
IN-RIL: INTERLEAVED REINFORCEMENT AND IMITATION LEARNING FOR POLICY FINE-TUNING

Dechen Gao¹, Hang Wang², Hanchu Zhou², Nejib Ammar³, Shatadal Mishra³,

Ahmadreza Moradipari³, Iman Soltani⁴, and Junshan Zhang²

¹Department of Computer Science, University of California, Davis

²Department of Electrical and Computer Engineering, University of California, Davis

³Toyota InfoTech Labs, Mountain View, CA

⁴Department of Mechanical and Aerospace Engineering, University of California, Davis

{dcgao, whang, hczhou, isoltani, jazh}@ucdavis.edu

{nejib.ammar, shatadal.mishra, ahmadreza.moradipari}@toyota.com

June 12, 2025

ABSTRACT

Imitation learning (IL) and reinforcement learning (RL) each offer distinct advantages for robotics policy learning: IL provides stable learning from demonstrations, and RL promotes generalization through exploration. While existing robot learning approaches using IL-based pre-training followed by RL-based fine-tuning are promising, this two-step learning paradigm often suffers from instability and poor sample efficiency during the RL fine-tuning phase. In this work, we introduce IN-RIL, INterleaved Reinforcement learning and Imitation Learning, for policy fine-tuning, which periodically injects IL updates after multiple RL updates and hence can benefit from the stability of IL and the guidance of expert data for more efficient exploration throughout the entire fine-tuning process. Since IL and RL involve different optimization objectives, we develop gradient separation mechanisms to prevent destructive interference during IN-RIL fine-tuning, by separating possibly conflicting gradient updates in orthogonal subspaces. Furthermore, we conduct rigorous analysis, and our findings shed light on why interleaving IL with RL stabilizes learning and improves sample-efficiency. Extensive experiments on 14 robot manipulation and locomotion tasks across 3 benchmarks, including FurnitureBench, OpenAI Gym, and Robomimic, demonstrate that IN-RIL can significantly improve sample efficiency and mitigate performance collapse during online finetuning in both long- and short-horizon tasks with either sparse or dense rewards. IN-RIL, as a general plug-in compatible with various state-of-the-art RL algorithms, can significantly improve RL fine-tuning, e.g., from 12% to 88% with 6.3x improvement in the success rate on Robomimic Transport. Project page: <https://github.com/ucd-dare/IN-RIL>.

Keywords Imitation Learning · Reinforcement Learning · Robotics Manipulation

1 Introduction

Recent advances in robotics policy learning have largely been driven by imitation learning (IL) and reinforcement learning (RL) [1, 2, 3, 4]. These two approaches offer complementary strengths for robot learning, yet each comes with limitations when used in isolation. More specifically, in IL (such as behavioral cloning [5, 6]), an agent learns a policy to mimic expert demonstrations, using supervised learning. It is known that while IL provides stable learning dynamics, it faces three critical challenges: the high cost of collecting expert demonstrations [7], limited generalization beyond the demonstration distribution, and vulnerability to compounding errors [8, 9]. Even small deviations from the

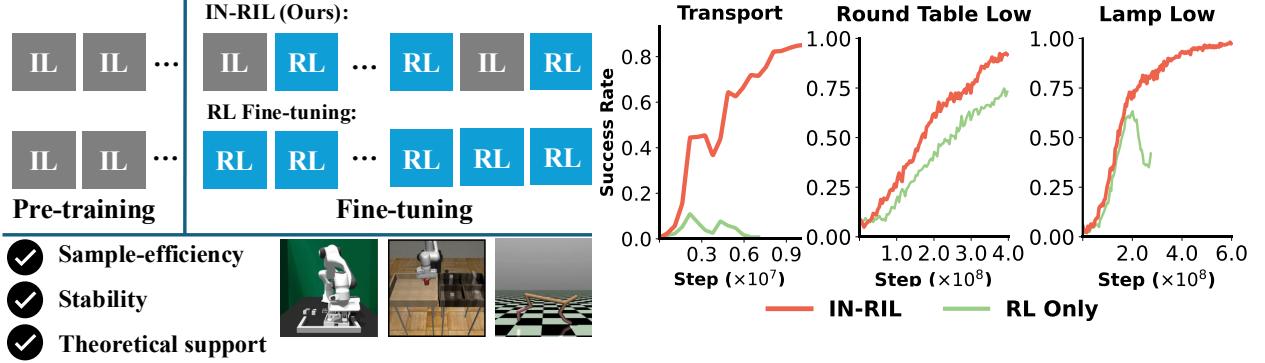


Figure 1: Comparison between IN-RIL (interleaved RL/IL) fine-tuning and RL fine-tuning on Transport, Round-Table, and Lamp, which are challenging multi-stage and sparse-reward tasks. Extensive experiments show that IL benefits from expert demonstrations but performance saturates at low success rates; and RL fine-tuning can suffer from stability and poor sample efficiency. IN-RIL fine-tuning succeeds to learn and outperforms RL fine-tuning by a significant margin in all tasks.

demonstration distribution could accumulate and drastically degrade the performance. RL approaches, in contrast, learn policies through environmental interaction to maximize accumulated rewards in a Markov Decision Process (MDP) [10]. Many empirical studies have shown that the RL approach enables active exploration beyond expert knowledge but often suffers from instability, sample inefficiency, and hypersensitivity to parameter choices. In particular, these problems are amplified in robotics tasks with sparse rewards and long horizons. For instance, as shown in Figure 1, the IL method alone yields poor performance due to the inherent limited coverage of demonstrations, whereas the RL method struggles to learn effectively through random exploration alone.

To address the above challenges, recent studies [9, 11, 12, 13] have proposed hybrid approaches that combine IL-based initialization with subsequent RL fine-tuning. While this paradigm leverages the unique strengths of both methods, the critical fine-tuning stage using RL alone continues to face significant challenges that limit its effectiveness. Specifically, RL fine-tuning often suffers from performance collapse, instability, and poor sample efficiency [14, 12, 8]. Existing approaches may improve fine-tuning by adding demonstrations into replay buffers [15, 16], which requires reward annotations and complex sampling strategies [15, 17], or add regularization terms to constrain policy drift [8, 18], which demands careful hyperparameter tuning. These limitations presents a fundamental question that we aim to address in this work:

*How to synergize the stability of IL with the exploration benefits of RL
for efficient policy fine-tuning?*

Thus motivated, we propose IN-RIL (INterleaved Reinforcement and Imitation Learning) fine-tuning that can cleverly exploit demonstration data throughout the fine-tuning process. As illustrated in Figure 2, IN-RIL integrates IL updates with RL fine-tuning by periodically inserting one IL update after every few RL updates. As shown clearly in Figure 1, IN-RIL fine-tuning outperforms RL fine-tuning by a significant margin in the challenging long-horizon and sparse-reward tasks. We summarize our key insight for IN-RIL as follows.

As illustrated in Figure 3, IL and RL objectives create different non-convex optimization landscapes, which are often not aligned. **Both IL and RL have multiple local minima/optima, indicating that when fine-tuning using RL or IL alone could be trapped at a local minimum.** By interleaving IL and RL updates during fine-tuning, IN-RIL can help RL to jump out of a lower reward neighborhood towards a higher reward neighborhood, and in the meanwhile RL updates can help to move IL out its local minima in its loss landscape to another local minima with lower losses.

Given that IL and RL involve different optimization landscapes, we caution that it is of critical importance to avoid destructive interference between their respective gradient updates in IN-RIL. To address this challenge, we devise

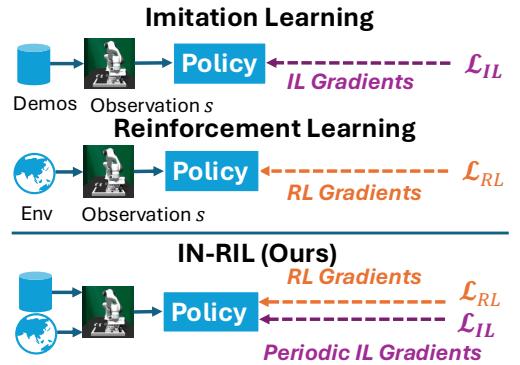


Figure 2: An illustration of IN-RIL which updates the policy network with both IL and RL objectives.

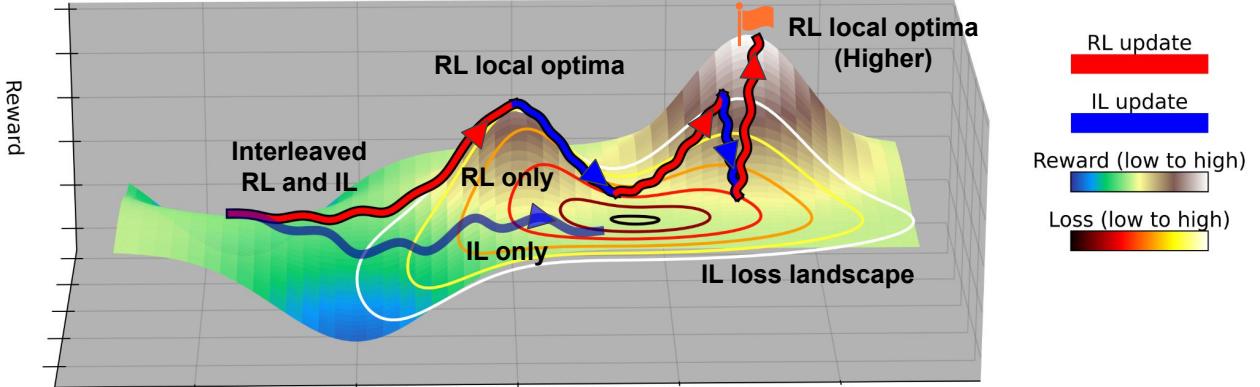


Figure 3: Optimization landscapes for IN-RIL. The IL loss landscape, represented by the 3D surface topology and its corresponding contour lines (where each contour connects points of equal IL loss value); and the landscape of RL rewards (or negative of the loss), represented by the color gradient mapped onto the surface (where the blue-to-white spectrum indicates low-to-high reward values as shown in the legend). IL updates drive the policy toward regions with lower losses, while RL updates steer toward higher rewards. Both optimization processes are stochastic and non-convex with multiple local optima. When using either RL or IL alone, training often converges to suboptimal solutions (as shown in the “IL only” and “RL only” trajectories). In contrast, our IN-RIL approach enables each objective to help escape the other’s local optima: periodic IL updates help RL escape lower-reward regions toward higher-reward neighborhoods, while RL updates help IL traverse between different local minima in the loss landscape.

gradient separation mechanisms that effectively combine learning signals while preventing conflicts between these different objectives. In particular, we have developed two implementation approaches: (1) gradient surgery [19, 20], which mitigates interference through gradient projection techniques; and (2) network separation, which isolates RL gradients in a residual policy while the base policy continues to leverage IL. Both methods effectively separate IL and RL gradient updates in different subspaces to prevent destructive interactions. It is worth noting that IN-RIL is algorithm-agnostic and can serve as a plug-in to existing RL frameworks, as demonstrated through our integration with state-of-the-art methods including DPPO [11], IDQL [21], residual PPO [9, 13], covering both on-policy and off-policy approaches.

Summary of Contributions In summary, our work makes the following contributions:

- **IN-RIL.** We introduce IN-RIL, a fine-tuning approach that periodically interleaves imitation learning updates with reinforcement learning updates, addressing the limitations of conventional two-step methods. Intuitively, by periodically inserting one IL iteration after every few RL iterations, IN-RIL synergizes the stability of IL using expert demonstrations with the exploration capabilities of RL throughout the fine-tuning process.
- **Gradient Separation Mechanisms.** Given that IL and RL involve different optimization objectives, we develop gradient separation mechanisms to prevent destructive interference during interleaved training. Our methods effectively separate possibly conflicting gradient updates in orthogonal subspaces, reaping the benefits of both approaches while minimizing conflicts during fine-tuning. More specifically, we propose two separation techniques: 1) gradient surgery, where RL and IL gradients are projected into independent subspaces to mitigate conflicts; 2) network separation, where IN-RIL introduces a residual policy network updated by RL gradients, while base policy is not updated by RL gradients, and therefore, avoids the conflicts.
- **Analytic Foundation.** We carry out analysis to characterize the foundational reason why IN-RIL outperforms conventional RL fine-tuning in both stability and sample efficiency, offering insights into the optimal interleaving ratio for maximizing performance across diverse robotics tasks.
- **Algorithm-Agnostic Design with Comprehensive Validation.** We demonstrate IN-RIL’s effectiveness as a general plug-in compatible with state-of-the-art RL algorithms, including on-policy methods (DPPO, residual PPO) and off-policy approaches (IDQL). Through extensive experiments on 14 challenging robotics tasks across FurnitureBench [22], Robomimic [23], and OpenAI Gym [24], we show that IN-RIL can substantially improve performance — e.g., boosting success rates to 88%, when integrated with RL algorithms that originally yield only 12% success rates on Robomimic Transport. Our evaluations span both long-horizon and short-horizon scenarios with sparse and dense rewards, demonstrating the broad applicability of our approach.

