SCaR: Refining Skill Chaining for Long-Horizon Robotic Manipulation via Dual Regularization

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Long-horizon robotic manipulation tasks typically involve a series of interrelated sub-tasks spanning multiple execution stages. Skill chaining offers a feasible solution for these tasks by pre-training the skills for each sub-task and linking them sequentially. However, imperfections in skill learning or disturbances during execution can lead to the accumulation of errors in skill chaining process, resulting in execution failures. In this paper, we investigate how to achieve stable and smooth skill chaining for long-horizon robotic manipulation tasks. Specifically, we propose a novel skill chaining framework called **S**kill **C**haining via Du**a**l **R**egularization (**SCaR**). This framework applies dual regularization to sub-task skill pre-training and fine-tuning, which not only enhances the *intra-skill dependencies* within each sub-task skill but also reinforces the *inter-skill dependencies* between sequential sub-task skills, thus ensuring smooth skill chaining and stable long-horizon execution. We evaluate the SCaR framework on two representative long-horizon robotic manipulation simulation benchmarks: IKEA furniture assembly and kitchen organization. Additionally, we conduct real-world validation in desktop robot pick-and-place tasks. The experimental results demonstrate that with the support of SCaR, the robot performs long-horizon tasks with a higher success rate than relevant baselines and is more robust to perturbations.

1 Introduction

Long-horizon robotic manipulation tasks are characterized by sequences of diverse and interdependent sub-tasks, which makes it crucial to maintain the stability of multi-stage sequential execution. For instance, in the robotic assembly of a stool (Fig. 1) involving two sub-tasks of leg installation, overall success is evaluated based on both the sequential installation success and factors affecting the assembly within environmental constraints. Although recent advances in deep reinforcement learning (RL) and imitation learning (IL) show promise in training robots for such complex tasks [1, 2, 3, 4, 5, 6, 7], managing long-horizon tasks with a scratch RL or IL policy remains challenging due to computational demands, extensive exploration, and intricate step dependencies [8, 9]. Skill chaining, which involves decomposing long-horizon tasks into smaller sub-tasks, pre-training skills for each, and executing them sequentially, offers a practical solution [10, 11]. However, as shown in Fig. 1(a)(b), such methods tend to fail when sub-task skills are insufficiently trained or unexpected states arise due to disturbances, especially when applied to high-degree-of-freedom robots performing contact-rich, long-horizon tasks. [12, 13, 14, 15, 16, 17].

In this paper, we argue that the coordination and enhancing of dependencies within and between sub-task skills is necessary for stable and smooth skill chaining of long-horizon robotic manipulation [10]. For instance, as depicted in Fig. 1 (a)(b), the robot must consider following two points to ensure the overall task is accomplished: 1) ensuring the gripper consistently grasps and installs the stool leg

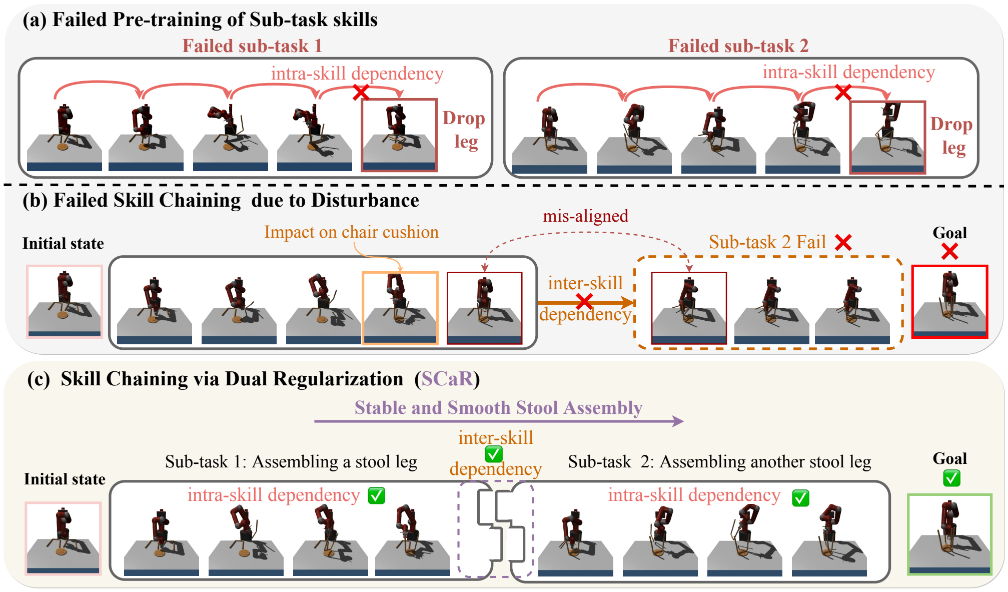


Figure 1: Illustration of the problem setting and the motivation of SCaR, using the example of a stool assembly task with two sub-tasks.

