PEG-INSERT, using success rate as the metric. (iii) **Robot air hockey.** Finally, we test IWR in the Air-Hockey simulator [13] using the velocity-observation mode. The task is to hit a falling puck into a small circular goal of radius $r = 0.06$, where the paddle starts on the lower board and strikes the puck toward the goal. We further evaluate performance in a consecutive goal-reaching setting. This setting challenges the trained policy to reach as many goals as possible within one minute. Once a goal is reached, a new goal is sampled. We report both the number of successful goal reaches and the survival rate, which indicates whether the puck avoids falling to the bottom within one minute. For sim-to-real transfer, we add observation noise and switch from velocity observations to history-based observations (Sec. 6.2).

We compare our method against two groups of baselines. (i) **Reward-based RL with sparse rewards.** We evaluate PPO [67] and SAC [28]. For SAC, we also consider variants with a hindsight replay buffer [3] and an interaction replay buffer [15]. (ii) **Contrastive RL methods.** We compare against single goal CRL [50] and CRTR [88]. CRL and CRTR differ in whether sampling is repeated R times from a single episode. Since our method belongs to the same family as CRTR, we set $R = 8$. CRTR can also be viewed as an ablation of our method with interaction-agnostic uniform sampling. For the CRL family, after the policy first succeeds at goal reaching, we further evaluate consecutive goal reaching: once a goal is reached, the goal position is changed, and the policy continues acting. We report the average consecutive survival rate, defined by whether the puck avoids falling to the bottom, and the goal-reaching count. Each task is repeated over 5 seeds and details are in Appendix D.

Results Table 1 reports the best average performance of all algorithms. The reward-free **CRL family outperforms the PPO algorithm and SAC family**, even when SAC is augmented with hindsight replay and interaction replay. The reward-based method has no success on MetaWorld and Air Hockey Real Robot. This agrees with previously observed [20] comparative results. IWR outperforms prior methods on most tasks, with an average gain of 19.8%. A key pattern is that the gains are **largest when interaction and goal-reaching signals are sparse**, supporting our hypothesis that IWR amplifies interaction-relevant supervision and learns a better goal-reaching energy landscape.

In `Box2D`, precise-control tasks require the agent to keep the green ball near the goal center, demanding both exploration and accurate interaction modeling. The hard variants make the objects smaller,

Table 1: Success rates over Air Hockey, Box2D, and MetaWorld tasks. Light red rows highlight interaction-centric tasks where IWR provides larger gains.

Task	PPO	SAC	SAC+HER	SAC+HINT	CRL	CRTR	IWR (Ours)
Air Hockey (Simulation)	0.617	0.145	0.398	0.422	0.695	0.727	**0.742 (+2.1%)**
Air Hockey (real-transfer)	0.160	0.215	0.129	0.125	0.477	0.465	**0.500 (+4.8%)**
Air Hockey (Real Robot)	0/20	0/20	0/20	0/20	5/20	2/20	**12/20 (+140.0%)**
Box2D (center)	0.086	0.058	0.088	0.088	0.278	0.274	**0.288 (+3.6%)**
Box2D (goal)	0.089	0.046	0.086	0.064	0.450	0.558	**0.709 (+27.1%)**
Box2D (hard)	0.060	0.042	0.064	0.076	0.317	0.365	**0.565 (+54.8%)**
Box2D (hard velocity)	0.148	0.149	0.152	0.139	0.387	0.377	**0.436 (+12.7%)**
Box2D (maze)	0.033	0.012	0.031	0.035	0.217	0.206	**0.223 (+2.8%)**
MetaWorld (peg insert)	0.000	0.000	0.000	0.000	0.430	0.367	**0.438 (+1.9%)**
MetaWorld (pick place)	0.000	0.000	0.004	0.000	0.266	0.305	**0.570 (+86.9%)**
MetaWorld (push)	0.000	0.000	0.004	0.000	0.699	**0.750**	0.730
MetaWorld (sweep into)	0.000	0.004	0.020	0.004	0.805	0.910	**0.926 (+1.8%)**
Average IWR improvement							**+19.8%**

further reducing successful interaction signals; here IWR yields larger gains. In the hard-velocity setting, random initial velocities occasionally create early successes, so the relative improvement is smaller but still substantial. This shows that IWR is especially useful when interaction events are rare but necessary for success. **`Metaworld`** shows a similar trend. In PUSH and SWEEP-INTO, random behavior can sometimes succeed, and the goal can be reached through multiple strategies, such as pushing or stabbing the object into the goal. In contrast, PICK-PLACE requires precise long-horizon interaction, since the policy must grasp and carry the object. IWR achieves a much larger gain in this task, suggesting that interaction-weighted sampling extends beyond the simplified one-step setting in Proposition 1 to hybrid dynamics with repeated interaction propagation, e.g., terms of the form A_1^k. For **`air hockey`**, the single-goal success-rate gain is modest, but Fig. 3a shows great improvements in real-transfer setting, which includes noise and discontinuous observations. This indicates IWR would be consistent when the environment is noisy, and the learned policy and critic is more generalizable with higher sample coverage near interaction.

Fig. 3b shows the evaluation on consecutive goal-hitting evaluation. This indicates that IWR improves not only isolated successes, but also the robustness and precision of interaction control. The effect is even more pronounced in the real-world setting, as discussed in Sec. 6.2.

6.2 Real-Robot Experiment

Experimental Settings In this work, we utilize the setting of Chuck et al. [13] as seen in Fig. A3 with a sim-to-real pipeline with domain randomization to transfer policies from the 2D simulator to a UR5e robot on one end of an air hockey table (see Fig. 4). The state space is unified between both 2D and 3D through puck detection and coordinate frame transformation. No training is performed on the real robot setup—all policies are transferred sim-to-real, and evaluated on a grid of 4×5 goals. Additional robot details, especially the sim-to-real pipeline, can be found in Appendix F.

Results As we see in Table 1, the success rate for IWR significantly exceeds that of the other methods (140% more success), even though the gap in simulation (Air Hockey real-transfer) is significantly smaller (5% increase). We believe this is because when transferring sim-to-real, the qualitative aspects of the policy, that is, how it induces interactions, make a larger difference. Because IWR emphasizes greater representation capacity for interactions, it makes contact with the puck more consistently, and with greater upward force. This results in more overall strikes, which significantly boosts the performance. Other comparable methods make contact with the puck only once, resulting in unrecoverable failures. Nonetheless, the performance improvement of IWR is substantial, suggesting

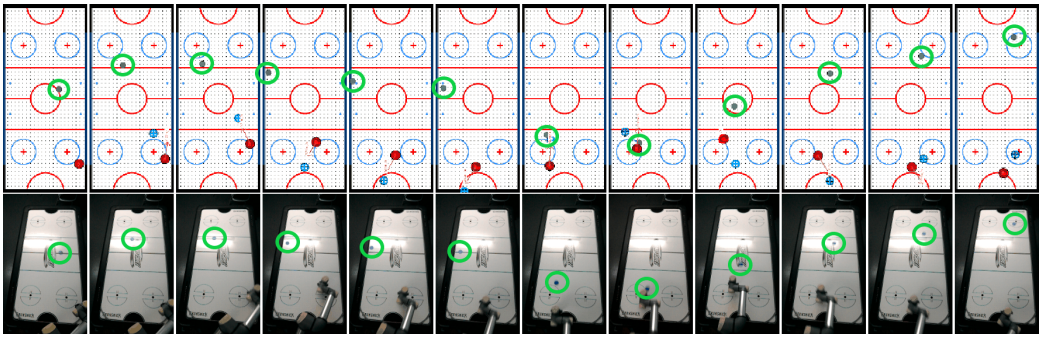


Figure 4: 2D sim paired with corresponding real frames. Puck emphasized with green circle.

that by focusing on interactions, it learns a more robust policy. This is qualitatively visible in the visualizations of the policy performance found in Appendix Figures A10-A9.

7 Conclusion and Limitations

By formulating manipulation as mode-conditioned dynamics, we illustrate why Contrastive Reinforcement Learning will struggle in manipulation. Interaction-weighted Resampling offers a practical, simple solution that provides an inductive bias towards this complexity, resulting in improved performance across interaction-rich tasks—especially those with rarer or more complex interactions, such as hard dynamic ball interactions, pick-and-place, and real-world air hockey. As robot learning moves towards more complex settings, handling this complexity will become increasingly crucial.

Limitations: Despite the promising success of Interaction-weighted Resampling, it remains an incomplete initial step toward leveraging the insight of mode-conditioned dynamics in robot learning. First, IWR still represents multimodal manipulation dynamics with a unimodal model—it only changes sampling. Second, CRL is one form of unsupervised RL, with limited representational power, but IWR can be applied to other unsupervised RL methods. Third, IWR assumes a factorized state and heuristic interaction indicators—integrating IWR with an end-to-end perception system is key to future systems. Fourth, CRL incurs a high sample-efficiency cost, preventing it from real-world application—our failures stem from the sim-to-real gap. Thus, future directions involve both algorithmic modifications and better robot systems for transferring core knowledge to the agent.

References

[1] Siddhant Agarwal, Harshit Sikchi, Peter Stone, and Amy Zhang. Proto successor measure: Representing the behavior space of an rl agent. *arXiv preprint arXiv:2411.19418*, 2024.

[2] Siddhant Agarwal, Caleb Chuck, Harshit Sikchi, Jiaheng Hu, Max Rudolph, Scott Niekum, Peter Stone, and Amy Zhang. A unified framework for unsupervised reinforcement learning algorithms. In *Workshop on Reinforcement Learning Beyond Rewards@ Reinforcement Learning Conference 2025*, 2025.

[3] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[4] Chenjia Bai, Lingxiao Wang, Yixin Wang, Zhaoran Wang, Rui Zhao, Chenyao Bai, and Peng Liu. Addressing hindsight bias in multigoal reinforcement learning. *IEEE Transactions on Cybernetics*, 53(1):392–405, 2021.

[5] Mahsa Bastankhah, Grace Liu, Dilip Arumugam, Thomas L. Griffiths, and Benjamin Eysenbach. Demystifying the mechanisms behind emergent exploration in goal-conditioned rl. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.14129.