2 Related Work

Robotics Policy Learning and Fine-Tuning. Imitation learning (IL) [25, 26, 1, 2, 27, 6] and reinforcement learning (RL) [28, 29, 30, 11, 4, 31, 9] have been widely studied in robotics. IL assumes access to expert demonstrations and is generally more stable to train [1, 5], but it suffers from distribution shifts and often fails to generalize beyond demonstrations [8]. In addition, collecting high-quality expert data can be labor-intensive and costly, sometimes requiring hundreds or even thousands of demonstrations per task [7] through teleoperation [2], or VR equipments [32]. On the other hand, RL enables agents to explore and self-improve, potentially overcoming IL limitations of labor-intensive data collection and generalization. However, RL is notoriously sample-inefficient [16], especially for long-horizon tasks with sparse rewards [33], where agents may easily fail to explore and learn. Recent works have proposed combining IL and RL in a two-stage pipeline: IL is first used to pre-train a reasonable policy to warm-start the RL process, followed by RL fine-tuning to further improve generalization via exploration [11, 9, 17]. The same paradigm was also applied to LLM fine-tuning [34]. In this work, we move beyond the two-stage paradigm, and show that the data used for pre-training, even after pre-training plateaus, is still valuable in improving sample-efficiency and stability of RL fine-tuning.

RL with Expert Demonstrations. Recent works have explored leveraging offline data for training RL policies. ROT [18, 8] introduces a regularization term to RL objectives to keep the policy close to expert behaviors, which, however, requires careful balancing between RL objectives and the regularization term. AWAC [12], Hy-Q [16], IBRL [17], RLPD [15], Cal-QL [14] add expert data with rewards to a replay buffer and perform off-policy updates during online learning. However, it can be infeasible to perform off-policy RL updates on expert demonstrations since reward annotations are not always available. Furthermore, sampling strategy is shown to be crucial for off-policy updates when there are both demonstration data and RL-collected data [17, 15]. In contrast, IN-RIL does not introduce explicit regularization terms which rely on delicate loss balancing, and can over-regularize the policy and damage performance. IN-RIL does not assume availability of rewards in IL data, or require sampling strategies to balance learning from offline and online data. Instead, it treats IL and RL as complementary optimization processes and interleaves them during fine-tuning without modifying the RL algorithm itself. This makes IN-RIL broadly applicable to both on-policy and off-policy RL methods.

3 IN-RIL: Interleaved RL and IL for Efficient Policy Finetuning

In this section, we provide a theoretical analysis of IN-RIL, aiming to answer two key questions: (1) what is the optimal interleaving ratio of RL updates to IL updates that balances learning stability and performance improvement, and (2) How much reduction in iteration complexity can be achieved by our proposed IN-RIL approach? We derive conditions under which IN-RIL achieves superior sample efficiency and faster convergence to target performance levels. These theoretical results not only justify our algorithmic design choices but also provide practical guidance for adapting the interleaving ratio based on gradient alignment during training.

Markov Decision Process. We consider a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and ρ_0 is the initial state distribution. A policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ maps states to probability distributions over actions. The action-value function, or Q-function, for a policy π is defined as $Q^\pi(s, a) = \mathbb{E}_\pi [\sum t = 0^\infty \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a]$, representing the expected cumulative discounted reward when taking action a in state s and following policy π thereafter. The objective in RL is to find a policy that maximizes the expected Q-value: $\mathbb{E}_{s \sim \rho_0, a \sim \pi(\cdot|s)} [Q^\pi(s, a)]$.

Pre-Training. We consider a parametric policy $\pi_\theta : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ that maps states to distributions over actions. We employ a direct policy representation where $\pi_\theta(a|s)$ gives the probability (or probability density) of taking action a in state s . This formulation allows for direct optimization through gradient-based methods while maintaining sufficient expressivity for complex robotic control tasks. During pre-training, we use behavior cloning to learn a policy that imitates expert demonstrations $\mathcal{D}_{\text{exp}} = \{\tau_1, \tau_2, \dots, \tau_N\}$, where each trajectory $\tau_i = \{(s_1, a_1), \dots, (s_T, a_T)\}$ contains state-action pairs. The objective is to maximize the likelihood of expert actions given the corresponding states:

$$\mathcal{L}_{\text{IL}}(\theta) = \mathbb{E}_{(s, a) \sim \mathcal{D}_{\text{exp}}} [-\log \pi_\theta(a|s)], \quad (1)$$

where a^* represents the expert action. This negative log-likelihood objective encourages the policy to assign high probability to actions demonstrated by experts in the same states. We then obtain a warm-start policy $\pi_0 = \arg \min_{\pi_\theta} \mathcal{L}_{\text{IL}}(\theta)$ that serves as the initialization for subsequent fine-tuning. This pre-training approach allows the policy to capture the basic structure of the task before reinforcement learning is applied to further optimize performance. After obtaining a

policy via imitation learning during pre-training, we proceed to the finetuning phase where we optimize the policy. In our analysis, we compare two distinct finetuning approaches, RL Finetuning and our proposed IN-RIL.

RL Finetuning. After pretraining, RL finetuning directly optimizes policy parameters to maximize the expected Q-value as defined earlier, through gradient updates of the form:

$$\theta_{t+1} = \theta_t - \alpha_{\text{RL}} \nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta_t)$$

where α_{RL} is the learning rate and $\mathcal{L}_{\text{RL}}(\theta) = -\mathbb{E}_{s \sim d^{\pi_\theta}}[Q^{\pi_\theta}(s, \pi_\theta(s))]$ is defined as the loss function, which represents the negative of the expected Q-value under the current policy's state distribution d^{π_θ} . This formulation directly connects to our optimization objective of maximizing $\mathbb{E}_{s \sim \rho_0, a \sim \pi(\cdot|s)}[Q^\pi(s, a)]$, but accounts for the evolving state distribution as the policy improves. While this approach aims to maximize the overall reward, it often suffers from instability and poor sample efficiency, particularly when finetuning complex models like diffusion policies.

IN-RIL. As depicted in Figure 1, the proposed IN-RIL systematically alternates between IL and RL updates:

$$\begin{aligned} \theta_{t+\frac{1}{1+m(t)}} &= \theta_t - \alpha_{\text{IL}} \nabla_{\theta} \mathcal{L}_{\text{IL}}(\theta_t) \\ \theta_{t+\frac{1+j}{1+m(t)}} &= \theta_{t+\frac{j}{1+m(t)}} - \alpha_{\text{RL}} \nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta_{t+\frac{j}{1+m(t)}}), \quad j \in \{1, \dots, m(t)\} \end{aligned}$$

where $m(t)$ represents the iteration-dependent number of RL updates performed after each IL update. The IL updates help maintain the desirable behaviors from pre-training while providing regularization, and the RL updates improve performance on the target task.

Our analysis uses standard assumptions regarding the pretraining performance, data coverage, smoothness properties of the loss functions, and gradient estimation quality. Specifically, we assume that: (1) the initial policy obtained by pretraining results in a training loss within a bounded distance from the IL objective; (2) the expert demonstration dataset provides reasonably sufficient coverage of the relevant state space for the target task; (3) both the IL and RL objectives satisfy smoothness conditions; and (4) the stochastic gradient estimates for both objectives have bounded variance that decreases proportionally with batch size. The formal statements of these assumptions (Assumptions 2-5) and their implications are provided in Appendix A.

Next, we introduce the assumptions on the geometric relationship between the gradients of the IL and RL objectives in ?? 1. In particular, we use the parameter $\rho(t)$ to capture the cosine similarity between these gradients, with positive values indicating opposing gradients and negative values indicating aligned gradients. Such assumption has been commonly used in multi-objective optimization [19, 35].

Assumption 1 (Gradient Relationship). *In the finetuning regime, the gradients of IL and RL objectives exhibit the following relationship:*

$$\langle \nabla_{\theta} \mathcal{L}_{\text{IL}}(\theta_t), \nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta_t) \rangle = -\rho(t) \|\nabla_{\theta} \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta_t)\|$$

where $\rho(t) \in [-1, 1]$ represents the time-varying relationship between gradients, with positive values indicating opposition (negative cosine similarity) and negative values indicating alignment (positive cosine similarity).

Based on these assumptions, we establish the following key results on the optimal ratio of RL updates to IL updates in the proposed IN-RIL. This ratio is crucial for balancing the stability provided by IL updates with the performance improvements offered by RL updates.

Theorem 1 (Optimal Interleaving Ratio). *Under Assumptions 1-5, at iteration t , the optimal ratio $m(t)$ for IN-RIL satisfies $m_{\text{opt}}(t) \geq 1$.*

Theorem 1 provides a principled formula for adapting the interleaving ratio throughout training based on current gradient information. The optimal ratio $m_{\text{opt}}(t)$ increases when gradients strongly “oppose” each other ($\rho(t) > 0$) and decreases when they are more aligned ($\rho(t) < 0$), reflecting the intuition that more RL updates are needed to make progress when IL updates work against the RL objective. This result suggests that monitoring gradient alignment during training can lead to more efficient optimization strategies compared to using a fixed interleaving ratio. Given this optimal ratio, we next quantify exactly how much more efficient IN-RIL can be compared to RL-only approaches.

Denote $\Delta_{\text{IL-RL}} = -\sum_{t=0}^{T-1} \frac{c_{\text{IL}}^2 \rho(t)}{L_{\text{IL}}} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| - \frac{c_{\text{IL}}^2 \sigma_{\text{IL}}^2 T}{2L_{\text{IL}} N_{\text{IL}}}$. Then we have:

Theorem 2 (Iteration Complexity of IN-RIL). *Under Assumptions 1-5, for a fixed computational budget of T total updates, IN-RIL with $m > 1$ and $\Delta_{\text{IL-RL}} > \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{m+1}$ requires fewer iterations to reach a target accuracy ϵ than RL-only finetuning, i.e., $\frac{T_{\text{RL-only}}}{T_{\text{IN-RIL}}} > 1$.*

Theorem 2 establishes the conditions under which IN-RIL achieves superior efficiency compared to RL-only finetuning. Specifically, when the regularization benefit Δ_{IL-RL} exceeds the threshold $\frac{L_{RL}(\mathcal{L}_{RL}(\theta_0) - \mathcal{L}_{RL}^*)}{m+1}$, IN-RIL requires fewer total updates to reach the same performance level. This threshold depends critically on the interleaving ratio m , with higher values of m reducing the required regularization benefit for efficiency gain. Intuitively, this means that when the stabilizing effect of periodically revisiting the demonstration data is sufficiently strong, and the interleaving ratio is properly set, IN-RIL can achieve the same performance with fewer total updates. This theoretical guarantee aligns with our empirical observations across multiple robotics tasks, where IN-RIL consistently demonstrates faster convergence and higher sample efficiency than pure RL approaches. The result provides formal justification for the IN-RIL and offers practical guidance for setting the interleaving ratio based on task characteristics.

4 Experiments

Based on the above analysis, we further conduct a comprehensive empirical evaluation to address two key questions: 1) What are the benefits of IN-RIL compared to RL fine-tuning? 2) What is the impact of the interleaving ratio m on the performance? To this end, we evaluate IN-RIL on 14 different tasks across three widely adopted robotics benchmarks, including FurnitureBench [22], OpenAI Gym [24], and Robomimic [23]. These benchmarks represent a diverse spectrum of robotics challenges, encompassing both locomotion and manipulation tasks with varying reward structures (sparse and dense) and time horizons (short and long).

Robomimic [23]. We evaluate IN-RIL on four robot manipulation tasks from Robomimic: Lift, Can, Square, and Transport. Among these, Square and Transport are particularly challenging for RL agents [11]. All tasks feature sparse rewards upon successful completion, with each task providing 300 demonstrations. For Transport and Lift, we specifically use noisy multi-human demonstration data to test robustness. Notably, when coupled with IN-RIL, IDQL, one of the best off-policy fine-tuning algorithms, achieves only 12% success rates on Transport, while IN-RIL boosts it to 88%, a $6.3\times$ improvement.

FurnitureBench [22]. FurnitureBench presents the most challenging tasks in our experiments, featuring long-horizon, multi-stage manipulation tasks with sparse rewards. We include three assembly tasks: One-Leg, Lamp, and Round-Table, each with both Low and Med randomness settings for state distributions. Each task includes 50 human demonstrations and provides sparse stage-completion rewards. We additionally incorporate two tasks from ResiP [9]: Mug-Rack and Peg-in-Hole, resulting in a total of 7 tasks when accounting for randomness variants.

OpenAI Gym [24]. To evaluate performance on dense-reward tasks, we include three classic locomotion benchmarks: Hopper (v2), Walker2D (v2), and HalfCheetah (v2). For these tasks, we utilize the medium-level imitation datasets from D4RL [36].

4.1 Training

We evaluate IN-RIL with multiple policy parameterizations for pre-training, including diffusion policy (DP)[1] and Gaussian policy[10], both of which are widely adopted in recent IL and RL literature [1, 7, 11, 9]. Particularly, DP has consistently demonstrated superior performance across robotics tasks in both pre-training [1] (see Table 1) and fine-tuning [11]. We employ action chunking [2] to enhance temporal consistency. For fine-tuning, we select three state-of-the-art RL algorithms spanning both on-policy and off-policy approaches: 1) PPO [37, 9, 13], a widely used on-policy algorithm; 2) DPPO [11], an on-policy, policy gradient-based RL algorithm; and 3) IDQL [6], an off-policy, Q-learning-based RL algorithm. DPPO and IDQL are both DP-based RL algorithms. This diverse selection enables us to comprehensively evaluate IN-RIL’s effectiveness across different RL algorithms and policy parameterizations.