stably within each sub-task skill range. and 2) ensuring the terminal state of previous skill aligns with the initial state of next skill for smooth skill chaining. We define the above two points as *intra-skill dependencies* between sequential actions within each sub-task skill and *inter-skill dependencies* between sequential sub-task skills, respectively. In this context, we propose a novel robotic skill chaining framework, **S**kill **C**haining via Du**a**l **R**egularization (**SCaR**). This framework enhances the aforementioned dependencies alternately through dual regularization during sub-task skill learning and chaining, aiming to provide stability for the execution of long-horizon robotic manipulation.

Specifically, in the pre-training phase of each sub-task skill, we propose the *adaptive sub-task skill learning* shceme, which employs a two-part policy learning objective that focuses on what sub-tasks the robot should perform (via RL) and how the robot should perform that task (via IL), and utilizes a novel adaptive equilibrium scheduling (AES) regularization to balance these two parts based on the robot's learning progress. This process aims to reinforce the *intra-skill dependencies*, ensuring a coherent sequence of actions in each sub-task skill. Subsequently, *bi-directional adversarial learning* is introduced in the fine-tuning phase of SCaR for better chaining sequential sub-task skills. This mechanism uses bi-directional regularization to bring the terminal state of the current skill close to the initial state of its successor, and also to bring the initial state of the successor close to the terminal state of the current skill. This bi-directional alignment aims to reinforce robust *inter-skill dependencies* between sequential skills. Through the two innovative designs described, SCaR ensures coordination between the *intra-skill* and *inter-skill* dependencies, provides dual constraints for skill learning and skill chaining, as described in Fig. 1 (c), leading to a smooth skill chaining from the inside **(within the sub-task skills)** to the outside **(between sub-task skills)**. Experimental results show that compared to scratch-training and skill chaining baselines, SCaR provides better task execution performance and stronger robustness to environmental perturbations in various long-horizon and contact-rich robotic manipulation simulation tasks. In addition, SCaR achieves higher task success rates in long-horizon real-robot pick-and-place tasks compared to previous skill chaining method.

The principal contributions of our work are delineated as follows: **1)** We propose a novel robotic skill chaining framework via dual regularization, SCaR, for smoothly executing long-horizon manipulation tasks. **2)** We introduce an adaptive sub-task skill learning scheme that acts as a regularization to enhance *intra-skill dependencies* between sequential actions within each sub-task skill. **3)** We develop a bi-directional adversarial learning mechanism that serves as a regularization for reinforcing *inter-skill dependencies* between sequential sub-task skills. **4)** In all eight simulated long-horizon robotic manipulation tasks, SCaR performs significantly better than scratch-training and skill chaining baselines. In addition, SCaR also shows better task completion performance compared to skill chaining baseline in real-robot long-horizon pick-and-place experiments. Video demonstrations are available at: `https://tinyurl.com/4333d6np`.

2 Related Work

2.1 Long-horizon Robotic Manipulation

Training robots from scratch for complex, long-horizon tasks using reinforcement learning (RL) and imitation learning (IL) is challenging due to computational demands and distributional errors. Solutions involve decomposing tasks into reusable sub-tasks [18]. Typically, such algorithms consist of a set of sub-policies that can be obtained through various methods, such as unsupervised exploration [19, 20, 21, 22, 23], learning from demonstrations [5, 6, 24, 25], and predefined measures [26, 27, 28, 29, 14]. Despite the merits of each of these approaches, they do not address well the challenges of long-horizon robot manipulation in environments that are object-rich, contact-rich, and characterized by multi-stage tasks [28, 29, 14]. Thus, even when pre-trained skills are provided, ensuring a smooth connection between manipulation policies remains a formidable challenge.