[6] Roy Betser, Eyal Gofer, Meir Yossef Levi, and Guy Gilboa. Infonce induces gaussian distribution. *arXiv preprint arXiv:2602.24012*, 2026.

[7] Abhijat Biswas, Badal Arun Pardhi, Caleb Chuck, Jarrett Holtz, Scott Niekum, Henny Admoni, and Alessandro Allievi. Gaze supervision for mitigating causal confusion in driving agents. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2331–2338. IEEE, 2024.

[8] Erin Catto. Box2D: A 2d physics engine for games. https://box2d.org/, 2026. Accessed: 2026-05-28.

[9] Cheng Chi, Siyuan Feng, Zhenjia Xu, Eric A Cousineau, Benjamin Burchfiel, Shuran Song, et al. Visuomotor policy learning via action diffusion, September 4 2025. US Patent App. 18/594,842.

[10] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[11] Caleb Chuck, Supawit Chockchowwat, and Scott Niekum. Hypothesis-driven skill discovery for hierarchical deep reinforcement learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5572–5579. IEEE, 2020.

[12] Caleb Chuck, Kevin Black, Aditya Arjun, Yuke Zhu, and Scott Niekum. Granger causal interaction skill chains. *arXiv preprint arXiv:2306.09509*, 2023.

[13] Caleb Chuck, Carl Qi, Michael J Munje, Shuozhe Li, Max Rudolph, Chang Shi, Siddhant Agarwal, Harshit Sikchi, Abhinav Peri, Sarthak Dayal, et al. Robot air hockey: A manipulation testbed for robot learning with reinforcement learning. *arXiv preprint arXiv:2405.03113*, 2024.

[14] Caleb Chuck, Sankaran Vaidyanathan, Stephen Giguere, Amy Zhang, David Jensen, and Scott Niekum. Automated discovery of functional actual causes in complex environments. *arXiv preprint arXiv:2404.10883*, 2024.

[15] Caleb Chuck, Fan Feng, Carl Qi, Chang Shi, Siddhant Agarwal, Amy Zhang, and Scott Niekum. Null counterfactual factor interactions for goal-conditioned reinforcement learning. *arXiv preprint arXiv:2505.03172*, 2025.

[16] Tal Daniel, Carl Qi, Dan Haramati, Amir Zadeh, Chuan Li, Aviv Tamar, Deepak Pathak, and David Held. Latent particle world models: Self-supervised object-centric stochastic dynamics modeling. *arXiv preprint arXiv:2603.04553*, 2026.

[17] Julien Denize, Jaonary Rabarisoa, Astrid Orcesi, Romain Hérault, and Stéphane Canu. Similarity contrastive estimation for self-supervised soft contrastive learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2706–2716, 2023.

[18] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9588–9597, 2021.

[19] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[20] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tc5qisoB-C.

[21] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35603–35620, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e7663e974c4ee7a2b475a4775201ce1f-Paper-Conference.pdf.

[22] Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via interpolation: Contrastive representations provably enable planning and inference. *Advances in Neural Information Processing Systems*, 37:58901–58928, 2024.

[23] Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Marc G Bellemare, Alessandro Lazaric, and Ahmed Touati. Compositional planning with jumpy world models. *arXiv preprint arXiv:2602.19634*, 2026.

[24] Chen Feng and Ioannis Patras. Adaptive soft contrastive learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 2721–2727. IEEE, 2022.

[25] Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

[26] Tuba Girgin and Emre Uğur. Multiobject graph affordance network: Goal-oriented planning through learned compound object affordances. *IEEE Transactions on Cognitive and Developmental Systems*, 17(4):847–858, 2024.

[27] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[28] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[29] Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors*, 23(7):3762, 2023.

[30] Dan Haramati, Carl Qi, Tal Daniel, Amy Zhang, Aviv Tamar, and George Konidaris. Hierarchical entity-centric reinforcement learning with factored subgoal diffusion. *arXiv preprint arXiv:2602.02722*, 2026.

[31] David T Hoffmann, Nadine Behrmann, Juergen Gall, Thomas Brox, and Mehdi Noroozi. Ranking info noise contrastive estimation: Boosting contrastive learning via ranked positives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 897–905, 2022.

[32] Seyed Mohammad Hadi Hosseini and Mahdieh Soleymani Baghshah. Susd: Structured unsupervised skill discovery through state factorization. *arXiv preprint arXiv:2602.01619*, 2026.

[33] Jiaheng Hu, Zizhao Wang, Peter Stone, and Roberto Martín-Martín. Disentangled unsupervised skill discovery for efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 37:76529–76552, 2024.

[34] Yixuan Huang, Adam Conkey, and Tucker Hermans. Planning for multi-object manipulation with graph neural network relational classifiers. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1822–1829. IEEE, 2023.

[35] Zhiao Huang. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. *Robotics: Science and Systems XVIII*.

[36] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2785–2795, 2022.

[37] Inwoo Hwang, Yunhyeok Kwak, Suhyung Choi, Byoung-Tak Zhang, and Sanghack Lee. Fine-grained causal dynamics learning with quantization for improving robustness in reinforcement learning. *arXiv preprint arXiv:2406.03234*, 2024.

[38] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pp. 1094–8. Citeseer, 1993.

[39] Kushal Kedia, Atiksh Bhardwaj, Prithwish Dan, and Sanjiban Choudhury. Interact: Transformer models for human intent prediction conditioned on robot actions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 621–628. IEEE, 2024.

[40] Alexander Khazatsky, Ashvin Nair, Daniel Jing, and Sergey Levine. What can i do here? learning new skills by imagining visual affordances. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14291–14297. IEEE, 2021.

[41] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[42] Jaeill Kim, Duhun Hwang, Eunjung Lee, Jangwon Suh, Jimyeong Kim, and Wonjong Rhee. Enhancing contrastive learning with efficient combinatorial positive pairing. *arXiv preprint arXiv:2401.05730*, 2024.

[43] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[44] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021. DOI: 10.15607/RSS.2021.XVII.011. URL https://arxiv.org/abs/2107.04034.

[45] Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Unsupervised reinforcement learning with contrastive intrinsic control. *Advances in Neural Information Processing Systems*, 35:34478–34491, 2022.

[46] Tabitha Edith Lee, Shivam Vats, Siddharth Girdhar, and Oliver Kroemer. Scale: Causal learning and discovery of robot manipulation skills using simulation. In *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

[47] Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Spartan: A sparse transformer world model attending to what matters. *Advances in Neural Information Processing Systems*, 38: 154089–154114, 2025.

[48] Andrew Levy. *Unsupervised Skill Discovery with Empowerment*. PhD thesis, Brown University PROVIDENCE, RHODE ISLAND, 2025.

[49] Yixin Lin, Austin S Wang, Eric Undersander, and Akshara Rai. Efficient and interpretable robot manipulation with graph neural networks. *IEEE Robotics and Automation Letters*, 7(2): 2740–2747, 2022.

[50] Grace Liu, Michael Tang, and Benjamin Eysenbach. A single goal is all you need: Skills and exploration emerge from contrastive rl without rewards, demonstrations, or subgoals. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xCkgX4Xfu0.

[51] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[52] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, et al. Eureka: Human-level reward design via coding large language models. In *International conference on learning Representations*, volume 2024, pp. 26516–26560, 2024.

[53] Alireza Modirshanechi, Benjamin Eysenbach, Peter Dayan, and Eric Schulz. Unifying goal-conditioned rl and unsupervised skill learning via control-maximization. *arXiv preprint arXiv:2605.06145*, 2026.

[54] Vivek Myers, Andre Wang He, Kuan Fang, Homer Rich Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal representations for instruction following: A semi-supervised language interface to control. In *Conference on Robot Learning*, pp. 3894–3908. PMLR, 2023.

[55] Vivek Myers, Bill Zheng, Anca Dragan, Kuan Fang, and Sergey Levine. Temporal representation alignment: Successor features enable emergent compositionality in robot instruction following. *Advances in Neural Information Processing Systems*, 38:149934–149961, 2026.

[56] Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. *Advances in Neural Information Processing Systems*, 38:19654–19679, 2026.

[57] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

[58] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

[59] Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. In *International Conference on Learning Representations*, volume 2024, pp. 18579–18603, 2024.

[60] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018. DOI: 10.1109/ICRA.2018.8460528. URL https://arxiv.org/abs/1710.06537.

[61] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 33:3976–3990, 2020.

[62] Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems*, 35: 18143–18156, 2022.

[63] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[64] Carl Qi, Dan Haramati, Tal Daniel, Aviv Tamar, and Amy Zhang. Ec-diffuser: Multi-object manipulation via entity-centric behavior generation. In *International Conference on Learning Representations*, volume 2025, pp. 74835–74858, 2025.

[65] Rafael Rodriguez-Sanchez, Cameron Allen, and George Konidaris. From pixels to factors: Learning independently controllable state variables for reinforcement learning. *arXiv preprint arXiv:2510.02484*, 2025.

[66] Max Rudolph, Caleb Chuck, Kevin Black, Misha Lvovsky, Scott Niekum, and Amy Zhang. Learning action-based representations using invariance. *arXiv preprint arXiv:2403.16369*, 2024.

[67] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. DOI: 10.48550/arXiv .1707.06347. URL https://arxiv.org/abs/1707.06347.

[68] Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:22905–22918, 2021.

[69] Maximilian Sieb, Zhou Xian, Audrey Huang, Oliver Kroemer, and Katerina Fragkiadaki. Graph-structured visual imitation. In *Conference on Robot learning*, pp. 979–989. PMLR, 2020.

[70] Harshit Sushil Sikchi, Siddhant Agarwal, Pranaya Jajoo, Samyak Parajuli, Caleb Chuck, Max Rudolph, Peter Stone, Amy Zhang, and Scott Niekum. Rlzero: Direct policy inference from language without in-domain supervision. *Advances in Neural Information Processing Systems*, 38:83365–83398, 2026.

[71] Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3352–3360. IEEE, 2025.

[72] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8(1):153–188, 2025.