Pre-Training. Taking FurnitureBench as an example, we pre-train different policy parameterizations using 50 demonstrations with IL until convergence. As shown in Table 1, Gaussian policy without action chunking fails entirely on these challenging multi-stage sparse-reward tasks, while Gaussian policy with action chunking achieves limited success. DP demonstrates the strongest overall performance across all tasks in FurnitureBench, Robomimic, and Gym. However, even DP pre-training remains sub-optimal, with 3 tasks showing below 5% success rates after loss plateau, primarily due to limited dataset coverage.

Fine-Tuning. While DP yields the best pre-training performance, fine-tuning DP with conventional RL algorithms presents significant challenges and can lead to failure [11, 38]. We consider two strategies for RL fine-tuning: 1) *Full network fine-tuning*, where we use specialized RL algorithms (DPPO and IDQL) to fine-tune the entire pre-trained DP network; and 2) *Residual policy fine-tuning*, where we introduce an additional Gaussian policy as a residual policy on top of the pre-trained DP (base) policy. The residual policy, implemented as an MLP network, is fine-tuned with

Policy Parameterization		OneLeg		Lamp		RoundTable		MugRack	PegInHole
		Low	Med	Low	Med				
BC	Gaussian w/ Action Chunking	0.38	0.17	0.07	0.02	0.01		0.14	0.02
	Gaussian w/o Action Chunking	0.0	0.0	0.0	0.0	0.0		0.0	0.0
	DP	0.47	0.28	0.05	0.1	0.10		0.19	3

Table 1: Success rates across FurnitureBench tasks [9, 22] using pre-trained policies.

conventional RL (PPO) [37, 9] while the base policy is updated solely with IL. The residual policy learns to adjust the base policy’s actions at each time step. For each task, we fine-tune the pre-trained DP checkpoint with the highest success rate (or reward) using IN-RIL, and compare against RL-only fine-tuning. While our theory suggests an adaptive ratio $m(t)$, we use a constant value of m throughout training for simplicity. Based on our results, values of m between 5 and 15 work well across most tasks, balancing performance improvement with policy stability. We conduct a detailed ablation study on the impact of different m values in Section 4.3.

Separation of RL and IL gradients for IN-RIL. RL and IL each operate within distinct optimization landscapes, meaning a policy that is optimal from an RL perspective (high rewards) may not be optimal from an IL perspective (low BC losses), and vice versa. Directly updating a single network with these potentially conflicting objectives can degrade policy performance (as demonstrated in our ablation study in Section 4.4).

To address this challenge, as illustrated in Figure 4, we introduce two gradient separation techniques that prevent interference between RL and IL objectives. The first technique, 1) *gradient surgery*, projects each gradient onto the dual cone [20], ensuring that updates benefit both individual objectives. The second technique, 2) *network separation*, is naturally integrated with the residual RL fine-tuning strategy. This approach allocates IL gradients to the base policy while RL gradients update the residual policy, effectively mitigating interference.

4.2 IN-RIL vs. RL Fine-tuning

We demonstrate that IN-RIL can enhance the performance of state-of-the-art RL fine-tuning algorithms across diverse robotic tasks. For each benchmark, we select the best-performing RL algorithms according to recent literature: DPPO [11] and IDQL [21] for Robomimic and Gym tasks, and residual PPO [9] for FurnitureBench. Our comprehensive evaluation reveals that IN-RIL consistently outperforms these top-performing algorithms in terms of sample efficiency, stability, and final performance. The results for Robomimic and Gym tasks using DPPO and IDQL are presented in Figure 5 and Figure 6, respectively. FurnitureBench results are shown in Figure 7.

We also compare IN-RIL with other RL fine-tuning algorithms in Table 2 and Table 3. The other baselines include DPPO augmented by BC loss regularization [8] (denoted as “BC Loss” in the table), AWC [39, 11], and DIPO [38].

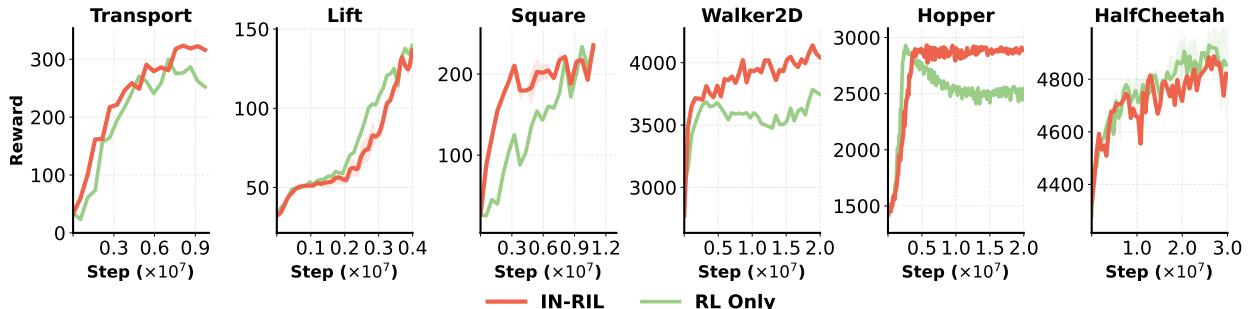


Figure 5: Comparing IN-RIL with RL fine-tuning on Robomimic and Gym using DPPO.

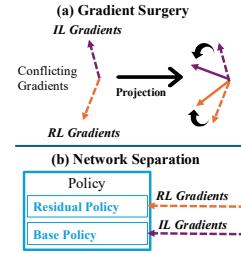


Figure 4: An illustration of the two gradient separation mechanisms: a) gradient surgery, and b) network separation.

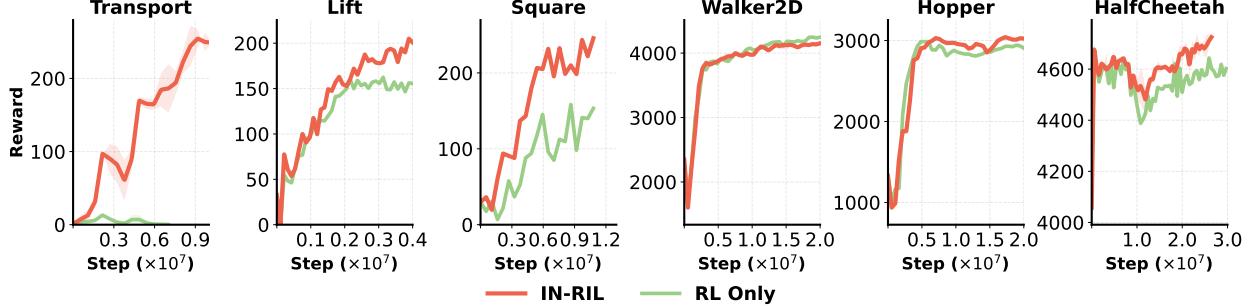


Figure 6: Comparing IN-RIL with RL fine-tuning on Robomimic and Gym using IDQL.

Task	IN-RIL (DPPO)	DPPO	IN-RIL (IDQL)	IDQL	BC Loss	DIPO	AWR
Transport	0.95	0.89	0.88	0.12	0.41	0.16	0.16
Can	1.00	1.00	0.98	1.00	0.96	0.94	0.65
Lift	1.00	1.00	1.00	1.00	0.98	0.97	0.99
Square	0.91	0.90	0.98	0.80	0.64	0.59	0.51
Walker2D	4139	3786	4186	4248	3457	3715	4250
Hopper	2930	2929	3042	2988	2896	2938	1427
HalfCheetah	4887	5011	4742	4671	4532	4644	4611

Table 2: Performance comparison for all fine-tuning methods on Robomimic (using success rates) and Gym tasks (using rewards). Bold values indicate the best in the DPPO group, or IDQL group. Italic values indicate the overall best across all methods.

Figure 5 and Figure 6 show that IN-RIL consistently improves upon both DPPO and IDQL across manipulation and locomotion tasks. Notably, on the two most challenging Robomimic tasks, Transport and Square [11], IN-RIL substantially boosts performance of both DPPO and IDQL. The gains are especially prominent when combined with IDQL, where RL-only fine-tuning fails on Transport with 12% success rates, while IN-RIL successfully solves the task and achieves 88% success rates, as shown in Figure 6 and Table 2; on Square, IN-RIL improves IDQL by 22.5% in success rates; and reduces 62% environment steps needed for DPPO to converge in Figure 5. This highlights the crucial role of IL guidance for RL exploration. For Gym locomotion tasks, IN-RIL either matches or surpasses RL-only fine-tuning. In Figure 5, DPPO degrades after peaking on Hopper, while IN-RIL avoids this drop and ultimately surpasses it by 16% in rewards.

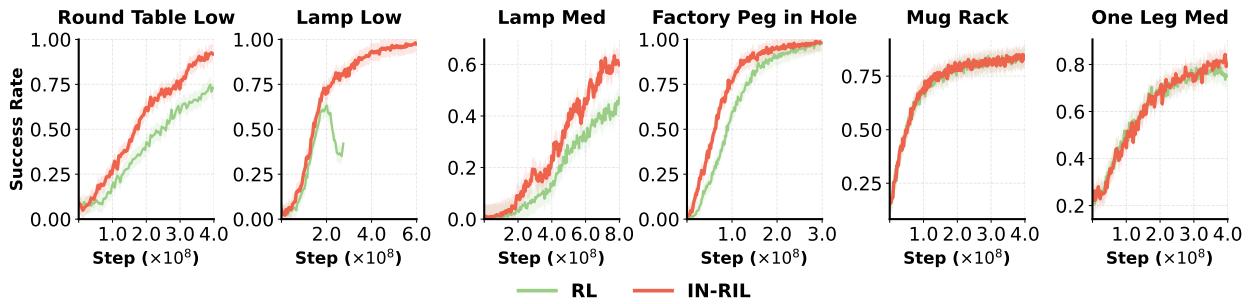


Figure 7: Comparing IN-RIL and with fine-tuning on FurnitureBench using residual PPO.

FurnitureBench features multi-stage furniture assembly with sparse rewards—conditions that are particularly difficult for RL agents, especially when IL pre-training converges at low success rates. As shown in Table 1, pre-training success rates for 3 tasks remain below 5%, with only One-Leg Low exceeding 30%. Meanwhile, IN-RIL significantly outperforms residual PPO across most tasks, as shown in Table 3, when consuming the same amount of environment steps. For the challenging Lamp Low task, RL-only fine-tuning frequently collapsed during training, while IN-RIL

maintains stable learning dynamics across multiple runs. On Round-Table Low, where pre-training achieves only 5% success rate, IN-RIL reaches 73% success rate with approximately $\times 10^8$ fewer environment interactions than RL-only fine-tuning with 25% improvement in sample efficiency.

Task	IN-RIL (Residual PPO)	Residual PPO	DPPO	IDQL
Lamp low	0.98	0.63	0.85	0.11
Lamp med	0.67	0.46	0.36	0.01
Round table low	0.93	0.73	0.88	0.09
One leg low	0.94	0.95	0.92	0.45
One leg med	0.82	0.74	0.80	0.24

Table 3: Comparing IN-RIL with other RL fine-tuning algorithms on FurnitureBench. Bold values indicate the best of all. For each method and task, we report the best success rates among all the checkpoints.

4.3 Ablation Studies on Interleaving Ratio m

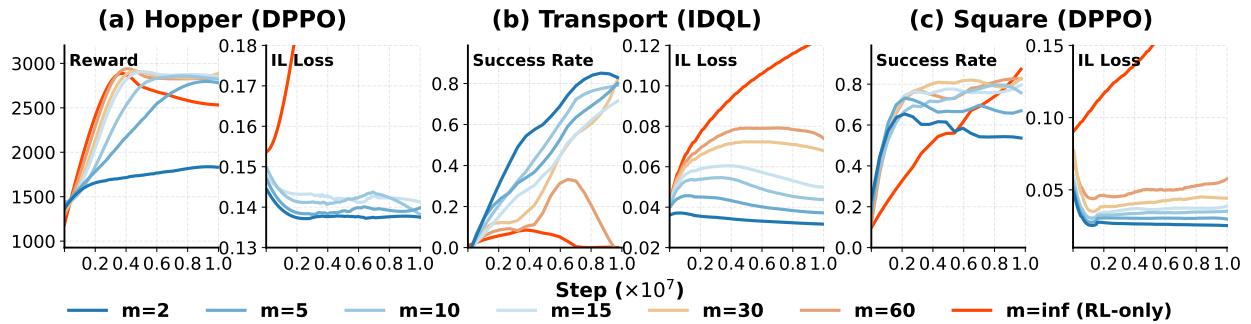


Figure 8: The impact of the interleaving period m on IN-RIL RL performance (rewards), and IL performance (IL losses). We use 7 different values for m , and train the agent with all the values using 10^7 environment steps. The figure shows how RL rewards and IL losses change with different m . The curves are smoothed using a Savitzky-Golay filter to better show the patterns.