2.2 Skill Chaining for Long-horizon Tasks

Previous skill chaining methods for long-horizon tasks mainly focus on updating each sub-task policy to encompass the terminal state of the previous policy [11, 14, 30], implementing option chains [11, 31, 32] to forge logical skill sequences, or utilizing modulated skills to facilitate smoother transitions [33, 34, 35, 36, 14, 16]. However, these methods, while effective, often lead to a broad range of skill start and end states, a challenge in complex robotic manipulation tasks. T-STAR [15] is closely related to our work, addressing this by regularizing the learning process with a discriminator to control the expansion of the terminal state space. However, it focuses only on uni-directional dependencies between skills and ignores intra-skill dependencies within sub-task skills under long-horizon goals. Sequential Dexterity [17] centers on dexterous hand manipulation, introducing an optimization process to backpropagate long-term rewards across a policy chain. However, its scope still primarily emphasizes strengthening the dependencies between sub-task skills. GSC [37] attempts to solve skill chaining by employing diffusion models. It trains and chains primitive skills (pick, place, push, pull) through a Transformer-based skill diffusion model. However, due to the use of Transformer-based techniques, GSC requires high computational resources and cannot scale well to task environments with object-rich and contact-rich conditions. Our method instead employs simple and intuitive dual regularization constraints based on the lightweight policy network. By coordinating the dependencies within and between skills, we achieve refinement within sub-task policies and bi-directional alignment between them. This allows for stable skill chaining while also being scalable to various long-horizon manipulation tasks.

3 Preliminaries

Among several related works on skill chaining, we consider a challenging yet practical problem setting that *deals with long-horizon manipulation tasks through a combination of reinforcement learning (RL) and imitation learning (IL)*. In each sub-task in the long-horizon task, we consider robotic agents acting within a finite-horizon Markov Decision Process [38] $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, d_{\mathcal{I}}, T)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(s'|s,a)$ is the transition function, $r(s,a,s')$ is the reward function, γ is the discount factor, $d_{\mathcal{I}}$ is the initial state distribution, and T is the episode horizon of sub-task. We define a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maps states to actions and correspondingly moves the robotic agent to a new state according to the transition probabilities. This sub-task policy is trained to maximize the expected sum of discounted rewards $\mathbb{E}_{(s,a)\sim\pi}[\sum_{t=1}^{T} \gamma^t r(s_t, a_t, s_{t+1})]$. We assume that each sub-task policy has an initial state set $\mathcal{I} \in \mathcal{S}$ and a terminal state set $\beta \in \mathcal{S}$, where the initial set $\mathcal{I}$ contains all the initial states that lead to the successful execution of the policy and the terminal state set β contains all the final states of the successful execution. The environment provides the environmental feedback for each step taken by the agent and success metrics for each sub-task, derived from the terminal states of sub-task policy. For instance, as shown in Fig. 1(c), the alignment of the back and legs of the stool triggers the connect action and the realization of the sub-task goal, which indicates the successful completion of the sub-task. Additionally, we posit that during each sub-task policy learning, the agent receives a set of pre-defined expert demonstrations, $\mathbb{D}^E = \{\tau_1^E, \ldots, \tau_N^E\}$, to facilitate the IL process. Here, N represents the number of episodes, and each demonstration comprises a sequence of state-action pairs, $\tau^E = (s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T)$.