[73] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. DOI: 10.1109/IROS.2017.8202133. URL https://arxiv.org/abs/1703.06907.

[74] Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

[75] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

[76] Kevin Wang, Ishaan Javali, Michał Bortkiewicz, Tomasz Trzcinski, and Benjamin Eysenbach. 1000 layer networks for self-supervised rl: Scaling depth can enable new goal-reaching capabilities. *Advances in Neural Information Processing Systems*, 38:157643–157670, 2026.

[77] Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim GJ Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. *arXiv preprint arXiv:2603.12231*, 2026.

[78] Zizhao Wang, Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. Elden: Exploration via local dependencies. *Advances in Neural Information Processing Systems*, 36:15456–15474, 2023.

[79] Zizhao Wang, Jiaheng Hu, Caleb Chuck, Stephen Chen, Roberto Martín-Martín, Amy Zhang, Scott Niekum, and Peter Stone. Skild: Unsupervised skill discovery guided by factor interactions. *Advances in Neural Information Processing Systems*, 37:77748–77776, 2024.

[80] Zizhao Wang, Chang Shi, Jiaheng Hu, Kevin Rohling, Roberto Martín-Martín, Amy Zhang, and Peter Stone. Factored latent action world models. *arXiv preprint arXiv:2602.16229*, 2026.

[81] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

[82] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. *arXiv preprint arXiv:2407.07889*, 2024.

[83] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Particle-grid neural dynamics for learning deformable object models from rgb-d videos. *arXiv preprint arXiv:2506.15680*, 2025.

[84] Bill Chunyuan Zheng, Vivek Myers, Benjamin Eysenbach, and Sergey Levine. Multistep quasimetric learning for scalable goal-conditioned reinforcement learning. *arXiv preprint arXiv:2511.07730*, 2025.

[85] Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. In *International Conference on Learning Representations*, volume 2024, pp. 47577–47601, 2024.

[86] Chongyi Zheng, Royina Karegoudra Jayanth, and Benjamin Eysenbach. Can we really learn one representation to optimize all rewards? *arXiv preprint arXiv:2602.11399*, 2026.

[87] Yifeng Zhu, Arisrei Lim, Peter Stone, and Yuke Zhu. Vision-based manipulation from single human video with open-world object graphs. *Autonomous Robots*, 50(2):27, 2026.

[88] Alicja Ziarko, Michał Bortkiewicz, Michał Zawalski, Benjamin Eysenbach, and Piotr Miłoś. Contrastive representations for temporal reasoning. *Advances in Neural Information Processing Systems*, 38:109229–109259, 2026.

[89] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Theory Details for Proposition 1 **17**

A.1 From Factored Dynamics to Local Affine Representation Dynamics 17

A.2 One-Reset Simplification 19

A.3 Future Mean Under One Interaction 20

A.4 Propagation of Bridge Error 20

A.5 Energy Error Induced by Bridge Error 21

A.6 Interpretation 22

B Extended Related Work **23**

C Interaction-weighted Resampling Details **23**

D Training Setup **24**

D.1 Tasks and Algorithm Grid 24

D.2 Shared Training Defaults 24

D.3 CRL-Family Objective 25

D.4 IWR Replay Sampling 25

D.5 Sparse-Reward PPO and SAC Baselines 25

D.6 Evaluation Metrics 26

E Simulation Environment Setup **26**

E.1 Simulation Setting 26

E.2 Air Hockey 27

E.3 Box2D 27

E.4 MetaWorld 27

E.5 Post-Hoc Air Hockey Consecutive Evaluation 29

F Real Setup **29**

G Energy Visualization **30**

A Theory Details for Proposition 1

A.1 From Factored Dynamics to Local Affine Representation Dynamics

We first justify the switched affine representation model used in the main paper. The key point is not that manipulation dynamics are globally linear. Rather, the claim is that after factorization, the local dependency structure of the target factor changes across passive and interaction modes. This change forces us to distinguish the passive operator A_0 from the interaction operator A_1, and to include an interaction-dependent affine term b_t.

Let the state be factored as $s_t = (u_t, y_t)$, where u_t is the actuated factor and y_t is the target factor. We write the target representation as

$$\psi_t \triangleq \psi(y_t).$$

The CRL critic conditions on the anchor through $\phi(s_t, a_t) = \phi(u_t, y_t, a_t)$. If the action is produced by a deterministic policy, $a_t = \pi(u_t, y_t, g)$, then the same argument can be viewed as conditioning on (u_t, y_t) under the current policy. Thus, for the theory, we only need to study how the anchor variables induce the next target representation ψ_{t+1}.

Dependency contrast. In locomotion-like domains, the controlled factor and the goal factor are typically the same state variable. If this factor is denoted by z_t, the local dynamics have the form

$$z_{t+1} = f_{\text{loc}}(z_t, a_t),$$

and the action affects the goal-relevant state almost everywhere:

$$\left\| \frac{\partial f_{\text{loc}}(z_t, a_t)}{\partial a_t} \right\| > 0 \quad \text{for a.e. } (z_t, a_t).$$

Under a fixed policy or fixed anchor action, this gives a single smooth closed-loop transition map. This is the setting in which a single local temporal operator is a natural approximation for the future representation, consistent with the linear-Gaussian view of temporal contrastive representations [22].

Manipulation has a different dependency structure. The target factor is often not directly actuated. Using the notation from the main paper, the target dynamics are

$$y_{t+1} = \begin{cases} f_0(y_t), & \omega_t = 0, \\ f_1(u_t, y_t, a_t), & \omega_t = 1, \end{cases}$$

where $\omega_t = 0$ denotes passive target dynamics and $\omega_t = 1$ denotes an interaction. Within each smooth mode, the action-to-target derivative is therefore gated by the interaction indicator:

$$\frac{\partial y_{t+1}}{\partial a_t} = \begin{cases} 0, & \omega_t = 0, \\ \frac{\partial f_1(u_t, y_t, a_t)}{\partial a_t}, & \omega_t = 1. \end{cases}$$

Thus, action influence on the target is absent in the passive mode and appears only during interaction. This is a structural property of manipulation, not a choice of parameterization.

Assumption 1 (Mode-dependent target controllability)**.** *Let*

$$H_0(y) \triangleq \psi(f_0(y)), \qquad H_1(u, y, a) \triangleq \psi(f_1(u, y, a)).$$

We assume that H_0 and H_1 are differentiable in the local regions considered. In the passive mode, the represented target update has no direct action dependence:

$$\frac{\partial H_0(y)}{\partial a} = 0.$$

In the interaction mode, there exists a local interaction point $(\bar{u}, \bar{y}, \bar{a})$ and an action direction v such that

$$D_a H_1(\bar{u}, \bar{y}, \bar{a}) v \neq 0.$$

Assumption 1 implies that a target-only predictor is appropriate in the passive mode, but insufficient in the interaction mode. The reason is that two anchors can share the same target state y_t while differing in the actuated factor or action. During interaction, these two anchors can produce different next target states. A predictor depending only on ψ_t would assign the same next representation to both anchors, and therefore cannot represent the interaction update.

Lemma 1 (Need for an interaction-dependent residual)**.** *Suppose Assumption 1 holds. In the passive mode, a first-order local predictor of the represented target update can be written as a target-only linear predictor. In the interaction mode, no target-only linear predictor of the form $A\psi(y)$ can approximate the represented target update to first order in the interaction context. Therefore, the interaction-mode local predictor requires an additional context-dependent residual, yielding the affine form*

$$\psi_{t+1} = A_1\psi_t + b_t.$$

Proof. We prove the two claims separately.

Passive mode. In the passive mode,

$$\psi_{t+1} = H_0(y_t) = \psi(f_0(y_t)).$$

Since H_0 depends only on y_t, its first-order local expansion in representation coordinates around $\bar{y}$ has the form

$$H_0(y) = H_0(\bar{y}) + A_0\big(\psi(y) - \psi(\bar{y})\big) + o(\|\psi(y) - \psi(\bar{y})\|)\,.$$

After centering the local coordinates, or equivalently absorbing the affine intercept into an augmented representation coordinate, this becomes

$$\psi_{t+1} = A_0\psi_t + o(\|\psi_t - \psi(\bar{y})\|)\,.$$

Thus, a target-only linear predictor is sufficient to first order in the passive mode.

Interaction mode. Assume for contradiction that a target-only linear predictor is first-order sufficient in the interaction mode. Then there exists a matrix A such that, for all sufficiently small h,

$$H_1(\bar{u}, \bar{y}, \bar{a} + hv) = A\psi(\bar{y}) + o(|h|).$$

Evaluating the same expression at $h = 0$ gives

$$H_1(\bar{u}, \bar{y}, \bar{a}) = A\psi(\bar{y}).$$

Subtracting the two equations yields

$$H_1(\bar{u}, \bar{y}, \bar{a} + hv) - H_1(\bar{u}, \bar{y}, \bar{a}) = o(|h|).$$

Dividing by h and taking $h \to 0$ gives

$$D_a H_1(\bar{u}, \bar{y}, \bar{a})v = 0.$$

This contradicts Assumption 1, which states that

$$D_a H_1(\bar{u}, \bar{y}, \bar{a})v \neq 0.$$

Therefore, no target-only linear predictor can be first-order sufficient in the interaction mode.

It remains to show the affine form. The first-order Taylor expansion of H_1 around $(\bar{u}, \bar{y}, \bar{a})$ is

$$H_1(u, y, a) = H_1(\bar{u}, \bar{y}, \bar{a}) + A_1\big(\psi(y) - \psi(\bar{y})\big) + D_u H_1(\bar{u}, \bar{y}, \bar{a})(u - \bar{u}) + D_a H_1(\bar{u}, \bar{y}, \bar{a})(a - \bar{a}) + o(\Delta),$$

where

$$\Delta = \|\psi(y) - \psi(\bar{y})\| + \|u - \bar{u}\| + \|a - \bar{a}\|.$$

Rearranging the first-order terms gives

$$H_1(u, y, a) = A_1\psi(y) + b(u, y, a) + o(\Delta),$$

where

$$b(u, y, a) = H_1(\bar{u}, \bar{y}, \bar{a}) - A_1\psi(\bar{y}) + D_u H_1(\bar{u}, \bar{y}, \bar{a})(u - \bar{u}) + D_a H_1(\bar{u}, \bar{y}, \bar{a})(a - \bar{a}).$$

At time t, define

$$b_t \triangleq b(u_t, y_t, a_t).$$

Therefore, the interaction-mode local predictor has the affine form

$$\psi_{t+1} = A_1\psi_t + b_t + o(\Delta).$$

Dropping the higher-order local error gives the switched affine approximation used in the main analysis. □

We now define the local affine approximation used in the proof. The operator A_0 is the local linear predictor for passive target evolution. The operator A_1 is the target-state coefficient in the interaction mode. The residual b_t is the part of the interaction update induced by the actuated factor and action after accounting for the target-state term.