Next, we investigate how the interleaving period m affects the learning dynamics of IN-RIL by examining changes in both online performance metrics (RL rewards) and offline performance metrics (IL losses) under different values of m . For RL-only fine-tuning ($m = \infty$), we compute IL losses to monitor how well the policy maintains fidelity to demonstrations during fine-tuning, but without updating the policy based on these losses. We evaluate IN-RIL with seven different values of m on Gym Hopper, Robomimic Transport, and Robomimic Square. In particular, Figure 8 reveals several key insights about IN-RIL’s behavior:

Double Descent of IL Losses. For RL-only fine-tuning ($m = \infty$), IL losses increase dramatically as RL exploration drives the policy away from the pre-trained behavior. In contrast, IN-RIL maintains controlled IL loss trajectories. Most remarkably, we observe that IL losses often experience a “double descent” phenomenon—after initially increasing, they begin decreasing again despite the pre-trained policy having fully converged. This empirically validates our hypothesis illustrated in Figure 3 that RL exploration can help IL escape local minima, enabling discovery of superior demonstration-aligned policies that would be inaccessible through IL alone.

Enhanced Sample Efficiency. Figure 8(c) demonstrates that IN-RIL dramatically improves the sample efficiency of DPPO, particularly during early fine-tuning. IN-RIL converges to high success rates within just 0.4×10^7 steps, while DPPO alone requires approximately 0.9×10^7 steps (2.25x more environment interactions) to achieve comparable performance.

Improved Stability. As shown in Figure 8(a), overly aggressive exploration in RL-only approaches can degrade performance after 0.4×10^7 steps. IN-RIL prevents this degradation across multiple interleaving ratios by maintaining IL losses within an appropriate range, effectively constraining exploration to promising regions of the policy space.

Guided Exploration. Figure 8(b) illustrates a critical advantage of IN-RIL: on challenging tasks where IDQL fine-tuning alone fails due to ungrounded exploration, IN-RIL successfully guides the agent toward task completion. By periodically refreshing the agent’s memory of expert demonstrations through IL gradients, IN-RIL effectively structures exploration, enabling success on tasks that RL-only approaches cannot solve.

4.4 Ablation of Separation of RL and IL Gradients.

When simultaneously leveraging IL and RL gradients to update policy networks, resolving potential interference between these distinct optimization objectives is crucial. When implementing gradient separation for IN-RIL with network separation, IL and RL gradients are naturally separated. In contrast, full-network fine-tuning, applies both gradients to the same network. To mitigate interference, we compute IL and RL gradients and apply gradient surgery before performing one gradient step to update the network. Figure 9 demonstrates that naive interleaving of IL and RL objectives without proper gradient management can significantly impair policy performance, while both separation strategies enable successful task completion.

5 Conclusion

We presented IN-RIL, a principled approach that enhances robotic policy learning by strategically interleaving IL and RL updates. Our framework maintains stability while enabling exploration through periodic IL regularization, coupled with a gradient separation mechanism that effectively combines complementary learning signals. Theoretical analysis establishes convergence guarantees and sample efficiency conditions, which align with our empirical validation across 14 diverse robot tasks from three benchmarks, demonstrating up to 6.25 \times improvement in success rate over standard RL finetuning. IN-RIL functions as a versatile plugin compatible with various state-of-the-art RL algorithms, substantially enhancing performance on both long- and short-horizon tasks with sparse or dense rewards. Future work will explore adaptive mechanisms to dynamically adjust the interleaving ratio based on gradient alignment during training, extend our approach to domains beyond robotics, and investigate additional techniques to further enhance the synergy between IL and RL objectives.

References

- [1] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.
- [2] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *8th Annual Conference on Robot Learning*, 2024.
- [3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control, 2024. URL <https://arxiv.org/abs/2410.24164>.
- [4] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.
- [5] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.
- [6] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR, 2022.
- [7] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.
- [8] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems*, 2018.
- [9] Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement-residual rl for precise assembly. *arXiv preprint arXiv:2407.16677*, 2024.
- [10] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

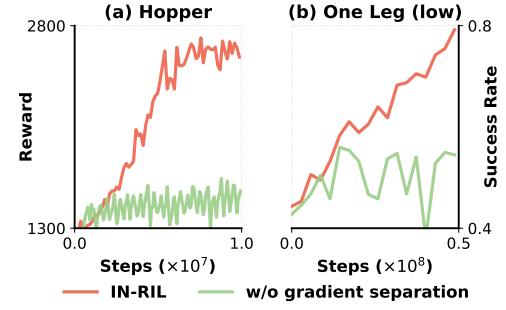


Figure 9: Impact of separation of gradients on Hopper using DPPO and One-Leg (Low) using residual PPO.

- [11] Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. [arXiv preprint arXiv:2409.00588](#), 2024.
- [12] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. [arXiv preprint arXiv:2006.09359](#), 2020.
- [13] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. [arXiv preprint arXiv:2412.13630](#), 2024.
- [14] Mitsuhiro Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. [Advances in Neural Information Processing Systems](#), 36:62244–62269, 2023.
- [15] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In [International Conference on Machine Learning](#), pages 1577–1594. PMLR, 2023.
- [16] Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. [arXiv preprint arXiv:2210.06718](#), 2022.
- [17] Hengyuan Hu, Suvir Mirchandani, and Dorsa Sadigh. Imitation bootstrapped reinforcement learning. [arXiv preprint arXiv:2311.02198](#), 2023.
- [18] Siddhant Haldar, Vaibhav Mathur, Denis Yarats, and Lerrel Pinto. Watch and match: Supercharging imitation with regularized optimal transport. In [Conference on Robot Learning](#), pages 32–43. PMLR, 2023.
- [19] O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. In [Advances in Neural Information Processing Systems](#), 2018.
- [20] Pierre Quinton and Valérian Rey. Jacobian descent for multi-objective optimization. [arXiv preprint arXiv:2406.16232](#), 2024.
- [21] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. [arXiv preprint arXiv:2304.10573](#), 2023.
- [22] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. [The International Journal of Robotics Research](#), page 02783649241304789, 2023.
- [23] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. [arXiv preprint arXiv:2108.03298](#), 2021.
- [24] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. [arXiv preprint arXiv:1606.01540](#), 2016.
- [25] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. [arXiv preprint arXiv:2212.06817](#), 2022.
- [26] 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.
- [27] Andrew Lee, Ian Chuang, Ling-Yuan Chen, and Iman Soltani. Interact: Inter-dependency aware action chunking with hierarchical attention transformers for bimanual manipulation. [arXiv preprint arXiv:2409.07914](#), 2024.
- [28] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In [Conference on robot learning](#), pages 651–673. PMLR, 2018.
- [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] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. [arXiv preprint arXiv:2301.04104](#), 2023.
- [31] Dechen Gao, Shuangyu Cai, Hanchu Zhou, Hang Wang, Iman Soltani, and Junshan Zhang. Cardreamer: Open-source learning platform for world model based autonomous driving. [IEEE Internet of Things Journal](#), 2024.
- [32] Ian Chuang, Andrew Lee, Dechen Gao, M Naddaf-Sh, Iman Soltani, et al. Active vision might be all you need: Exploring active vision in bimanual robotic manipulation. [arXiv preprint arXiv:2409.17435](#), 2024.

- [33] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. [arXiv preprint arXiv:1910.11956](#), 2019.
- [34] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. [arXiv preprint arXiv:2501.12948](#), 2025.
- [35] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. [Comptes Rendus Mathematique](#), 350(5-6):313–318, 2012.
- [36] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.
- [37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. [arXiv preprint arXiv:1707.06347](#), 2017.
- [38] Long Yang, Zhixiong Huang, Fenghao Lei, Yucun Zhong, Yiming Yang, Cong Fang, Shiting Wen, Binbin Zhou, and Zhouchen Lin. Policy representation via diffusion probability model for reinforcement learning. [arXiv preprint arXiv:2305.13122](#), 2023.
- [39] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. [arXiv preprint arXiv:1910.00177](#), 2019.
- [40] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. [Advances in Neural Information Processing Systems](#), 33:1877–1901, 2020.
- [41] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. [arXiv preprint arXiv:2108.07258](#), 2021.
- [42] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In [Proceedings of the fourteenth international conference on artificial intelligence and statistics](#), pages 627–635. JMLR Workshop and Conference Proceedings, 2011.
- [43] Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. volume 75, pages 297–325, 2009.
- [44] Yurii Nesterov. [Introductory Lectures on Convex Optimization: A Basic Course](#). Springer Science & Business Media, 2004.
- [45] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. [SIAM Review](#), 2018.
- [46] Herbert Robbins and Sutton Monro. A stochastic approximation method. [The Annals of Mathematical Statistics](#), 22(3):400–407, 1951.

Appendix

A Justifications on the Assumptions

Assumption 2 (Pretraining Performance). *The initial policy parameters θ_0 obtained from pretraining satisfies $\mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}(\theta^*) \leq \epsilon_{\text{IL}}$, where $\epsilon_{\text{IL}} > 0$ is a constant and θ^* is the optimal solution for optimizing the IL objective.*

Assumption 3 (Data Coverage). *The expert demonstration dataset \mathcal{D}_{exp} provides sufficient coverage of the state space relevant for the target task. Specifically, there exists a constant $C_{\text{coverage}} > 0$ such that:*

$$\mathbb{E}_{s \sim \mu^*} [\min_{s' \in \mathcal{D}_{\text{exp}}} \|s - s'\|] \leq C_{\text{coverage}}$$

where μ^* is the state distribution of the optimal policy for the target task.

Assumption 4 (Smoothness of Objectives). *Both the IL and RL objectives are L -smooth:*

$$\begin{aligned} \|\nabla_{\theta} \mathcal{L}_{\text{IL}}(\theta) - \nabla_{\theta} \mathcal{L}_{\text{IL}}(\theta')\| &\leq L_{\text{IL}} \|\theta - \theta'\|, \quad \forall \theta, \theta' \\ \|\nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta) - \nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta')\| &\leq L_{\text{RL}} \|\theta - \theta'\|, \quad \forall \theta, \theta' \end{aligned}$$

Assumption 5 (Bounded Variance). *The stochastic gradients have bounded variance:*

$$\begin{aligned} \mathbb{E}[\|\nabla_{\theta} \mathcal{L}_{\text{IL}}(\theta) - \hat{\nabla}_{\theta} \mathcal{L}_{\text{IL}}(\theta)\|^2] &\leq \frac{\sigma_{\text{IL}}^2}{N_{\text{IL}}} \\ \mathbb{E}[\|\nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta) - \hat{\nabla}_{\theta} \mathcal{L}_{\text{RL}}(\theta)\|^2] &\leq \frac{\sigma_{\text{RL}}^2}{N_{\text{RL}}} \end{aligned}$$

where $\hat{\nabla}$ represents the stochastic gradient estimate, and N_{IL} and N_{RL} are the batch sizes.

We first provide the detailed justification on the assumptions used in Section 2.

Assumption 1 (Near-Optimal IL Performance) This assumption reflects the practical setting where we start from a pre-trained policy that already performs well on demonstration data. It's commonly used in transfer learning and foundation model literature where models are first trained on large datasets before task-specific adaptation [40, 41]. The small constant ϵ_{IL} quantifies how close the initial policy is to optimal imitation performance, capturing the idea that while the model has learned a good behavioral prior, there's still room for improvement through reinforcement learning.

Assumption 2 (Data Coverage) The data coverage assumption ensures that the expert demonstrations provide adequate representation of the states relevant to the target task. This is a standard assumption in imitation learning [42, 43] and reflects the intuition that learning can only occur for regions of the state space that have been demonstrated. The constant C_{coverage} quantifies the maximum expected distance between a state from the optimal policy and its nearest neighbor in the demonstration dataset, with smaller values indicating better coverage.

Assumption 3 (Smoothness of Objectives) Smoothness is a standard assumption in optimization theory [44, 45] that ensures the gradient doesn't change too drastically between nearby points. This enables reliable gradient-based optimization and allows us to derive convergence rates. Practically, this assumption holds for most neural network architectures with commonly used activation functions when properly normalized, and is critical for establishing the descent lemma used in our analysis.

Assumption 4 (Gradient Alignment) This assumption characterizes the geometric relationship between the gradients of the IL and RL objectives. The parameter $\rho(t)$ captures the cosine similarity between these gradients, with positive values indicating opposing gradients and negative values indicating aligned gradients. Similar assumptions appear in multi-task learning literature [19] and multi-objective optimization [35]. This formulation allows us to analyze how the IL updates affect progress on the RL objective, which is crucial for determining the optimal interleaving strategy.

Assumption 5 (Bounded Variance) The bounded variance assumption is standard in stochastic optimization literature [46, 45] and reflects the fact that stochastic gradient estimates contain noise due to mini-batch sampling. The variance terms σ_{IL}^2 and σ_{RL}^2 quantify this noise, with the variance decreasing as batch size increases. This assumption is necessary for establishing convergence rates in the presence of stochastic gradients and is satisfied in practice when using proper mini-batch sampling techniques.