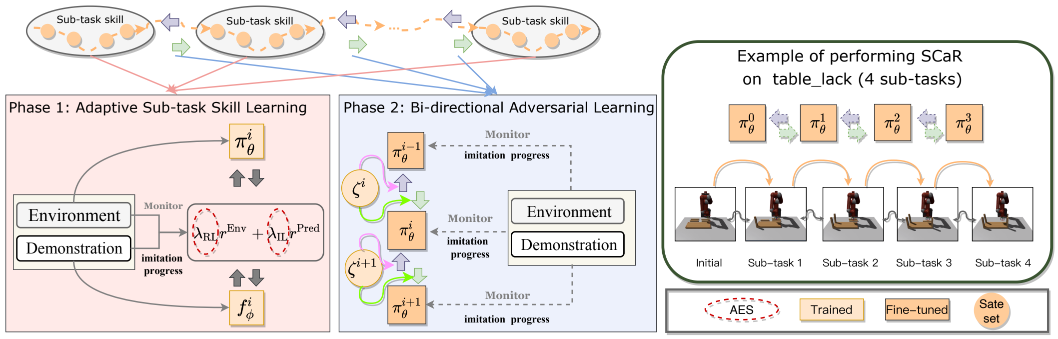


Figure 2: The Pipeline of **S**kill **C**haining via Du**a**l **R**egularization (SCaR). **(Left) Phase 1**: Sub-task skill pre-training ( ) merges environmental feedback and expert guidance, using adaptive equilibrium scheduling (AES) regularization to balance learning, which enhances intra-skill dependencies within skills. **(Middle) Phase 2**: Bi-directional discriminators ( ) coupled with AES to fine-tune pre-trained sub-task skills, as regularization for reinforcing inter-skill dependencies. **(Right) Evaluation**: Evaluation of SCaR on long-horizon manipulation.

4 Method

In Section 4.1, we present the pipeline of the **SCaR** framework. Sections 4.2 and 4.3 provide further elaboration on the key design elements.

4.1 Overall Pipeline

As illustrated in Fig. 2, the **SCaR** framework has two phases: **(a) pre-training (adaptive sub-task skill learning) and (b) fine-tuning (bi-directional adversarial learning)**. In the pre-training phase, the agent co-learns sub-task skills by integrating environmental feedback and expert demonstrations. In the fine-tuning phase, it refines these skills through bi-directional adversarial learning, enabling sequential integration of sub-task skills. After fine-tuning, SCaR can smoothly chain sub-task skills to complete long-horizon manipulation tasks. Specific modules and mechanisms for these phases are detailed in Sections 4.2 and 4.3.

4.2 Adaptive Sub-task Skill Learning

Weighted Reward Function To learn sub-task skills better, we combine goal-conditional RL and generative adversarial imitation learning (GAIL) [39], to pre-train skills that enable the agent to perform challenging sub-tasks in a desired expert behavioral style [40, 15]. More specifically, we consider the weighted reward function that is used to train each sub-task policy π_i^θ consists of two components specifying: *what sub-task the agent should perform* - learning from environmental feedback, and 2) *how the agent should perform that task* - learning from expert demonstrations:

$$r(s_t, a_t, s_{t+1}; \phi) = \lambda_{\mathrm{RL}} r_i^{\mathrm{Env}}(s_t, a_t, s_{t+1}, g) + \lambda_{\mathrm{IL}} r_i^{\mathrm{Pred}}(s_t, a_t; \phi). \tag{1}$$

As shown in Eq. 1, the first component is represented by a task-specific reward $r_i^{\mathrm{Env}}(s_t, a_t, s_{t+1}, g)$, which defines general objectives that the agent should satisfy to fulfill a given sub-task goal g for current MDP $\mathcal{M}$ (e.g. assembling a stool leg). The second component is represented through a learned task-agnostic predict-reward $r_i^{\mathrm{Pred}}(s_t, a_t; \phi)$, which specifies manipulation details of the behaviors that the agent should adopt when performing the sub-task (e.g., the expert way to grab a stool leg and attach it), and $r_i^{\mathrm{Pred}}(s_t, a_t; \phi)$ is the predicted reward by a least-square GAIL discriminator f_ϕ^i [41, 40, 15], which is more stable than the standard GAIL objective using the sigmoid cross-entropy loss function. Therefore, the predicted reward is:

$$r_i^{\mathrm{Pred}}(s_t, a_t; \phi) = \max[0, 1 - 0.25 \cdot [f_\phi^i(s_t, a_t) - 1]^2]. \tag{2}$$

We adopt the training objective of the least-squares GAIL discriminator with a gradient penalty term [42, 43], This penalty term mitigates the instability of the training dynamics due to the interplay between the discriminator and the policy [40], as follows:

$$\mathrm{argmin}_{f_\phi^i} \mathbb{E}_{(s)\sim\mathbb{D}^E}[(f_\phi^i(s) - 1)^2] \quad + \mathbb{E}_{(s)\sim\pi_\theta^i}[(f_\phi^i(s) + 1)^2] + \frac{\eta^{\mathrm{gp}}}{2}\mathbb{E}_{(s)\sim\mathbb{D}^E}[\|\nabla_s f_\phi^i(s)\|^2], \tag{3}$$