Assumption 2 (Mode-wise local affine representation dynamics)**.** *Along the local trajectory segment analyzed in Appendix A, the conditional mean dynamics of the target representation satisfy*

$$\psi_{t+1} = A_0\psi_t, \qquad \omega_t = 0,$$

and

$$\psi_{t+1} = A_1\psi_t + b_t, \qquad \omega_t = 1.$$

Equivalently, the two cases can be written as the switched affine model

$$\psi_{t+1} = A_{\omega_t}\psi_t + \omega_t b_t.$$

Here b_t is zero outside interaction because it is multiplied by ω_t.

This assumption is a local regression statement in representation space. It does not require the original manipulation dynamics to be globally linear. Instead, it says that near a trajectory segment, the passive target update is approximated by a target-only linear operator, while the interaction update requires a separate local operator plus an interaction-dependent residual. This is the representation-level form of the dependency change in the factored dynamics. The one-reset analysis below studies how an error in this residual is propagated to later endpoint energies.

A.2 One-Reset Simplification

We now formalize the one-reset setting used in Proposition 1. Let τ denote the first future interaction offset after anchor time t. The one-reset mode sequence is

$$\underbrace{0, \ldots, 0}_{\tau \text{ passive steps}}, 1, \underbrace{0, \ldots, 0}_{k \text{ passive steps}},$$

where k is the number of passive steps after the interaction. Thus, the endpoint is at time $t + \tau + 1 + k$.

Definition 1 (Interaction bridge shift)**.** *The interaction bridge shift is the difference between the actual interaction update and the passive update that would have occurred without interaction:*

$$\delta_\tau \triangleq (A_1 - A_0)\psi_{t+\tau} + b_{t+\tau}. \tag{A1}$$

Equivalently,

$$A_1\psi_{t+\tau} + b_{t+\tau} = A_0\psi_{t+\tau} + \delta_\tau. \tag{A2}$$

The bridge shift isolates the effect of the interaction itself. Before the bridge, the target evolves passively. At the bridge, the target receives the additional shift δ_τ. After the bridge, this shift is carried forward by passive dynamics.

Definition 2 (Bridge estimation error)**.** *Let $\widehat{\delta}_\tau$ be the bridge shift represented by the learned energy model. The local interaction-bridge estimation error is*

$$e \triangleq \widehat{\delta}_\tau - \delta_\tau. \tag{A3}$$

A.3 Future Mean Under One Interaction

Lemma 2 (One-reset future mean)**.** *Under Assumption 2 and the one-reset setting, the target representation k passive steps after the interaction has mean*

$$\mu_k = A_0^{\tau+1+k}\psi_t + A_0^k\delta_\tau. \tag{A4}$$

Proof. Before the interaction, the target follows passive dynamics for τ steps:

$$\psi_{t+\tau} = A_0^\tau \psi_t. \tag{A5}$$

At the interaction step, Definition 1 gives

$$\psi_{t+\tau+1} = A_0\psi_{t+\tau} + \delta_\tau. \tag{A6}$$

After the interaction, the target follows passive dynamics for k more steps:

$$\psi_{t+\tau+1+k} = A_0^k\left(A_0\psi_{t+\tau} + \delta_\tau\right). \tag{A7}$$

Substituting Eq. (A5) into Eq. (A7) yields

$$\begin{aligned}\psi_{t+\tau+1+k} &= A_0^k\left(A_0A_0^\tau\psi_t + \delta_\tau\right) &\text{(A8)}\\ &= A_0^{\tau+1+k}\psi_t + A_0^k\delta_\tau. &\text{(A9)}\end{aligned}$$

This proves Eq. (A4). □

Lemma 2 shows the key structure: the interaction bridge shift is not observed at the endpoint as δ_τ, but as the propagated quantity $A_0^k\delta_\tau$.

A.4 Propagation of Bridge Error

Lemma 3 (Propagated bridge error)**.** *If the learned bridge shift has error $e = \widehat{\delta}_\tau - \delta_\tau$, then the induced endpoint mean error after k passive steps is*

$$\widehat{\mu}_k - \mu_k = A_0^k e. \tag{A10}$$

Consequently,

$$\|\widehat{\mu}_k - \mu_k\| \le \|A_0^k\|\|e\|. \tag{A11}$$

Proof. By Lemma 2, the true endpoint mean is

$$\mu_k = A_0^{\tau+1+k}\psi_t + A_0^k\delta_\tau. \tag{A12}$$

Using the learned bridge shift $\widehat{\delta}_\tau$ gives

$$\widehat{\mu}_k = A_0^{\tau+1+k}\psi_t + A_0^k\widehat{\delta}_\tau. \tag{A13}$$

Subtracting Eq. (A12) from Eq. (A13) gives

$$\begin{aligned}\widehat{\mu}_k - \mu_k &= A_0^k\left(\widehat{\delta}_\tau - \delta_\tau\right) &\text{(A14)}\\ &= A_0^k e. &\text{(A15)}\end{aligned}$$

Taking norms and using submultiplicativity gives Eq. (A11). □

A.5 Energy Error Induced by Bridge Error

We now translate the propagated endpoint mean error into an energy error. For the appendix proof, we use the whitened local Gaussian energy

$$E_k = -\frac{1}{2}\left\|\psi(y_f) - \mu_k\right\|^2 + \text{const.} \tag{A16}$$

This is the isotropic form of the local Gaussian score used in the main analysis. A non-isotropic covariance only changes the norm and introduces constant conditioning factors, which are suppressed in the main-paper proportionality statement.

Assumption 3 (Normalized local endpoint region)**.** *The sampled endpoint lies in the local quadratic region of the correct mean:*

$$\left\|\psi(y_f) - \mu_k\right\| \le 1. \tag{A17}$$

This normalization is only for presentation. If the local residual is bounded by a constant other than 1, the first term in the final bound is multiplied by that constant.

Lemma 4 (Exact energy perturbation)**.** *Suppose the propagated bridge-error relation in Lemma 3 holds. Then the energy perturbation caused by bridge error is*

$$\widehat{E}_k - E_k = \left(\psi(y_f) - \mu_k\right)^\top A_0^k e - \frac{1}{2}\left\|A_0^k e\right\|^2. \tag{A18}$$

Proof. By Lemma 3,

$$\widehat{\mu}_k = \mu_k + A_0^k e. \tag{A19}$$

Using Eq. (A16), the learned energy is

$$\widehat{E}_k = -\frac{1}{2}\left\|\psi(y_f) - \widehat{\mu}_k\right\|^2 + \text{const.} \tag{A20}$$

Substituting Eq. (A19) gives

$$\widehat{E}_k = -\frac{1}{2}\left\|\psi(y_f) - \mu_k - A_0^k e\right\|^2 + \text{const.} \tag{A21}$$

Therefore,

$$\widehat{E}_k - E_k = -\frac{1}{2}\left[\left\|\psi(y_f) - \mu_k - A_0^k e\right\|^2 - \left\|\psi(y_f) - \mu_k\right\|^2\right] \tag{A22}$$

$$= \left(\psi(y_f) - \mu_k\right)^\top A_0^k e - \frac{1}{2}\left\|A_0^k e\right\|^2. \tag{A23}$$

This proves Eq. (A18). □

Corollary A.1 (Endpoint energy-error bound)**.** *Suppose Lemma 4 holds and Assumption 3 holds. Then*

$$|\widehat{E}_k - E_k| \le \|A_0^k\|\|e\| + \frac{1}{2}\|A_0^k\|^2\|e\|^2. \tag{A24}$$

Proof. Starting from Lemma 4, apply the triangle inequality:

$$|\widehat{E}_k - E_k| \le \left|\left(\psi(y_f) - \mu_k\right)^\top A_0^k e\right| + \frac{1}{2}\left\|A_0^k e\right\|^2. \tag{A25}$$

By Cauchy–Schwarz,

$$\left|\left(\psi(y_f) - \mu_k\right)^\top A_0^k e\right| \le \left\|\psi(y_f) - \mu_k\right\|\left\|A_0^k e\right\|. \tag{A26}$$

Using Assumption 3,

$$\left|(\psi(y_f) - \mu_k)^\top A_0^k e\right| \leq \left\|A_0^k e\right\|. \tag{A27}$$

Finally, by submultiplicativity,

$$\left\|A_0^k e\right\| \leq \|A_0^k\|\|e\|. \tag{A28}$$

Substituting these bounds into Eq. (A25) gives

$$|\widehat{E}_k - E_k| \leq \|A_0^k\|\|e\| + \frac{1}{2}\|A_0^k\|^2\|e\|^2. \tag{A29}$$

□

Corollary A.1 is the formal version of Proposition 1. Removing constants associated with local normalization and whitening gives the main-paper scaling:

$$\sup|\widehat{E}_k - E_k| \propto \|A_0^k\|\|e\| + \frac{1}{2}\|A_0^k\|^2\|e\|^2.$$

A.6 Interpretation

The proof separates the endpoint error into two factors. The first factor, $\|e\|$, is the local error in estimating the interaction bridge shift. This error becomes large when few training samples directly cover the interaction boundary. The second factor, $\|A_0^k\|$, is the passive post-interaction propagation factor. Even if the bridge error is small, passive dynamics can rotate, attenuate, or amplify the error before the endpoint is used in the contrastive objective.