Based on these assumptions, we first establish the following key results (proofs in the appendix). We begin our theoretical analysis by establishing convergence analysis for RL-only finetune and IN-RIL, respectively.

Theorem 3 (Convergence of RL-Only Training). *Under Assumptions 2-5, with learning rate $\alpha_{\text{RL}} = \frac{c_{\text{RL}}}{L_{\text{RL}}}$ for $c_{\text{RL}} \in (0, 1)$, RL-only training for T iterations achieves:*

$$\min_{0 \leq t < T} \mathbb{E}[\|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2] \leq \frac{2L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})T} + \frac{c_{\text{RL}}\sigma_{\text{RL}}^2}{(1 - \frac{c_{\text{RL}}}{2})N_{\text{RL}}}$$

Theorem 4 (Convergence with IN-RIL). *Under Assumptions 2-5, with learning rates $\alpha_{\text{IL}} = \frac{c_{\text{IL}}}{L_{\text{IL}}}$ and $\alpha_{\text{RL}} = \frac{c_{\text{RL}}}{L_{\text{RL}}}$ for $c_{\text{IL}}, c_{\text{RL}} \in (0, 1)$, interleaved 1:m(t) training for T cycles achieves:*

$$\min_{0 \leq t < T} \mathbb{E}[\|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2] \leq \frac{2(L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}})}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})\bar{m}T} + \frac{c_{\text{RL}}\sigma_{\text{RL}}^2}{(1 - \frac{c_{\text{RL}}}{2})N_{\text{RL}}}$$

where $\bar{m} = \frac{1}{T} \sum_{t=0}^{T-1} m(t)$ is the average interleaving ratio, and $\Delta_{\text{IL-RL}}$ represents the benefit from IL regularization, i.e., $\Delta_{\text{IL-RL}} = -\sum_{t=0}^{T-1} \frac{c_{\text{IL}}\rho(t)}{L_{\text{IL}}} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| - \frac{c_{\text{IL}}^2\sigma_{\text{IL}}^2 T}{2L_{\text{IL}}N_{\text{IL}}}$

Theorem 3 establishes that with appropriate learning rates, RL-only finetuning achieves the standard $O(1/T)$ convergence rate for smooth objectives. Theorem 4 reveals that IN-RIL can achieve better convergence guarantees than RL-only finetuning through the regularization benefit term $\Delta_{\text{IL-RL}}$. This term captures how IL updates can enhance RL performance, especially when gradient alignment is favorable ($\rho(t) < 0$). Having established the benefits of IN-RIL, we now derive the optimal ratio of RL updates to IL updates. This ratio is crucial for balancing the stability provided by IL updates with the performance improvements offered by RL updates.

B Proof of Theorem 3

We first establish the following technical lemmas that will be used in the proof of the main theorems.

Lemma 1 (Descent Lemma). *For a function f with L -smoothness, we have:*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2$$

Lemma 2 (Progress Bound for Gradient Descent). *For a function f with L -smoothness and step size $\alpha = \frac{c}{L}$ where $c \in (0, 1)$, one step of gradient descent gives:*

$$f(x - \alpha \nabla f(x)) \leq f(x) - \frac{c(1 - \frac{c}{2})}{L} \|\nabla f(x)\|^2$$

Lemma 3 (Error Bound for Stochastic Gradient Descent). *For a function f with L -smoothness, step size $\alpha = \frac{c}{L}$ where $c \in (0, 1)$, and stochastic gradient $\widehat{\nabla} f(x)$ with bounded variance $\mathbb{E}[\|\nabla f(x) - \widehat{\nabla} f(x)\|^2] \leq \frac{\sigma^2}{N}$, one step of stochastic gradient descent gives:*

$$\mathbb{E}[f(x - \alpha \widehat{\nabla} f(x))] \leq f(x) - \frac{c(1 - \frac{c}{2})}{L} \|\nabla f(x)\|^2 + \frac{c^2\sigma^2}{2LN}$$

Proof. The RL-only update rule is:

$$\theta_{t+1} = \theta_t - \alpha_{\text{RL}} \widehat{\nabla}_\theta \mathcal{L}_{\text{RL}}(\theta_t)$$

Where $\widehat{\nabla}_\theta \mathcal{L}_{\text{RL}}(\theta_t)$ is the stochastic gradient estimate. Applying Lemma 3 to the RL objective, with $\alpha_{\text{RL}} = \frac{c_{\text{RL}}}{L_{\text{RL}}}$:

$$\mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+1})] \leq \mathcal{L}_{\text{RL}}(\theta_t) - \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 + \frac{c_{\text{RL}}^2\sigma_{\text{RL}}^2}{2L_{\text{RL}}N_{\text{RL}}}$$

Rearranging:

$$\frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \mathcal{L}_{\text{RL}}(\theta_t) - \mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+1})] + \frac{c_{\text{RL}}^2\sigma_{\text{RL}}^2}{2L_{\text{RL}}N_{\text{RL}}}$$

Summing from $t = 0$ to $T - 1$:

$$\frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \mathcal{L}_{\text{RL}}(\theta_0) - \mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_T)] + \frac{c_{\text{RL}}^2\sigma_{\text{RL}}^2 T}{2L_{\text{RL}}N_{\text{RL}}}$$

By Assumption 6, $\mathcal{L}_{\text{RL}}(\theta_T) \geq \mathcal{L}_{\text{RL}}^*$ (the optimal value), so:

$$\frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^* + \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2 T}{2L_{\text{RL}} N_{\text{RL}}}$$

By the pigeonhole principle, there must exist at least one iteration $t^* \in \{0, 1, \dots, T-1\}$ such that:

$$\|\nabla \mathcal{L}_{\text{RL}}(\theta_{t^*})\|^2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2$$

Therefore:

$$\min_{0 \leq t < T} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})T} + \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})N_{\text{RL}}}$$

Simplifying the second term:

$$\min_{0 \leq t < T} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})T} + \frac{c_{\text{RL}} \sigma_{\text{RL}}^2}{2(1 - \frac{c_{\text{RL}}}{2})N_{\text{RL}}}$$

Taking expectation and adjusting the constant in the second term:

$$\min_{0 \leq t < T} \mathbb{E}[\|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2] \leq \frac{2L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})T} + \frac{c_{\text{RL}} \sigma_{\text{RL}}^2}{(1 - \frac{c_{\text{RL}}}{2})N_{\text{RL}}}$$

For the IL performance bound, we use the L_{IL} -smoothness of the IL objective (Assumption 3):

$$\begin{aligned} \mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta_0) &\leq \langle \nabla \mathcal{L}_{\text{IL}}(\theta_0), \theta_T - \theta_0 \rangle + \frac{L_{\text{IL}}}{2} \|\theta_T - \theta_0\|^2 \\ &\leq \|\nabla \mathcal{L}_{\text{IL}}(\theta_0)\| \cdot \|\theta_T - \theta_0\| + \frac{L_{\text{IL}}}{2} \|\theta_T - \theta_0\|^2 \end{aligned}$$

From Assumption 1 (Near-Optimal IL Performance), the gradient $\|\nabla \mathcal{L}_{\text{IL}}(\theta_0)\|$ is small. For simplicity, we can absorb this term into the quadratic term:

$$\mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta_0) \leq \frac{L_{\text{IL}}}{2} \|\theta_T - \theta_0\|^2$$

Combining with Assumption 1, we have:

$$\begin{aligned} \mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta^*) &= \mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta_0) + \mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}(\theta^*) \\ &\leq \frac{L_{\text{IL}}}{2} \|\theta_T - \theta_0\|^2 + \epsilon_{\text{IL}} \end{aligned}$$

This completes the proof. \square

C Proof of Theorem 4

Proof. The interleaved training consists of cycles where each cycle has one IL update followed by $m(t)$ RL updates. Let θ_t denote the parameters at the beginning of cycle t , and $\theta_{t+\frac{j}{1+m(t)}}$ denote the parameters after the j -th update within cycle t .

First, let's analyze the IL update within cycle t :

$$\theta_{t+\frac{1}{1+m(t)}} = \theta_t - \alpha_{\text{IL}} \hat{\nabla} \mathcal{L}_{\text{IL}}(\theta_t)$$

Applying Lemma 3 to the IL objective with $\alpha_{\text{IL}} = \frac{c_{\text{IL}}}{L_{\text{IL}}}$:

$$\mathbb{E}[\mathcal{L}_{\text{IL}}(\theta_{t+\frac{1}{1+m(t)}})] \leq \mathcal{L}_{\text{IL}}(\theta_t) - \frac{c_{\text{IL}}(1 - \frac{c_{\text{IL}}}{2})}{L_{\text{IL}}} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2L_{\text{IL}} N_{\text{IL}}}$$

Now, let's analyze how this IL update affects the RL objective. Using the smoothness of the RL objective (Assumption 3):

$$\begin{aligned}\mathcal{L}_{\text{RL}}(\theta_{t+\frac{1}{1+m(t)}}) &\leq \mathcal{L}_{\text{RL}}(\theta_t) + \langle \nabla \mathcal{L}_{\text{RL}}(\theta_t), \theta_{t+\frac{1}{1+m(t)}} - \theta_t \rangle + \frac{L_{\text{RL}}}{2} \|\theta_{t+\frac{1}{1+m(t)}} - \theta_t\|^2 \\ &= \mathcal{L}_{\text{RL}}(\theta_t) + \langle \nabla \mathcal{L}_{\text{RL}}(\theta_t), -\alpha_{\text{IL}} \hat{\nabla} \mathcal{L}_{\text{IL}}(\theta_t) \rangle + \frac{L_{\text{RL}} \alpha_{\text{IL}}^2}{2} \|\hat{\nabla} \mathcal{L}_{\text{IL}}(\theta_t)\|^2\end{aligned}$$

Taking expectations and using the fact that $\mathbb{E}[\hat{\nabla} \mathcal{L}_{\text{IL}}(\theta_t)] = \nabla \mathcal{L}_{\text{IL}}(\theta_t)$ (unbiased estimator):

$$\mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+\frac{1}{1+m(t)}})] \leq \mathcal{L}_{\text{RL}}(\theta_t) - \alpha_{\text{IL}} \langle \nabla \mathcal{L}_{\text{RL}}(\theta_t), \nabla \mathcal{L}_{\text{IL}}(\theta_t) \rangle + \frac{L_{\text{RL}} \alpha_{\text{IL}}^2}{2} \mathbb{E}[\|\hat{\nabla} \mathcal{L}_{\text{IL}}(\theta_t)\|^2]$$

Using Assumption 4 (Gradient align*ment):

$$\langle \nabla \mathcal{L}_{\text{IL}}(\theta_t), \nabla \mathcal{L}_{\text{RL}}(\theta_t) \rangle = -\rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|$$

And using Assumption 5 (Bounded Variance):

$$\mathbb{E}[\|\hat{\nabla} \mathcal{L}_{\text{IL}}(\theta_t)\|^2] \leq \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{\sigma_{\text{IL}}^2}{N_{\text{IL}}}$$

We get:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+\frac{1}{1+m(t)}})] &\leq \mathcal{L}_{\text{RL}}(\theta_t) + \alpha_{\text{IL}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &\quad + \frac{L_{\text{RL}} \alpha_{\text{IL}}^2}{2} \left(\|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{\sigma_{\text{IL}}^2}{N_{\text{IL}}} \right)\end{aligned}$$

Substituting $\alpha_{\text{IL}} = \frac{c_{\text{IL}}}{L_{\text{IL}}}$:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+\frac{1}{1+m(t)}})] &\leq \mathcal{L}_{\text{RL}}(\theta_t) + \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &\quad + \frac{L_{\text{RL}} c_{\text{IL}}^2}{2 L_{\text{IL}}^2} \left(\|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{\sigma_{\text{IL}}^2}{N_{\text{IL}}} \right)\end{aligned}$$

Now, let's analyze the $m(t)$ RL updates. For each RL update $j \in \{1, \dots, m(t)\}$:

$$\theta_{t+\frac{1+j}{1+m(t)}} = \theta_{t+\frac{j}{1+m(t)}} - \alpha_{\text{RL}} \hat{\nabla} \mathcal{L}_{\text{RL}}(\theta_{t+\frac{j}{1+m(t)}})$$

Applying Lemma 3 to each RL update, with $\alpha_{\text{RL}} = \frac{c_{\text{RL}}}{L_{\text{RL}}}$:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+\frac{1+j}{1+m(t)}})] &\leq \mathcal{L}_{\text{RL}}(\theta_{t+\frac{j}{1+m(t)}}) - \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_{t+\frac{j}{1+m(t)}})\|^2 \\ &\quad + \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2 L_{\text{RL}} N_{\text{RL}}}\end{aligned}$$

For simplicity of analysis, we can bound the gradient norms at intermediate steps using the gradient at the beginning of the cycle:

$$\|\nabla \mathcal{L}_{\text{RL}}(\theta_{t+\frac{j}{1+m(t)}})\|^2 \geq (1 - \delta)^2 \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2$$

for some small $\delta > 0$ that depends on the learning rates and smoothness constants. This approximation is reasonable because the parameters don't change drastically within a cycle when using small learning rates.