where η^{gp} is a manually-specified coefficient. The scales of r^{Env} and r^{Pred} in previous related works are set by fixed weights and linearly combined into the final reward function [40, 15]. This could lead to the agent rigidly imitating experts and curbing self-exploration, finding it difficult to adjust intra-skill dependencies and adapt to dynamic task perturbations. We propose a principle to counter this: ***If the agent fails to imitate the expert's demonstration well, it should shift focus to self-learning from the environment. Conversely, effective imitation should continue, focusing on the expert to mitigate low sample efficiency in reinforcement learning.*** Accordingly, we extend the automatic discount scheduling (ADS) solution [9] to our problem setting, and propose adaptive equilibrium scheduling (AES) to regularize the scales of r^{Env} and r^{Pred} in sub-task skill learning for adaptive scheduling the focus of reinforcement and imitation learning, as shown in Fig. 3.

Adaptive Equilibrium Scheduling (AES) Regularization Specifically, AES balances the scales of r^{Env} and r^{Pred} during the learning process of each skill through adaptive scheduling of λ_{RL} and λ_{IL}, according to how well the agent imitates the expert's demonstration. To capture the agent's imitation progress, AES refers to the solution in ADS [9] and uses the imitation identifier Φ to continuously monitor whether the agent is imitating the expert demonstration well enough.

At the beginning of training, the agent is assigned two initial balance factors $\lambda_{\mathrm{RL}} = \alpha, \lambda_{\mathrm{IL}} = 1 - \alpha$, where base exponent $\alpha \in [0, 1]$. We set $\alpha = 0.5$ in the experiments and the agent is assigned two identical balance factors $\lambda_{\mathrm{RL}} = \lambda_{\mathrm{IL}} = 0.5$[1], indicating that at the beginning of learning, the agent imitates the expert's behavior with the same weight as the behavior of environment exploration according to the task goal. As training progresses, the imitation progress recognizer Φ is queried periodically to monitor the progress of the agent's imitation of the expert's behavior. Φ receives the agent's collected trajectories and infers the agent's current imitation progress $p \in [0, T)$, where p in an integer and T is the step of the entire episode.

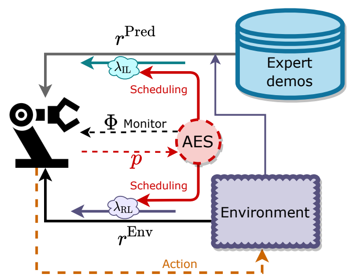


Figure 3: AES regularization for sub-task skill learning.

The construction of Φ, with reference to ADS, first requires the construction of a sequence $\mathbf{Q}(q_1, \ldots, q_T)$, where $q_i = \mathrm{argmin}_j c(s_i, s_j^E)$ is the index of the nearest neighbor of s_i in τ^E, c is the cosine similarity. The progress alignment between τ and τ_j^E is measured as the length of the longest increasing subsequence (LIS) in $\mathbf{Q}$, denoted as $LIS(\tau, \tau^E)$. Specifically, the agent's imitation progress p is increased by 1 if the following inequality holds:

$$\max_{\acute{\tau}^E \in \mathbb{D}^E} LIS(\tau_{1:p+1}, \acute{\tau}^E_{1:p+1}) \geq \rho \times \min_{\acute{\tau}^E, \acute{\acute{\tau}}^E \in \mathbb{D}^E} LIS(\acute{\tau}^E_{1:p+1}, \acute{\acute{\tau}}^E_{1:p+1}), \tag{4}$$

where $\acute{\tau}^E \neq \acute{\acute{\tau}}^E$, the subscript $1 : p + 1$ denotes the first $p + 1$ steps of the trajectory, and $\rho \in [0, 1]$ controls the strictness of the imitation progress monitoring. This suggests that the similarity of the agent trajectory to its best matching expert trajectory at time step $p+1$ exceeds the minimal similarity criterion within the expert demonstration. See Appendix B for detailed explanation of AES.