This explains why endpoint-only CRL can learn fragile energies near contact. If the positive future is sampled long after the interaction, CRL supervises the interaction action only through the propagated term $A_0^k \delta_\tau$. Interaction-weighted resampling targets the two terms in the bound: it increases bridge-centered samples, reducing $\|e\|$, and it favors endpoints closer to the interaction, reducing the effective post-interaction propagation length k.

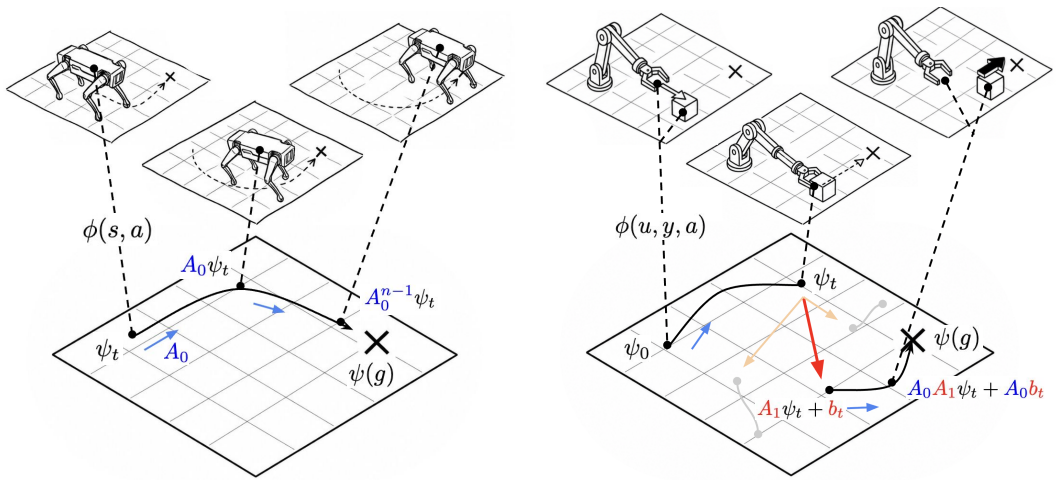


Figure A1: **Interaction-induced mode changes break the single-operator CRL geometry**. In smooth locomotion domains, future representations can often be organized by repeatedly applying a single temporal operator, producing a coherent trajectory toward the goal. In manipulation, interactions such as contact or object transfer switch the active dynamics and introduce residual displacements that depend on the actuated factor and action. Hence, the future distribution becomes piecewise and branching, making standard CRL smoothing over critical interaction transitions.

B Extended Related Work

This work both formalizes the challenges when applying contrastive reinforcement learning to manipulation and proposes a sampling-based strategy for leveraging interactions to mitigate those challenges. This places it in the intersection of work investigating CRL and robot learning for manipulation using interactions.

Contrastive Reinforcement Learning: CRL is the application of contrastive learning [57] to goal-conditioned reinforcement learning (GCRL), both as a classification [20] and regression objective [21]. Recent work has expanded this towards metric learning [84, 56], planning as time series modeling [22], language alignment [54, 55] and combinatorial reasoning and search [77, 88]. Interaction-weighted Resampling bears the closest similarity to Ziarko et al. [88], which also employs a resampling procedure, but IWR leverages interactions to target the resampling towards difficult-to-model dynamics, while Ziarko et al. [88] utilizes different contexts for in-trajectory sampling. We also build on the successes of CRL when applied in simulated manipulation [76], especially related to emergent exploration [50, 5], leveraging these properties which allow for limited manipulation performance in a sample-efficient manner relative to prior GCRL work [63, 38, 51, 3, 4]. Besides contrastive learning as applied to GCRL, IWR draws on ideas from general contrastive learning, especially related to leveraging privileged information such as class labels (wherein our case we leverage interactions and in-trajectory sampling) [24, 17, 41, 31], and debiasing strategies which carefully select negative examples, which we adapt for interaction-based sampling [10, 36, 18]. Our observations about the relationship between non-smooth dynamics and smooth representations learned by contrastive methods are in part inspired by Betser et al. [6], which observes that the distribution of representations tends towards a thin-shell Gaussian. Finally, recent work has shown close connections between contrastive RL methods and the general class of methods used in unsupervised Reinforcement Learning [2], including learning to represent all rewards through successor representations [70, 86, 74, 1, 85], mutual information skill learning [53, 48], representation learning [25, 81, 66] and world models [23]. However, the application of factorization and especially entity interactions to the methods based on contrastive learning, which we use in IWR, remains limited.

Manipulation Learning with Interactions: Robot manipulation encompasses a wide range of tasks, from grasping and moving objects to striking, inserting or throwing [29]. Interactions [14] have been incorporated into learning for manipulation in a variety of ways, including controllability [68], data augmentation [61, 62], hierarchical reinforcement learning [11, 12, 79], skill learning [33, 65, 32], causal modeling [46, 7] and exploration [78]. This work introduces a novel formalization for the challenges of robot learning for manipulation as mode-changing dynamics, and IWR is a novel algorithm for incorporating interaction information into the training of contrastive RL. IWR shows significantly improved sample efficiency, and the first application to robot air hockey [13]. Other work in manipulation that is more loosely leveraging interactions includes work centered around modeling dynamics, such as through graph networks [69, 34, 49, 83, 35, 39], (which incorporate interactions through the message passing framework), human conditional intent [82] or human video [71, 87], and affordances [40, 26]. Most closely related is the work investigating dynamics modeling [30, 80] which learn structures that incorporate factors for policy learning, often also identifying the factors themselves from observations [16, 64]. IWR is applied to CRL, which learns structure for GCRL policies, but these model-learning algorithms are likely to be effective in concert with CRL methods, where the dynamics model and policies can work together.

C Interaction-weighted Resampling Details

In this section, we provide some description to give a better understanding of how IWR addresses the challenge of mode switching.

IWR does not change the CRL loss or its energy landscape directly. Instead, it changes the effective sampling distribution of contrastive updates. This provides an inductive bias toward mode-switching

regions, giving more critic updates to interaction-relevant transitions rather than to easy passive motion or object-reaching behavior. In terms of our analysis (Section 4), IWR increases the probability that sampled positive futures cross the interaction bridge near the latent interaction time τ_t. Such samples expose the interaction shift directly, instead of observing it only after long passive propagation through powers of A_0. Intuitively, based on the analysis of intrinsic reward [5], our paradigm shifts intrinsic reward-driven exploration from "moving the object to the future" to "learning how to control the object to move to the future." This also connects to Proposition 1 as increasing the coverage of interaction-centered samples reduces the error contribution from the interaction residual. Exact bridge samples provide direct supervision for the piecewise jump term, while the soft weighting in Eq. 8 also covers nearby positives and reduces error accumulation from repeatedly propagating the interaction signal through A_0^k. Consequently, IWR reduces the passive bias of ordinary future sampling and improves the conditioning of interaction-relevant representation learning.

D Training Setup

D.1 Tasks and Algorithm Grid

Paper experiments use five seeds $\{7, 8, 9, 10, 11\}$. Experiment grid contains three simulation backends and seven algorithms.

Table A1: Paper-core benchmark grid.

Backend	Tasks	Count
Air Hockey	puck-goal position, $r = 0.06$, observation mode=velocity / real-transfer-history	2
Box2D	center, goal, hard, hard-velocity, maze	5
MetaWorld	push, pick-place, peg-insert-side, sweep-into	4
Algorithms	PPO, SAC, SAC+HER, SAC+HINT, CRL, CRTR, IWR	7

D.2 Shared Training Defaults

Shared training details can be found in Table A2

Table A2: Shared Training Defaults

Setting	Value
Total environment steps	10,000,000
Train environments	256
Batch size	64
Replay capacity	200,000
Minimum replay size	1,024
Random exploration warmup	200,000
Updates per environment iteration	16
Network architecture	MLP with hidden sizes $(512, 512)$
Representation dimension	256 for CRL-family critics
Future discount for future sampling	$\gamma_f = 0.99$
Random actor-goal fraction	0.5
Critic type	CPC
Log-sum-exp regularizer coefficient	0.01
Actor learning rate	3×10^{-4}
Critic learning rate	3×10^{-4}
Optimizer	Adam

Table A3: Backend-specific training settings.

Backend	Eval Envs	Termination
Air Hockey	128	First Success or 200 ticks
Box2D	128	200 ticks
MetaWorld	32	First Success or 200 ticks

D.3 CRL-Family Objective

For CRL, CRTR, and IWR, the actor and critic are identical except for replay sampling. Given anchor state-action pair (s_i, a_i) and sampled future goal g_i^+, the critic score is

$$E_\theta(s_i, a_i, g_j) = \phi_\theta(s_i, a_i)^\top \psi_\theta(g_j),$$

with cosine-normalized score on Air Hockey and dot-product score on Box2D and MetaWorld. The in-batch CPC objective is

$$\mathcal{L}_{\text{CRL}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(E_\theta(s_i, a_i, g_i^+))}{\sum_{j=1}^{B} \exp(E_\theta(s_i, a_i, g_j^+))} + \lambda_{\text{lse}} \frac{1}{B} \sum_i \left(\log \sum_j \exp(E_\theta(s_i, a_i, g_j^+)) \right)^2,$$

where $B = 64$ and $\lambda_{\text{lse}} = 0.01$.

Future goals are sampled from the same episode. For anchor time t, offsets $\Delta \geq 1$ are sampled with weights proportional to

$$\gamma_f^{\Delta-1}, \qquad \gamma_f = 0.99,$$

subject to the remaining episode length.

D.4 IWR Replay Sampling

We calculate the sample weight with $w = \epsilon + \exp(-\frac{1}{2\sigma^2}\|d - c\|_2)$ with parameters in Table A4. Note that the threshold c we select is at least 1.5 times larger than the true distance of factor and target object. We only need a loose heuristic about where the exact interaction will happen.

Table A4: IWR geometry parameters.