With this approximation, we get:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+\frac{1+j}{1+m(t)}})] &\leq \mathcal{L}_{\text{RL}}(\theta_{t+\frac{j}{1+m(t)}}) - \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})(1 - \delta)^2}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \\ &\quad + \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2 L_{\text{RL}} N_{\text{RL}}}\end{aligned}$$

Applying this recursively for all $m(t)$ RL updates and combining with the effect of the IL update, we get:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+1})] &\leq \mathcal{L}_{\text{RL}}(\theta_t) + \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &\quad + \frac{L_{\text{RL}} c_{\text{IL}}^2}{2L_{\text{IL}}^2} \left(\|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{\sigma_{\text{IL}}^2}{N_{\text{IL}}} \right) \\ &\quad - m(t) \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})(1 - \delta)^2}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 + m(t) \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2L_{\text{RL}} N_{\text{RL}}}\end{aligned}$$

For simplicity, we'll absorb $(1 - \delta)^2$ into the constants. Rearranging:

$$\begin{aligned}m(t) \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 &\leq \mathcal{L}_{\text{RL}}(\theta_t) - \mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_{t+1})] \\ &\quad + \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &\quad + \frac{L_{\text{RL}} c_{\text{IL}}^2}{2L_{\text{IL}}^2} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{L_{\text{RL}} c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2L_{\text{IL}}^2 N_{\text{IL}}} \\ &\quad + m(t) \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2L_{\text{RL}} N_{\text{RL}}}\end{aligned}$$

Summing over $t = 0$ to $T - 1$:

$$\begin{aligned}\sum_{t=0}^{T-1} m(t) \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 &\leq \mathcal{L}_{\text{RL}}(\theta_0) - \mathbb{E}[\mathcal{L}_{\text{RL}}(\theta_T)] \\ &\quad + \sum_{t=0}^{T-1} \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &\quad + \sum_{t=0}^{T-1} \frac{L_{\text{RL}} c_{\text{IL}}^2}{2L_{\text{IL}}^2} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + T \frac{L_{\text{RL}} c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2L_{\text{IL}}^2 N_{\text{IL}}} \\ &\quad + \sum_{t=0}^{T-1} m(t) \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2L_{\text{RL}} N_{\text{RL}}}\end{aligned}$$

By Assumption 6, $\mathcal{L}_{\text{RL}}(\theta_T) \geq \mathcal{L}_{\text{RL}}^*$, so:

$$\begin{aligned}\sum_{t=0}^{T-1} m(t) \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 &\leq \mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^* \\ &\quad + \sum_{t=0}^{T-1} \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &\quad + \sum_{t=0}^{T-1} \frac{L_{\text{RL}} c_{\text{IL}}^2}{2L_{\text{IL}}^2} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + T \frac{L_{\text{RL}} c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2L_{\text{IL}}^2 N_{\text{IL}}} \\ &\quad + \sum_{t=0}^{T-1} m(t) \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2L_{\text{RL}} N_{\text{RL}}}\end{aligned}$$

For the sum of IL gradient norms, we can use the IL update analysis. From our earlier bound on IL updates:

$$\sum_{t=0}^{T-1} \frac{c_{\text{IL}}(1 - \frac{c_{\text{IL}}}{2})}{L_{\text{IL}}} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 \leq \mathcal{L}_{\text{IL}}(\theta_0) - \mathbb{E}[\mathcal{L}_{\text{IL}}(\theta_T)] + \frac{c_{\text{IL}}^2 \sigma_{\text{IL}}^2 T}{2L_{\text{IL}} N_{\text{IL}}}$$

This gives us:

$$\sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 \leq \frac{L_{\text{IL}} (\mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}^*)}{c_{\text{IL}} (1 - \frac{c_{\text{IL}}}{2})} + \frac{c_{\text{IL}} \sigma_{\text{IL}}^2 T}{2(1 - \frac{c_{\text{IL}}}{2}) N_{\text{IL}}}$$

Substituting this bound and defining $\bar{m} = \frac{1}{T} \sum_{t=0}^{T-1} m(t)$ as the average interleaving ratio:

$$\begin{aligned} \bar{m}T \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 &\leq \mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^* \\ &+ \sum_{t=0}^{T-1} \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &+ \frac{L_{\text{RL}} c_{\text{IL}}^2}{2L_{\text{IL}}^2} \cdot \frac{L_{\text{IL}}(\mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}^*)}{c_{\text{IL}}(1 - \frac{c_{\text{IL}}}{2})} + T \frac{L_{\text{RL}} c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2L_{\text{IL}}^2 N_{\text{IL}}} \\ &+ \bar{m}T \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2L_{\text{RL}} N_{\text{RL}}} \end{aligned}$$

The term with IL gradient norms can be simplified to:

$$\frac{L_{\text{RL}} c_{\text{IL}}^2}{2L_{\text{IL}}^2} \cdot \frac{L_{\text{IL}}(\mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}^*)}{c_{\text{IL}}(1 - \frac{c_{\text{IL}}}{2})} = \frac{L_{\text{RL}} c_{\text{IL}}}{2L_{\text{IL}}} \cdot \frac{(\mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}^*)}{(1 - \frac{c_{\text{IL}}}{2})}$$

By Assumption 1, $\mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}^* \leq \epsilon_{\text{IL}}$, which is small. For large enough T , this term becomes negligible.

Define the IL regularization benefit:

$$\Delta_{\text{IL-RL}} = - \sum_{t=0}^{T-1} \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| + \frac{c_{\text{IL}}^2 \sigma_{\text{IL}}^2 T}{2L_{\text{IL}} N_{\text{IL}}}$$

With this, our bound becomes:

$$\bar{m} \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \frac{\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^* - \Delta_{\text{IL-RL}}}{T} + \bar{m} \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2L_{\text{RL}} N_{\text{RL}}}$$

By the pigeonhole principle, there must exist at least one iteration $t^* \in \{0, 1, \dots, T-1\}$ such that:

$$\|\nabla \mathcal{L}_{\text{RL}}(\theta_{t^*})\|^2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2$$

Therefore:

$$\min_{0 \leq t < T} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \leq \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^* - \Delta_{\text{IL-RL}})}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2}) \bar{m} T} + \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2}) N_{\text{RL}}}$$

Taking expectation and adjusting the constant in the second term:

$$\min_{0 \leq t < T} \mathbb{E}[\|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2] \leq \frac{2(L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}})}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2}) \bar{m} T} + \frac{c_{\text{RL}} \sigma_{\text{RL}}^2}{(1 - \frac{c_{\text{RL}}}{2}) N_{\text{RL}}}$$

For the IL performance bound, using the earlier bound on IL updates and summing over all cycles:

$$\mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta_0) \leq - \sum_{t=0}^{T-1} \frac{c_{\text{IL}}(1 - \frac{c_{\text{IL}}}{2})}{L_{\text{IL}}} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{c_{\text{IL}}^2 \sigma_{\text{IL}}^2 T}{2L_{\text{IL}} N_{\text{IL}}}$$

Combining with Assumption 1:

$$\begin{aligned} \mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta^*) &= \mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta_0) + \mathcal{L}_{\text{IL}}(\theta_0) - \mathcal{L}_{\text{IL}}(\theta^*) \\ &\leq - \sum_{t=0}^{T-1} \frac{c_{\text{IL}}(1 - \frac{c_{\text{IL}}}{2})}{L_{\text{IL}}} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2 + \frac{c_{\text{IL}}^2 \sigma_{\text{IL}}^2 T}{2L_{\text{IL}} N_{\text{IL}}} + \epsilon_{\text{IL}} \end{aligned}$$

Additionally, by the L_{IL} -smoothness of the IL objective:

$$\mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta_0) \leq \frac{L_{\text{IL}}}{2} \|\theta_T - \theta_0\|^2$$

Combining these bounds:

$$\mathcal{L}_{\text{IL}}(\theta_T) - \mathcal{L}_{\text{IL}}(\theta^*) \leq \epsilon_{\text{IL}} + \frac{L_{\text{IL}}}{2} \|\theta_T - \theta_0\|^2 - \sum_{t=0}^{T-1} \frac{c_{\text{IL}}(1 - \frac{c_{\text{IL}}}{2})}{L_{\text{IL}}} \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\|^2$$

This shows that the periodic IL updates in interleaved training help maintain good IL performance compared to RL-only training. \square

D Proof of Theorem 1

Proof. To find the optimal ratio $m(t)$ at iteration t , we want to maximize the progress per update. From our analysis in Theorem 2, the progress for one complete cycle is:

$$\begin{aligned} \mathcal{L}_{\text{RL}}(\theta_t) - \mathcal{L}_{\text{RL}}(\theta_{t+1}) &\approx m(t) \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \\ &\quad - \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| \\ &\quad - \frac{L_{\text{RL}} c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2 L_{\text{IL}}^2 N_{\text{IL}}} - m(t) \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2 L_{\text{RL}} N_{\text{RL}}} \end{aligned}$$

Since each cycle consists of $1 + m(t)$ updates, the progress per update is:

$$\frac{\mathcal{L}_{\text{RL}}(\theta_t) - \mathcal{L}_{\text{RL}}(\theta_{t+1})}{1 + m(t)} \approx \frac{m(t) \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 - \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| - \frac{L_{\text{RL}} c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2 L_{\text{IL}}^2 N_{\text{IL}}} - m(t) \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2 L_{\text{RL}} N_{\text{RL}}}}{1 + m(t)}$$

To find the optimal $m(t)$, we differentiate this expression with respect to $m(t)$ and set it to zero. Let's denote:

$$\begin{aligned} A &= \frac{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})}{L_{\text{RL}}} \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\|^2 \\ B &= \frac{c_{\text{IL}}}{L_{\text{IL}}} \rho(t) \|\nabla \mathcal{L}_{\text{IL}}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{\text{RL}}(\theta_t)\| + \frac{L_{\text{RL}} c_{\text{IL}}^2 \sigma_{\text{IL}}^2}{2 L_{\text{IL}}^2 N_{\text{IL}}} \\ C &= \frac{c_{\text{RL}}^2 \sigma_{\text{RL}}^2}{2 L_{\text{RL}} N_{\text{RL}}} \end{aligned}$$

Then the progress per update is:

$$\frac{mA - B - mC}{1 + m}$$

Differentiating with respect to m :

$$\begin{aligned} \frac{d}{dm} \left(\frac{mA - B - mC}{1 + m} \right) &= \frac{(A - C)(1 + m) - (mA - B - mC)}{(1 + m)^2} \\ &= \frac{A - C + mA - mC - mA + B + mC}{(1 + m)^2} \\ &= \frac{A - C + B}{(1 + m)^2} \end{aligned}$$

For this to be zero, we need $A - C + B = 0$, which is not possible in general if $A > C$ (which is the case when the RL objective has room for improvement). Therefore, the derivative is always positive or always negative.

Since we're looking for a maximum, we need to check the second derivative:

$$\begin{aligned} \frac{d^2}{dm^2} \left(\frac{mA - B - mC}{1+m} \right) &= \frac{d}{dm} \left(\frac{A - C + B}{(1+m)^2} \right) \\ &= (A - C + B) \cdot \frac{d}{dm} \left(\frac{1}{(1+m)^2} \right) \\ &= (A - C + B) \cdot \left(-\frac{2}{(1+m)^3} \right) \\ &= -\frac{2(A - C + B)}{(1+m)^3} \end{aligned}$$

When $A - C > B$, the second derivative is negative, indicating a maximum. In this case, the progress per update increases with m , and the optimal $m(t)$ would be as large as possible.

However, for practical reasons, we want to maintain some IL updates, so we need to find a suitable $m(t)$ that balances progress and regularization. One approach is to equate the progress from RL updates with the potential negative impact of the IL update:

$$m(t) \frac{c_{RL}(1 - \frac{c_{RL}}{2})}{L_{RL}} \|\nabla \mathcal{L}_{RL}(\theta_t)\|^2 \approx \frac{c_{IL}}{L_{IL}} \rho(t) \|\nabla \mathcal{L}_{IL}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{RL}(\theta_t)\| + \frac{L_{RL} c_{IL}^2 \sigma_{IL}^2}{2 L_{IL}^2 N_{IL}}$$

Solving for $m(t)$:

$$\begin{aligned} m(t) &\approx \frac{\frac{c_{IL}}{L_{IL}} \rho(t) \|\nabla \mathcal{L}_{IL}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{RL}(\theta_t)\| + \frac{L_{RL} c_{IL}^2 \sigma_{IL}^2}{2 L_{IL}^2 N_{IL}}}{\frac{c_{RL}(1 - \frac{c_{RL}}{2})}{L_{RL}} \|\nabla \mathcal{L}_{RL}(\theta_t)\|^2} \\ &= \frac{L_{RL} c_{IL} \rho(t) \|\nabla \mathcal{L}_{IL}(\theta_t)\|}{L_{IL} c_{RL} (1 - \frac{c_{RL}}{2}) \|\nabla \mathcal{L}_{RL}(\theta_t)\|} + \frac{L_{RL}^2 c_{IL}^2 \sigma_{IL}^2}{2 L_{IL}^2 N_{IL} c_{RL} (1 - \frac{c_{RL}}{2}) \|\nabla \mathcal{L}_{RL}(\theta_t)\|^2} \end{aligned}$$

When gradients are opposing ($\rho(t) > 0$), this can give a reasonably large $m(t)$. When gradients are aligned ($\rho(t) < 0$), the optimal $m(t)$ would be smaller.