After obtaining the current imitation progress p of the agent, AES then adopts a mapping function $\varphi_\lambda(p)$ to schedule the two new balance discount factors λ_{RL} and λ_{IL}. Straightforward idea of setting $\varphi_\lambda(p)$ is that **If p is larger and reaches a certain threshold, i.e., the agent is able to imitate the expert behavior well, then the more the agent tends to imitate the expert's behavior in subsequent training, and vice versa.** Therefore, we set the threshold as $\frac{T}{2}$. If $p \in [0, \frac{T}{2})$, we propose $\varphi_\lambda(p) = 1 - e^{\left(-\frac{p}{k}\right)}$; if $p \in [\frac{T}{2}, T)$, we propose $\varphi_\lambda(p) = e^{\left(-\frac{p - \frac{T}{2}}{k}\right)}$, where k is used to flatten the curve of the mapping function. Then λ_{RL} and λ_{IL} are scheduled to be :

$$\begin{cases} \lambda_{\mathrm{RL}} = \alpha^{\varphi_\lambda(p)}, \lambda_{\mathrm{IL}} = 1 - \alpha^{\varphi_\lambda(p)}. & \text{if } p \in [0, \frac{T}{2}) \\ \lambda_{\mathrm{IL}} = \alpha^{\varphi_\lambda(p)}, \lambda_{\mathrm{RL}} = 1 - \alpha^{\varphi_\lambda(p)}. & \text{if } p \in [\frac{T}{2}, T) \end{cases} \tag{5}$$

Consequently, the RL and IL components of sub-task skill learning can be adaptively scheduled and regularized through AES, effectively enhancing *intra-skill dependencies* between sequential actions. The pseudo-code of adaptive sub-task skill learning is outlined in Algorithm 1 in Appendix A.1.

[1] We further explore what effect different α would have in the Ablation Experiments.

4.3 Bi-directional Adversarial Learning for Skill Chaining

Executing pre-trained sub-task skills sequentially without considering inter-skill dependencies may lead to failure. To address this, we propose bi-directional adversarial learning to further refine and better integrate sequential sub-task skills. The pseudo-code of bi-directional adversarial learning is outlined in Algorithm 2 in Appendix A.2.

Bi-directional Regularization In contrast to previous uni-directional regularization schemes that only augment the initial state set $\mathcal{I}i$ or regularize the terminal state set β_i [12, 15], we impose the *bi-directional constraints* ($\mathcal{C}_1$,$\mathcal{C}_2$) on inter-skill dependencies, facilitating smooth skill chaining, as shown in Fig 4. With the bi-directional constraint, we implement the bi-directional adversarial learning, centered on the joint training of a *bi-directional discriminator*, denoted by ζ_ω^i , which is adept at distinguishing between the terminal state set of the preceding policy and the initial state set of the subsequent policy. The bi-directional constraints $\mathcal{C}_1$,$\mathcal{C}_2$ are defined as Eq. 10:

$$\begin{aligned} &\textbf{next initial} \rightarrow \textbf{previous terminal:} \quad \mathcal{C}_1 = \mathbb{E}_{s_\mathcal{I} \sim \mathcal{I}_i}[\zeta_\omega^i(s_\mathcal{I}) - 1]^2 + \mathbb{E}_{s_T \sim \beta_{i-1}}[\zeta_\omega^i(s_T)]^2 \\ &\textbf{previous terminal} \rightarrow \textbf{next initial:} \quad \mathcal{C}_2 = \mathbb{E}_{s_T \sim \beta_i}[\zeta_\omega^i(s_T) - 1]^2 + \mathbb{E}_{s_\mathcal{I} \sim \mathcal{I}_{i+1}}[\zeta_\omega^i(s_\mathcal{I})]^2 \end{aligned} \tag{6}$$

ζ_ω^i is trained for each policy to minimize the objective function[2]:$\mathcal{L}_i(\omega) = \frac{1}{2}\mathcal{C}_1 + \frac{1}{2}\mathcal{C}_2$. Guided by ζ_ω^i, the bi-directional adversarial learning not only steers the terminal state set of the current policy towards the initial state set of the subsequent policy, but also ensures alignment of the initial state set of the subsequent policy with the terminal state set of current policy. This dual alignment establishes a balanced mapping between the initial and terminal states of sequential skills to reinforce inter-skill dependencies, ensure consistency and stability in multi-stage tasks, and guarantee smooth transitions between sequential skills. Accordingly, the *bi-directional regularization* can be added to the overall objective function of policy learning in the form of the following reward term: $r_i^{\mathrm{Bi}}(s;\omega) = \mathbb{1}_{s \in \beta_i}\zeta^{i+1}(s) + \mathbb{1}_{s \in \mathcal{I}_i}\zeta^{i-1}(s)$.