Task group	c	$2\sigma^2$
Air Hockey	0.12	30
MetaWorld	0.09	20
Box2D center / goal	3.3	100
Box2D hard / hard-velocity / maze	2.0	80

D.5 Sparse-Reward PPO and SAC Baselines

PPO, SAC, SAC+HER, and SAC+HINT use the same goal-conditioned state and action interfaces as the CRL-family methods. They do not use dense simulator rewards. The reward is the sparse success indicator $r_t = \mathbf{1}\{\text{success at } t\}$

For SAC+HER, each sampled transition uses the original desired goal with probability 0.2 and a future achieved goal with probability 0.8. The reward is recomputed against the relabeled goal.

Table A5: PPO and SAC-family baseline settings.

Algorithm	Setting	Value
PPO	rollout steps	16
PPO	epochs per rollout	4
PPO	clipping parameter	0.2
PPO	GAE λ	0.95
PPO	value loss coefficient	0.5
PPO	max gradient norm	0.5
SAC	target update rate τ	0.005
SAC	entropy coefficient α	0.2
SAC	discount	0.99
SAC	critics	two scalar Q networks, MLP $(512, 512)$
SAC+HER	HER ratio	0.8
SAC+HER	HER goal source	discounted future achieved goals
SAC+HINT	HER mode	interaction-weighted anchor sampling

SAC+HINT keeps the same HER ratio but samples anchors using interaction weights. For HINT, the anchor weight is

$$w_t = \exp(-d_t/\sigma) + 10^{-3},$$

where d_t is the source-target distance.

Table A6: SAC+HINT interaction parameters.

Backend	Contact distance	σ
Air Hockey	0.12	30
Box2D	2.0	80
MetaWorld	0.09	20

D.6 Evaluation Metrics

Let one logging batch contain $N = 256$ completed episodes, matching the number of train environments. For episode e, define

$$o_e = \mathbf{1}\{\text{episode } e \text{ reached success at least once}\},$$
$$z_e = \sum_t \mathbf{1}\{\text{success at step } t\}.$$

For Box2d we report on z_e and for Metaworld we report on o_e. On Table 1, Box2d result is normalized by 100 ticks. Since we repeat the experiment for 5 times, we first record the mean result every 200000 steps as checkpoints and report the best score for comparison.

E Simulation Environment Setup

E.1 Simulation Setting

All environments are wrapped as goal-conditioned vector environments. Each wrapper exposes a state vector s_t, achieved goal g_t^{ach}, desired goal g_t^{des}, action a_t, and boolean success flag. CRL methods use future achieved goals for contrastive relabeling; sparse-reward baselines use the same success flag as reward. Episodes are automatically reset in the vectorized training environment.

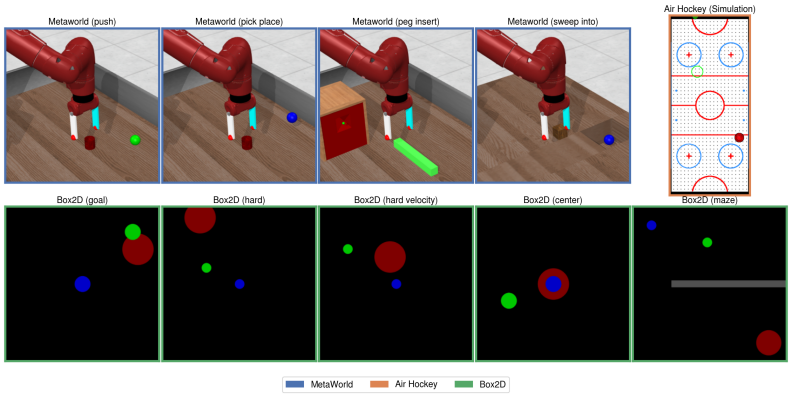


Figure A2: Task sets used in the simulation experiments.

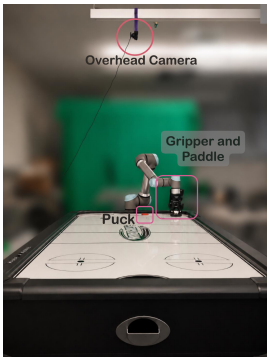


Figure A3: Real-world robotic air hockey testbed.

E.2 Air Hockey

The success radius from the goal state is $r_{\text{goal}} = 0.06$. The paddle, puck, and goal are randomized at reset. The wrapper terminates an episode on success during training. For the default physics tasks, episodes also terminate if the puck hits the bottom or passes the paddle; out-of-bounds termination is disabled. For real-transfer, out-of-bounds termination is enabled.

Table A7: Default Air Hockey simulator parameters.

Parameter	Value
Simulator	Box2D Air Hockey simulator
Table length / width	1.9304 / 0.8636
Render size	360
Paddle radius	0.0508
Puck radius	0.03175
Paddle density / damping	2500 / 3
Puck density / damping	250 / 0.5
Gravity	-0.5
Force scaling	1000
Max force timestep	100
Wall bounce scale	0.02
Blocks / obstacles / targets	0/0/0
Paddles / pucks	1/1
Randomized paddle spawn	true
Goal radius	0.06

E.3 Box2D

Box2D tasks are continuous-control 2D tasks with circular bodies. It has 400 steps per episode and success does not terminate the episode in the Box2D launch. Table A9 shows shared parameters. As visualized in Figure A2, Box2d aims to evaluate how well the agent could control the blue ball to punch the green ball into the goal position, and control it inside the goal position as long as possible. Initialization details could be seen in Table A10.

E.4 MetaWorld

MetaWorld tasks use compact point observations. The state dimension is 7:

$$s = [p_{\text{hand}} \in \mathbb{R}^3,\ q_{\text{gripper}} \in \mathbb{R},\ p_{\text{object}} \in \mathbb{R}^3].$$

Table A8: Air Hockey real-transfer physics and corruption settings.

Parameter	Value
Observation type	history
Domain randomization	enabled
Randomized variables	paddle density, puck damping, gravity
Paddle density range	$[2250, 3750]$
Puck damping range	$[0.1335, 0.2225]$
Gravity range	$[-0.826, -0.496]$
SysID gravity	-0.661
SysID puck damping	0.178
Paddle damping	17
Force scaling	0.99
Max force timestep	10000
Puck density	3000
Paddle / puck radius	0.0508 / 0.03175
Paddle restitution	1.0
Side / end wall restitution	0.99 / 0.7
PID control	enabled
PID gains	$k_p = 9000$, $k_d = 50$, $k_i = 0$
History length	2
Puck observation noise	enabled, std. 0.01
Random occlusion	enabled, start rate 0.05 per step
Occlusion length weights	$[75, 39, 18, 9, 4, 2, 1]$
Observation delay	enabled, 0.025s
Action delay	disabled
Near-paddle puck spawn probability	0.15
Near-paddle spawn offset	$[0.025, 0.05]$ m
Near-paddle spawn speed	$[0.0, 0.2]$ m/s
Linear top spawn speed	$[0.0, 0.5]$ m/s

Table A9: Default Box2D simulator parameters.

Parameter	Value
World length / width	10.0 / 10.0
Action space	continuous
Force scaling	200
Wall bounce factor	0.5
Control damping / density	2.0 / 0.75
Target damping / density	1.0 / 0.1
Ordinary control / target radius	0.5 / 0.5
Hard control / target radius	0.3 / 0.3
Goal success radius	1.0; maze uses 0.8
Render size	80; maze uses 128

Default goal gripper opening is 0.4 and default hand offset is $(0, 0, 0.03)$. The default minimum distance between the initial object and sampled goal is 0.15. Episodes terminate on success during training.

The `peg-insert-side-v2` wrapper maps to the canonical `peg-insert-side-v3` task, uses success scale $(1, 2, 2)$, and uses a goal hand offset $(0.13, 0, 0.03)$. The `sweep-into-v2` success calculation ignores the target z axis for the main distance while enforcing a separate z-axis threshold of 0.02.

Table A10: Box2D tasks initialization description

Paper task name	Reset / goal initialization
Box2D center	random bodies, goal fixed in the center
Box2D goal	random control, target, and goal
Box2D hard	random control, target, and goal
Box2D hard velocity	hard task with target initial velocity 5.0
Box2D maze	control fixed at $(-3.8, -3.8)$, goal fixed at $(3.8, 3.8)$, wall fixed at $(x, y, w, h) = (0, 2.5, 0.22, 5.0)$

Table A11: MetaWorld tasks and success definitions.

Paper task name	Env id	Achieved object	Success condition
MetaWorld push	`push-v2`	object position	distance < 0.05
MetaWorld pick place	`pick-place-v2`	object position	distance < 0.07
MetaWorld peg insert	`peg-insert-side-v2`	peg head position	scaled distance < 0.07
MetaWorld sweep into	`sweep-into-v2`	object position	distance < 0.05 and z error < 0.02