A more practical approach is to use a square root formula that balances these factors:

$$m_{opt}(t) = \max \left\{ 1, \sqrt{\frac{\|\nabla \mathcal{L}_{RL}(\theta_t)\|^2}{\rho(t) \|\nabla \mathcal{L}_{IL}(\theta_t)\| \cdot \|\nabla \mathcal{L}_{RL}(\theta_t)\| - \frac{c_{IL} L_{RL} \sigma_{IL}^2}{2 L_{IL}^2 N_{IL}}}} \right\}$$

This formula ensures that: 1. $m(t)$ is at least 1 (we always do at least one RL update per IL update) 2. $m(t)$ increases when RL gradients are large relative to IL gradients 3. $m(t)$ increases when gradients oppose each other ($\rho(t) > 0$ and large) 4. $m(t)$ decreases when gradients align* ($\rho(t) < 0$)

The specific constants may need to be adjusted based on empirical observations, but this formula provides a theoretically justified starting point for adaptive interleaving. \square

E Proof of Theorem 2

Proof. From Theorem 1, the number of iterations required for RL-only training to reach a target accuracy $\min_{0 \leq t < T} \|\nabla \mathcal{L}_{RL}(\theta_t)\|^2 \leq \epsilon$ is:

$$T_{RL\text{-only}} \approx \frac{2L_{RL}(\mathcal{L}_{RL}(\theta_0) - \mathcal{L}_{RL}^*)}{c_{RL}(1 - \frac{c_{RL}}{2})\epsilon}$$

From Theorem 2, the number of cycles required for interleaved 1:m(t) training to reach the same accuracy is:

$$T_{\text{interleaved, cycles}} \approx \frac{2(L_{RL}(\mathcal{L}_{RL}(\theta_0) - \mathcal{L}_{RL}^*) - \Delta_{IL-RL})}{c_{RL}(1 - \frac{c_{RL}}{2})\bar{m}\epsilon}$$

Since each cycle consists of $1 + m(t)$ updates, the total number of updates required for interleaved training is:

$$\begin{aligned} T_{\text{interleaved, updates}} &\approx (1 + \bar{m})T_{\text{interleaved, cycles}} \\ &\approx (1 + \bar{m}) \frac{2(L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}})}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})\bar{m}\epsilon} \end{aligned}$$

For a fair comparison, we compare the total number of updates required by both methods. The ratio is:

$$\begin{aligned} \frac{T_{\text{RL-only}}}{T_{\text{interleaved, updates}}} &= \frac{\frac{2L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})\epsilon}}{(1 + \bar{m}) \frac{2(L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}})}{c_{\text{RL}}(1 - \frac{c_{\text{RL}}}{2})\bar{m}\epsilon}} \\ &= \frac{\bar{m}}{1 + \bar{m}} \cdot \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}}} \end{aligned}$$

When $\Delta_{\text{IL-RL}} > 0$ (positive regularization benefit) and $\bar{m} > 1$, this ratio can be greater than 1, indicating that interleaved training requires fewer total updates than RL-only training to achieve the same level of accuracy.

Specifically, if we define the relative regularization benefit:

$$\beta = \frac{\Delta_{\text{IL-RL}}}{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}$$

Then the ratio becomes:

$$\frac{T_{\text{RL-only}}}{T_{\text{interleaved, updates}}} = \frac{\bar{m}}{1 + \bar{m}} \cdot \frac{1}{1 - \beta}$$

For interleaved training to be more efficient than RL-only training, we need:

$$\frac{\bar{m}}{1 + \bar{m}} \cdot \frac{1}{1 - \beta} > 1$$

This is satisfied when:

$$\beta > 1 - \frac{\bar{m}}{1 + \bar{m}} = \frac{1}{1 + \bar{m}}$$

For example, with $\bar{m} = 3$, interleaved training is more efficient when $\beta > \frac{1}{4}$, i.e., when the regularization benefit is at least 25% of the potential RL improvement. \square

E.1 Interpreting the Efficiency Advantage

Our theoretical analysis requires careful interpretation to properly understand the efficiency relationship between IN-RIL and RL-only methods. In what follows, we further examine the key results and their implications.

E.1.1 Efficiency Ratio

From our theoretical analysis, we derived the efficiency ratio comparing RL-only updates to total interleaved updates:

$$\frac{T_{\text{RL-only}}}{T_{\text{IN-RIL,total}}} = \frac{m_{\text{opt}}}{1 + m_{\text{opt}}} \cdot \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}}}$$

Let's examine this ratio's behavior in different scenarios:

1. **As $m_{\text{opt}} \rightarrow \infty$:** The term $\frac{m_{\text{opt}}}{1 + m_{\text{opt}}} \rightarrow 1$, and the ratio approaches $\frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}}}$
2. **When $\Delta_{\text{IL-RL}} = 0$:** The ratio simplifies to $\frac{m_{\text{opt}}}{1 + m_{\text{opt}}}$, which is always less than 1, indicating that IN-RIL requires more updates
3. **When $\Delta_{\text{IL-RL}} > 0$:** The ratio may exceed 1 if the regularization benefit is sufficiently large

To properly assess when IN-RIL is more efficient (ratio > 1), we need to solve:

$$\frac{m_{\text{opt}}}{1 + m_{\text{opt}}} \cdot \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}}} > 1$$

Rearranging, we get:

$$\Delta_{\text{IL-RL}} > L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) \cdot \left(1 - \frac{m_{\text{opt}}}{1 + m_{\text{opt}}}\right) = \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{1 + m_{\text{opt}}}$$

E.1.2 Key Insights

1. **Asymptotic Behavior:** As $m_{\text{opt}} \rightarrow \infty$, the efficiency condition approaches $\Delta_{\text{IL-RL}} > 0$. This means with very large interleaving ratios, even a small positive regularization benefit makes IN-RIL more efficient.
2. **Impact of Interleaving Ratio:** For any finite m_{opt} , IN-RIL includes an overhead factor of $\frac{1+m_{\text{opt}}}{m_{\text{opt}}}$ that must be overcome by the regularization benefit.
3. **Alternative View:** We can rewrite the ratio as:

$$\frac{T_{\text{RL-only}}}{T_{\text{IN-RIL,total}}} = \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}} + \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{m_{\text{opt}}}}$$

This form explicitly shows the penalty term $\frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{m_{\text{opt}}}$, which decreases as m_{opt} increases.

E.1.3 Practical Implications

Our theoretical analysis provides important practical guidance:

1. **Optimal Interleaving Ratio:** There is a trade-off in setting m_{opt} :
 - Small m_{opt} (e.g., $m_{\text{opt}} = 1$): IN-RIL needs $\Delta_{\text{IL-RL}} > \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{2}$ to be more efficient
 - Large m_{opt} (e.g., $m_{\text{opt}} = 9$): IN-RIL needs $\Delta_{\text{IL-RL}} > \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{10}$ to be more efficient
 - Very large m_{opt} : IN-RIL approaches the behavior of RL-only but retains modest regularization benefits
2. **Environment Interaction Efficiency:** If we consider only RL updates (environment interactions):

$$\frac{T_{\text{RL-only}}}{T_{\text{IN-RIL,RL}}} = \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*) - \Delta_{\text{IL-RL}}}$$

This ratio is greater than 1 whenever $\Delta_{\text{IL-RL}} > 0$, showing that IN-RIL always requires fewer environment interactions when there is any positive regularization benefit.

3. **Practical Recommendation:** Based on our empirical evaluations across multiple benchmarks, interleaving ratios between 3 and 5 typically provide the best balance. This aligns with our theory: with $m_{\text{opt}} = 4$, IN-RIL is more computationally efficient when $\Delta_{\text{IL-RL}} > \frac{L_{\text{RL}}(\mathcal{L}_{\text{RL}}(\theta_0) - \mathcal{L}_{\text{RL}}^*)}{5}$, a threshold often satisfied in practice.

E.1.4 Empirical Validation

Our experiments confirm the theoretical predictions:

- Across our benchmark tasks, IN-RIL demonstrated significant improvements in sample efficiency, significantly reducing required interactions
- The largest efficiency gains occurred in tasks where the estimated regularization benefit $\Delta_{\text{IL-RL}}$ was highest, exactly as predicted by our theory
- The relationship between efficiency gains and interleaving ratio matched our theoretical expectations, with diminishing returns for very large ratios

F Supplementary Experiments

F.1 Task Rollouts

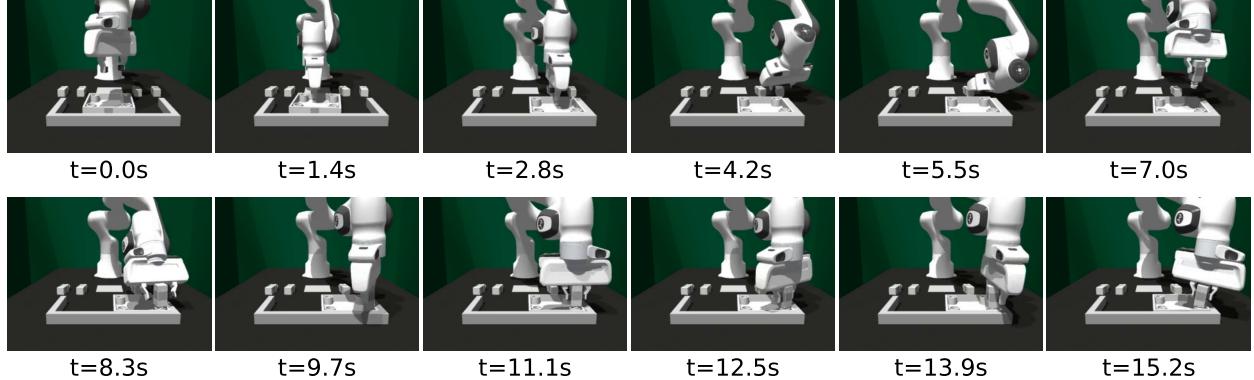


Figure 10: A successful rollout example of the One-Leg (Low) furniture assembly task

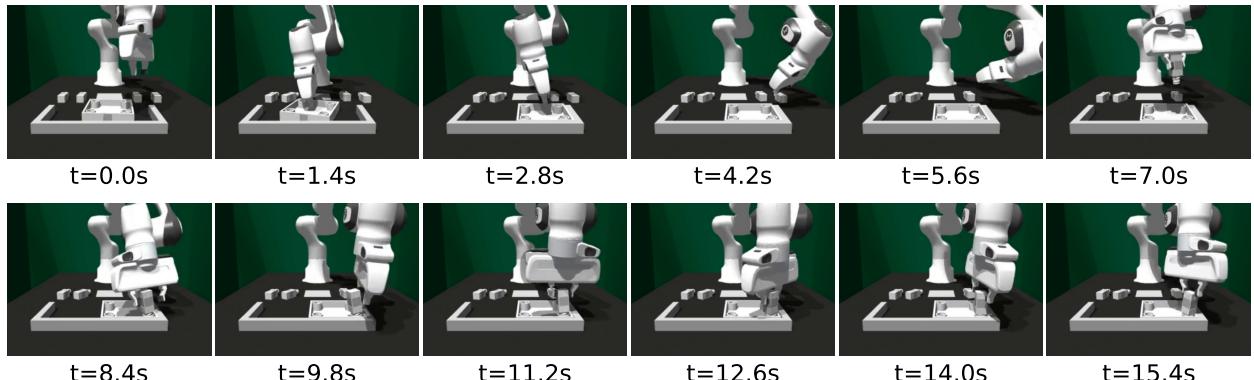


Figure 11: A successful rollout example of the One-Leg (Med) furniture assembly task