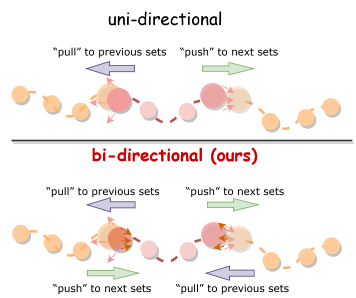


Figure 4: Bi-directional regularization for sub-task skill chaining.

Overall Objective Function So far, the objective function via dual regularization, i.e., AES regularization and bi-directional regularization, to pre-train, fine-tune and chain sub-task skills can be rewritten as a weighted sum of the individual reward terms:

$$r_i(s_t, a_t, s_{t+1}; \phi) = \underbrace{\lambda_{\mathrm{RL}} r_i^{\mathrm{Env}}(s_t, a_t, s_{t+1}, g) + \lambda_{\mathrm{IL}} r_i^{\mathrm{Pred}}(s_t, a_t; \phi)}_{\textbf{AES regularization}} \quad + \quad \underbrace{\lambda_{\mathrm{Bi}} r_i^{\mathrm{Bi}}(s_{t+1}; \omega)}_{\textbf{bi-directional regularization}} \quad , \tag{7}$$

where λ_{Re} is the weighting factor of the bi-directional regularization. The objective function features AES regularization and bi-directional regularization to enhance intra- and inter-skill dependencies. It enables the agent to adaptively pre-train skills that can solve different sub-tasks well through environmental feedback and expert guidance, and further fine-tune them through the bi-directional discriminator to achieve dual alignment between sequential skills. At the same time, the fine-tuned sub-task skills help to collect terminal and initial states to refine the bi-directional discriminator. This iterative process ensures smooth long-horizon task skill chaining.

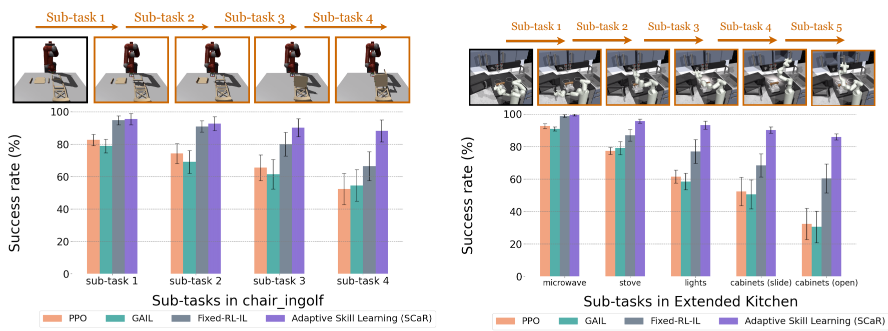
5 Experiments

5.1 Experiment Setup

We conduct simulation experiments on six IKEA furniture assembly tasks and two kitchen organization tasks, and also perform long-horizon pick-and-place experiments on the real Sagittarius K1 robot. Please refer to the Appendix for more detailed simulation experiment setup (Appendix G), network architecture (Appendix H), training details (Appendix I), more quantitative (Appendix D) and qualitative results (Appendix E) of the simulation tasks, and the real-robot experiments (Appendix F).

Furniture Assembly We conduct experiments in six IKEA furniture assembly tasks in [44]: *chair_agne*, *chair_bernhard*, *chair_ingolf*, *toy_table*, *table_lack* and *table_bjorkudden*.

[2]We explore the impact of different scales of $\mathcal{C}_1$ and $\mathcal{C}_2$ in Appendix D.3



(a) chair_ingolf (b) Extended Kitchen

Figure 5: Evaluation Performance of Sub-task Skill Learning. Best viewed zoomed.

1) ***chair_agne***: Two stool legs need to be picked up and aligned with the cross notches on the stool back. 2) ***chair_bernhard***: The two chair supports need to be taken and aligned with the slots at the bottom of the chair surface. 3) ***chair_ingolf***: Two chair supports and front legs need to be attached to the chair seat, which must then be secured to the chair back while avoiding collision with each other. 4) ***table_lack***: The four table legs need to be picked up and aligned with the corners of the tabletop. 5) ***toy_table***: The four table legs need to be picked up and aimed and inserted with the four notches on the table back. 6) ***table_dockstra***: After supporting the two bases with table leg, the table top needs to be mounted while preventing collision. For each assembly task, we define the assembly of individual parts as sub-tasks. We collect 200 demonstrations per sub-task using a procedural assembly policy for imitation learning. Each demonstration consists of 150 steps.