E.5 Post-Hoc Air Hockey Consecutive Evaluation

For the one-minute Air Hockey challenge, evaluation is separate from training. The evaluator disables terminate-on-success so the puck can be re-armed to a new goal after each success. The standard setting is

$$\text{horizon} = 1200 \text{ steps}, \qquad \text{seeds} = 1, \ldots, 16.$$

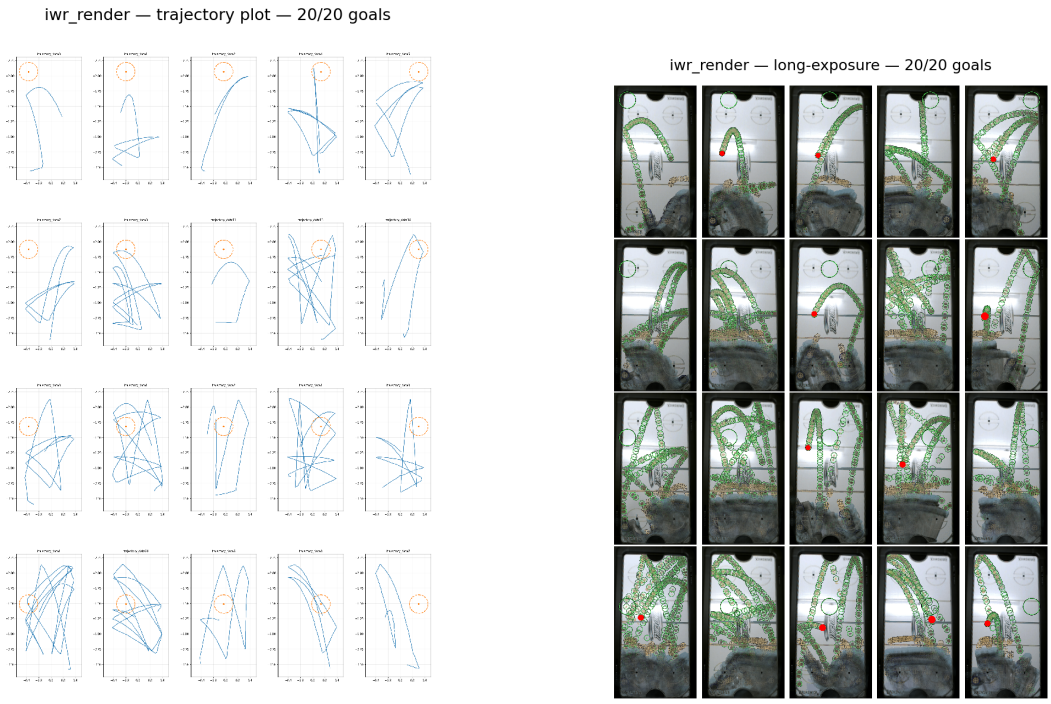
The recorded metrics are complete rate, mean number of consecutive goals, and maximum number of consecutive goals. Rendering is disabled by default for batch evaluation, but the same evaluator can render rollout videos for selected checkpoints.

F Real Setup

The physical setup utilizes a UR5e Universal Robotics robotic arm with a Robotiq two-fingered gripper holding an air hockey paddle. The robot is placed at one end of an inclined air hockey table, such that the puck will always fall back to the robot. By utilizing two 3D-printed curved side pieces, the robot can self-reset by pushing the puck up the side, then striking it. We use two human-demonstrated open-loop policies to perform the puck reset. The puck positions are detected using a 120Hz overhead camera with blob detection and converted with a homography into x,y coordinates. The full control loop runs at 20Hz, using the last 5 puck positions, the paddle's position and velocity, and the goal position, which is the same as in the simulation.

Trajectories are as follows: after the puck successfully crosses the midline of the table, the learned policy begins until the puck falls to a position lower (closer to the base) than the robot. This pattern is akin to juggling, except that the task requires putting the radius of the puck inside the radius of the goal position (success), before the puck falls past the paddle (failure). Evaluation utilizes a fixed set of 20 goals in a 4×5 pattern, with 5 goals per row and 4 rows getting progressively lower. Additional details, including the dimensions of the table, paddle, puck, etc., can be found in Chuck et al. [13].

To transfer trained policies from the simulator to the real robot, we first identified system parameters for the paddle using a small number (2 min) of human-generated paddle data. Puck parameters are then estimated based on human juggling data and the physics of the puck, as this was superior to



(a) Trajectory-space visualization.

(b) Long-exposure real-world rollout.

Figure A4: IWR evaluation rollouts across 20/20 achieved goals.

black box optimization for estimating these parameters. Consequently, a meaningful sim-to-real gap exists in the puck movement between the real and simulated environments. To train policies robust to this gap, we leverage domain randomization in the style of Tobin et al. [73], Peng et al. [60], Kumar et al. [44]. For system parameters, we randomize the paddle density (which affects collisions), puck damping, and gravity as follows:

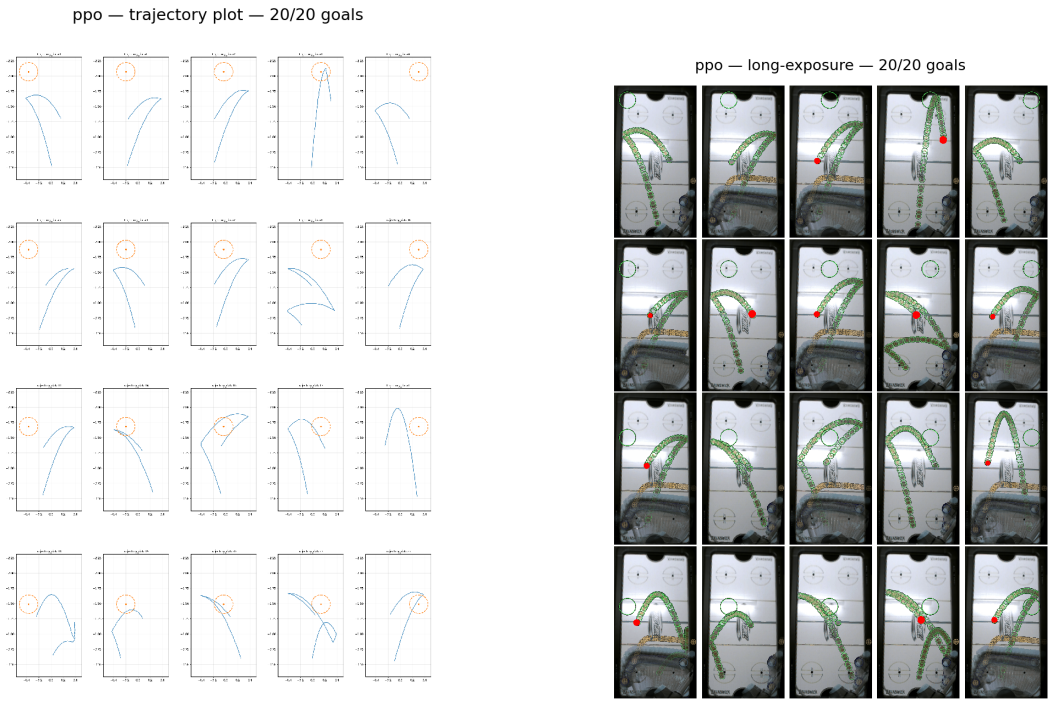
Parameter	**Range**	**SysID**	**Variation**
Paddle density	$[2250.0,\ 3750.0]$	3000	$\pm 25\%$
Puck damping	$[0.1335,\ 0.2225]$	0.178	$\pm 25\%$
Gravity	$[-0.826,\ -0.496]$	-0.661	$\pm 25\%$

We also perform substantial random resets for the puck reset, resetting the puck at positions across the top and middle with randomly selected velocities, as well as random positions near the paddle, to simulate the variance and challenging positions from where a trajectory might start. The puck is also spawned in random positions across its workspace. Random Noise and occlusions (missing values) are added to the observation of the puck position to force the policy to be robust to perceptual noise, as well as adding perceptual delay, where the puck position is lagged when given to the agent. Finally, 4-step smoothing is applied to the real robot, and the movement of the paddle in simulation is tuned with a PID controller to match this kind of smooth motion.

Trajectory visualization could be found in: IWR: Figure A4, PPO: Figure A5, SAC: Figure A6, SAC+HER: Figure A7, SAC+HINT: Figure A8, CRL: Figure A9, CRTR: Figure A10.

G Energy Visualization

Figure A11 shows the t-SNE visualization for learned $\phi(s, a)$. On locomotion tasks, e.g., Meta-world(Reach), the trajectory of latent space is smooth, while in manipulation tasks Air Hockey, the

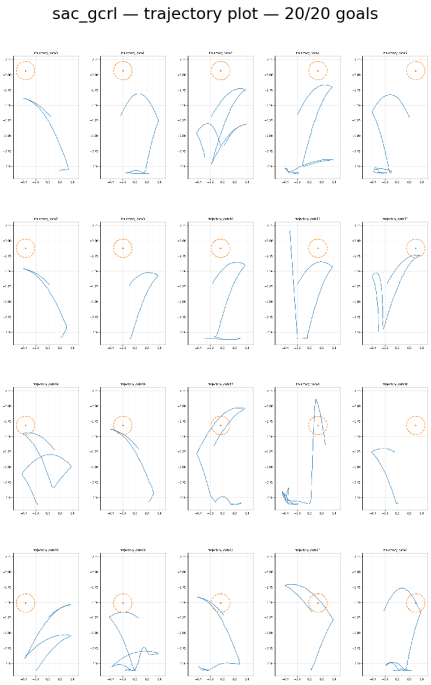


(a) Trajectory-space visualization.

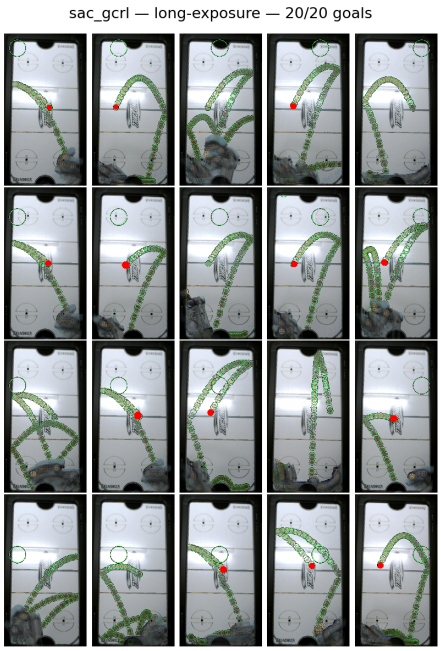
(b) Long-exposure real-world rollout.

Figure A5: PPO evaluation rollouts across 20/20 achieved goals.

trajectory jumps whenever the interaction happens, which is indicated with a red circle. This result supports our analysis of mode-switching challenge of CRL.

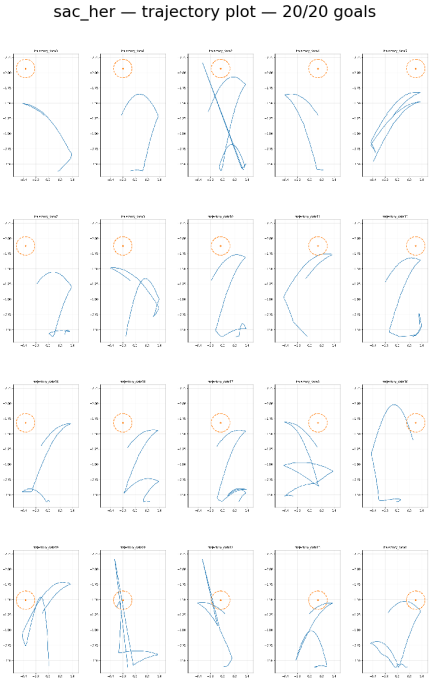


(a) Trajectory-space visualization.