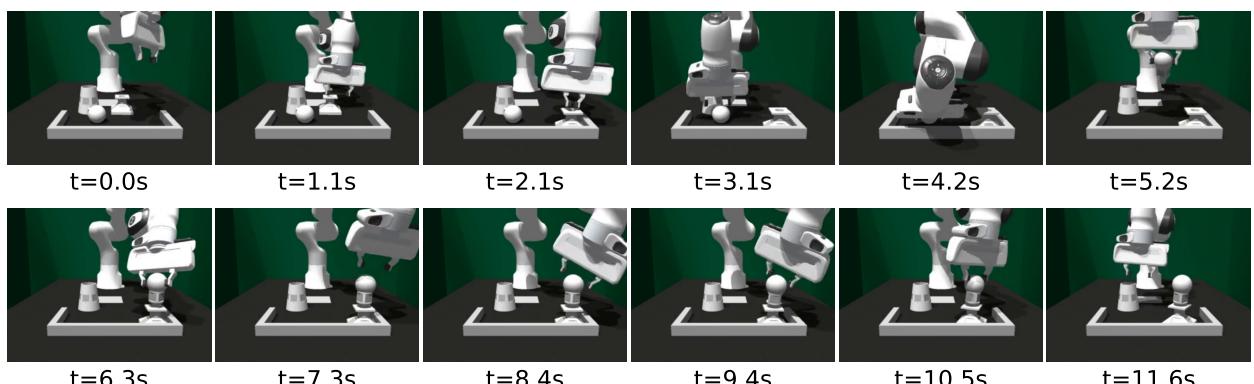


Figure 12: A successful rollout example of the Lamp (Low) assembly task

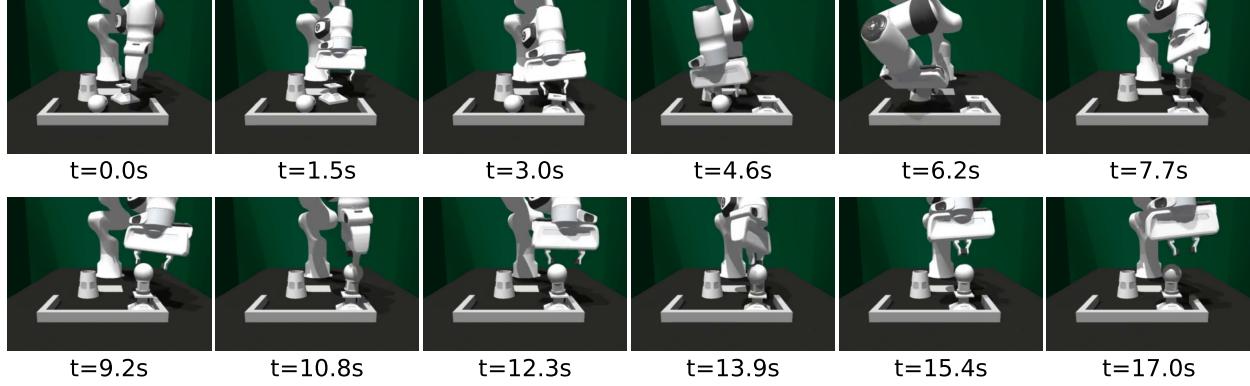


Figure 13: A successful rollout example of the Lamp (Med) task

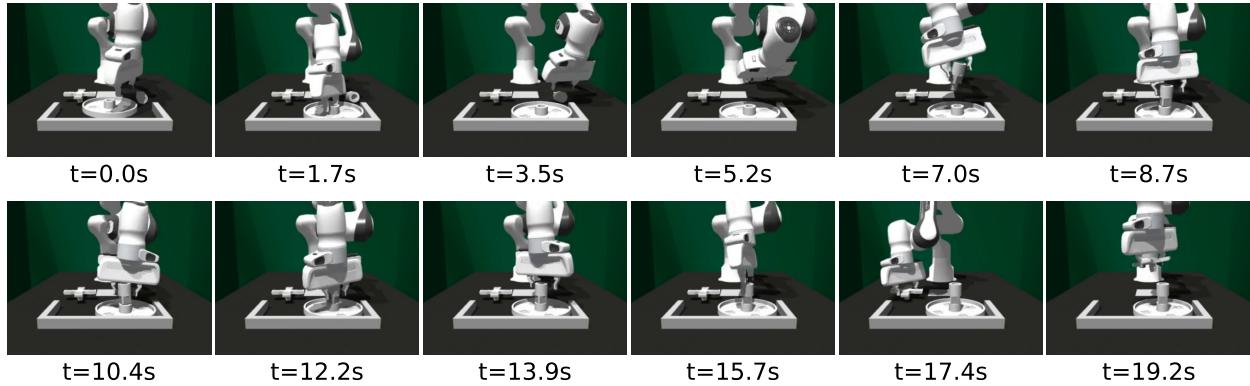


Figure 14: A successful rollout example of the Round-Table assembly task

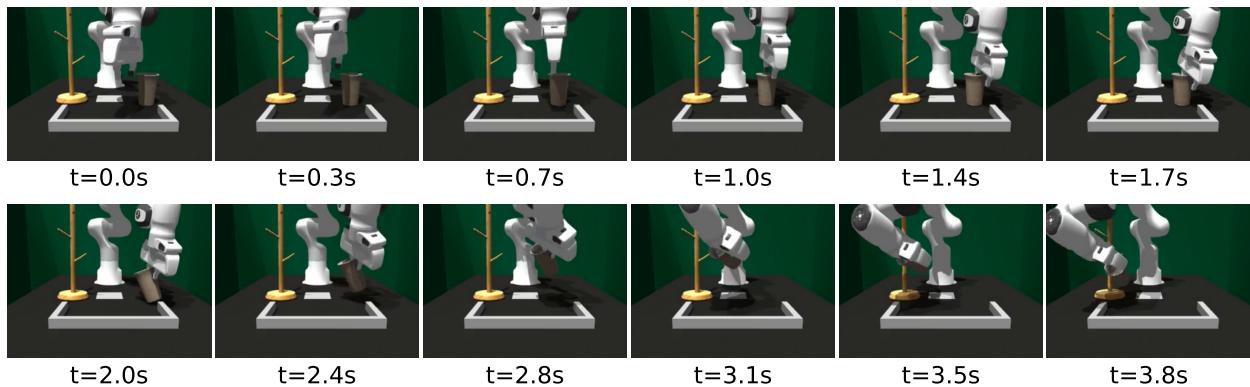


Figure 15: A successful rollout example of the Mug-Rack task

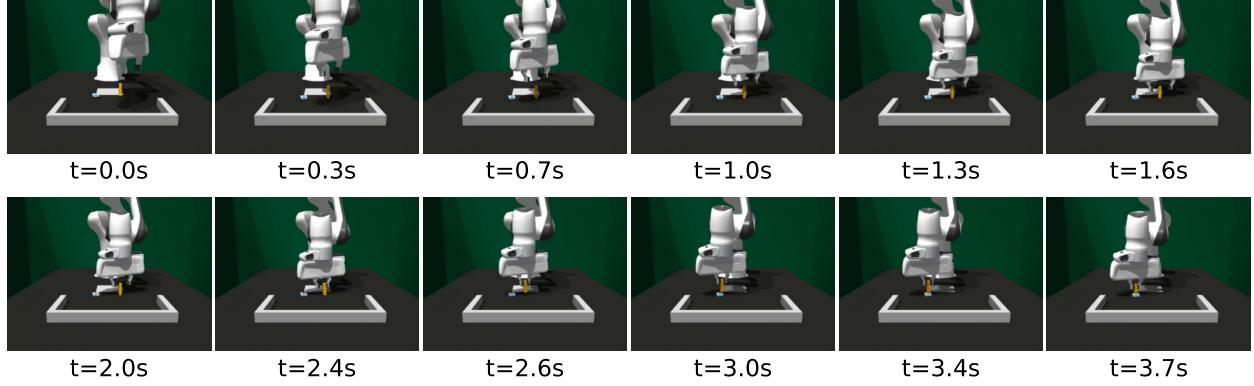


Figure 16: A successful rollout example of the Peg-in-Hole task

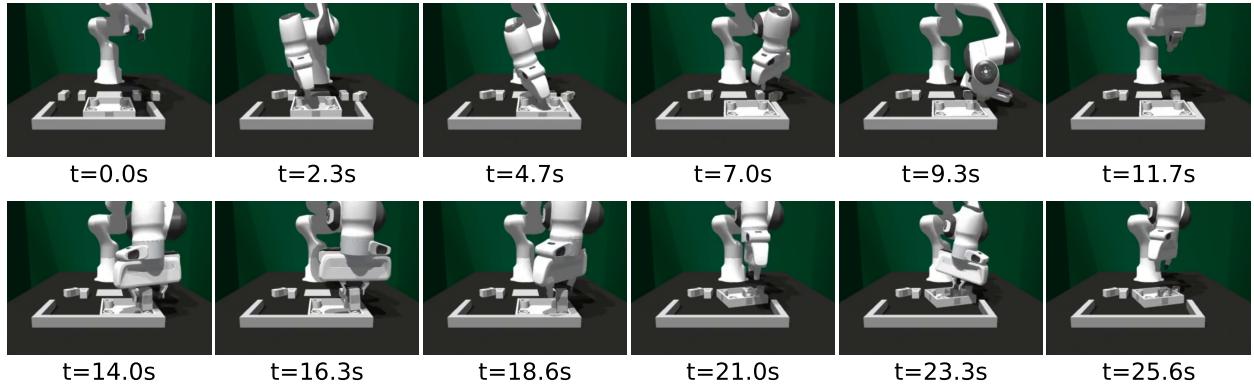


Figure 17: A failed rollout example of the One-Leg (Med) furniture assembly task

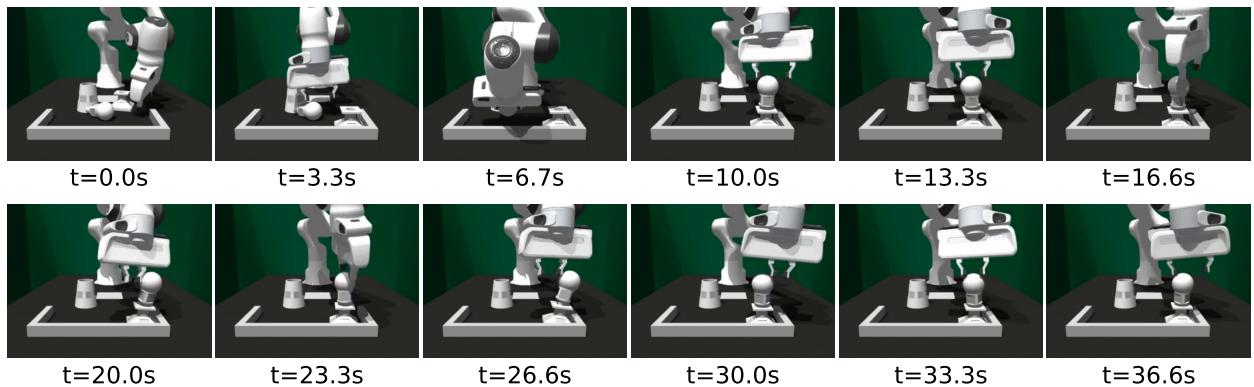


Figure 18: A failed rollout example of the Lamp (Med) assembly task

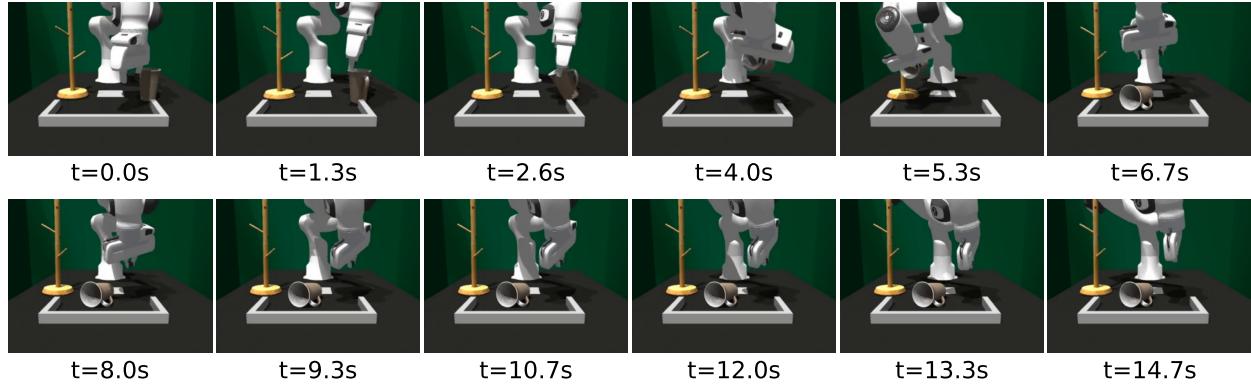


Figure 19: A failed rollout example of the Mug-Rack assembly task

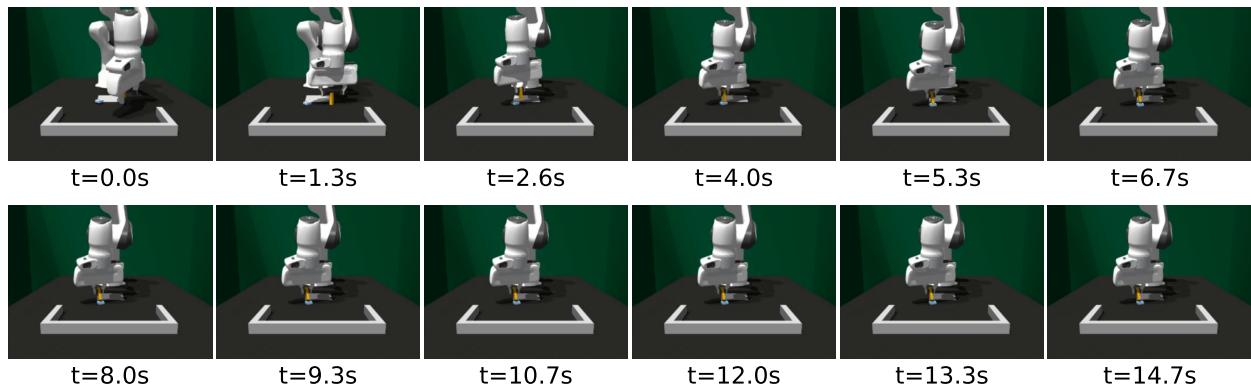


Figure 20: A failed rollout example of the Peg-in-Hole task