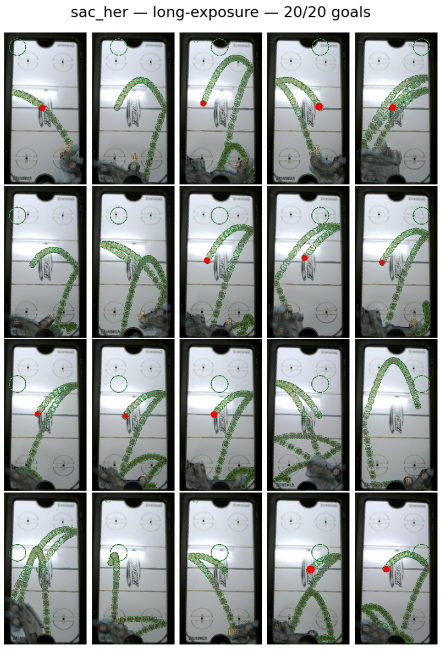


(b) Long-exposure real-world rollout.

Figure A6: SAC-GCRL evaluation rollouts across 20/20 achieved goals.

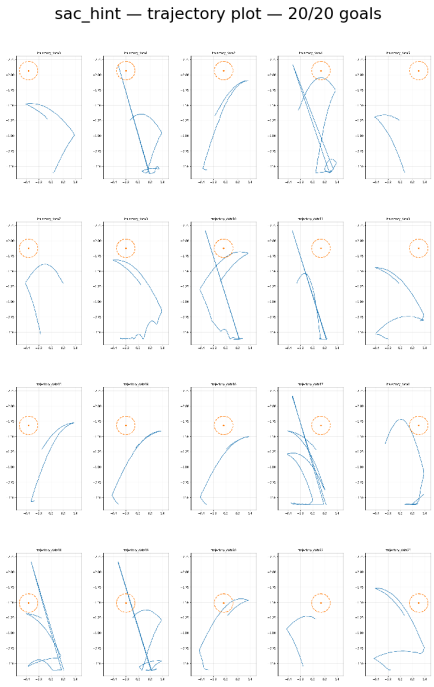


(a) Trajectory-space visualization.

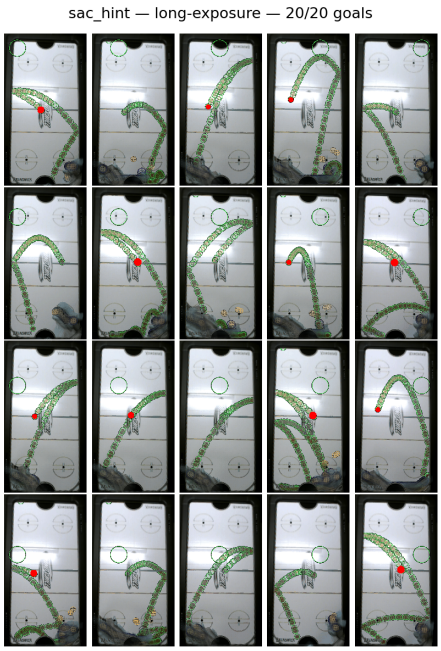


(b) Long-exposure real-world rollout.

Figure A7: SAC-HER evaluation rollouts across 20/20 achieved goals.

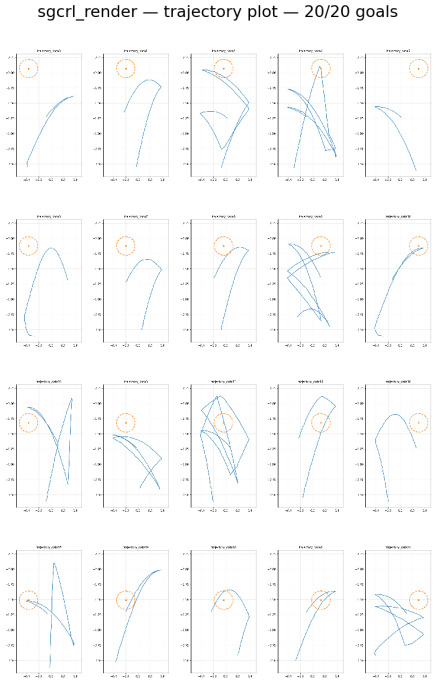


(a) Trajectory-space visualization.

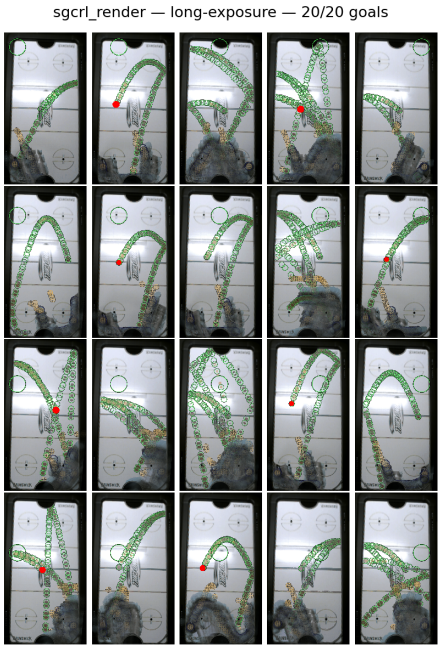


(b) Long-exposure real-world rollout.

Figure A8: SAC-Hint evaluation rollouts across 20/20 achieved goals.

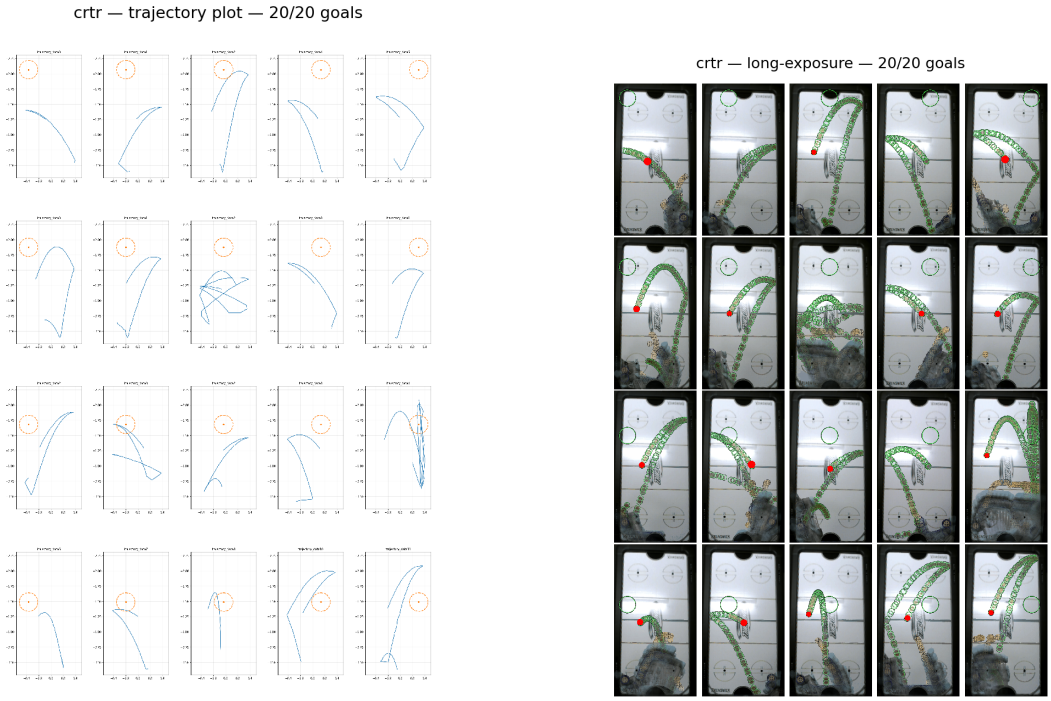


(a) Trajectory-space visualization.



(b) Long-exposure real-world rollout.

Figure A9: CRL evaluation rollouts across 20/20 achieved goals.



(a) Trajectory-space visualization.

(b) Long-exposure real-world rollout.

Figure A10: CRTR evaluation rollouts across 20/20 achieved goals.

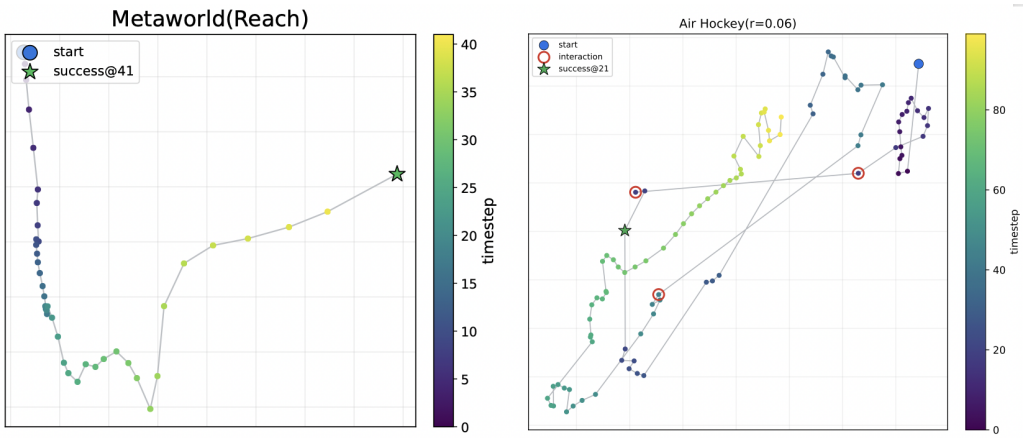


Figure A11: t-SNE visualization of $\phi(s, a)$ of CRL. Left: Metaworld(reach) Right: Air Hockey Simulator. Red circle indicates interactions. Star indicates success.