Kitchen Organization We use the Franka Kitchen tasks in D4RL [45] and collect 200 demonstrations per sub-task for imitation learning. Specifically, we refer to the kitchen task in [46] and further extend the task sequence: in the **Kitchen task**, the 7-DoF Franka Emika Panda arm needs to perform 4 sequential sub-tasks, namely *Turn on the microwave - Move the kettle - Turn on the stove - Turn on the light*. In the **Extended Kitchen task**, the robot needs to perform 5 sequential sub-tasks: *Turn on the microwave - Turn on the stove - Turn on the light - Slide the cabinet to the right - Open the cabinet*, in which the sub-tasks have a lower probability of switching and is more challenging.

Baselines We compare SCaR with the following two types of baselines:

Scratch Training: 1) **PPO** is a model-free RL algorithm [47] that utilizes environmental rewards to learn tasks from scratch. 2) **GAIL** [39] is an adversarial imitation learning method to learn tasks from scratch, with a trained discriminator for distinguishing state-action distributions of experts and agents. 3) **Fixed-RL-IL** [40] uses fixed-weight environmental rewards and GAIL rewards to train policies from scratch. 4) **SkiMo** [46] is a model-based hierarchical RL approach that learns dynamic skill models for predicting outcomes in downstream tasks, which is used to test if modularly skill chaining method can surpass model-based scratch-training method on long-horizon tasks.

Skill Chaining: 1) **Policy Sequencing** [12] focuses on sequentially expanding the initial sets in skill chaining. 2) **T-STAR** [15] incorporates a discriminator to uni-directionally regularize the terminal states of sub-skills in a skill chaining. 3) **SCaR w/o Bi** reference to T-STAR during the fine-tuning phase, only uni-directional regularization of the terminal state set is performed to verify the validity of the proposed bi-directional regularization. 4) **SCaR w/o AES** fixes the scales of the two reward terms at 0.5 at all times to verify the effectiveness of the proposed AES regularization.

5.2 Quantitative Results

Sub-task Skill Learning Performance First, we evaluate the proposed adaptive sub-task skill learning scheme in the sub-tasks of furniture assembly and kitchen organization. Specifically, we treat each sub-task as a separate task for policy learning and take the success rate of the trained policy tested in the reset sub-task as the criterion. All methods are trained in each sub-task with 5 random seeds, 150 million environment steps, and evaluated with the average success rate over 100 testing episodes. As shown in the Fig. 5, in *chair_ingolf* and Extended Kitchen tasks, even with the increase of objects in the environment and the increase of unpredictable perturbations, our proposed adaptive skill learning learns good sub-task skills and consistently maintains a task success rate of more than 85% in all stages of the sub-task. In contrast, the PPO (only RL rewards), GAIL (only IL

Table 1: Long-horizon tasks execution performance (varies by sub-task completion progress): *tasks with 2 sub-tasks progress by 0.5 per sub-task, *tasks with 4 sub-tasks by 0.25, *tasks with 5 sub-tasks by 0.2, and table_dockstra with 3 sub-tasks by 0.3, where 0.9 indicates completion of all tasks. Best viewed zoomed.

	Furniture Assembly							Kitchen Organization		
Method	*chair_agne*	*chair_bernhard*	*chair_ingolf*	*table_lack*	*toy_table*	*table_dockstra*	**All**	**Kitchen**	**E-Kitchen**	**All**
PPO (Scratch RL)	0.54 ± 0.18	0.42 ± 0.12	0.14 ± 0.03	0.09 ± 0.01	0.00 ± 0.00	0.31 ± 0.12	0.25 ± 0.15	0.13 ± 0.05	0.03 ± 0.00	0.08 ± 0.04
GAIL (Scratch IL)	0.31 ± 0.05	0.23 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.21 ± 0.04	0.12 ± 0.09	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Fixed-RL-IL	0.68 ± 0.12	0.53 ± 0.07	0.22 ± 0.08	0.21 ± 0.11	0.13 ± 0.02	0.43 ± 0.07	0.37 ± 0.15	0.33 ± 0.06	0.18 ± 0.02	0.26 ± 0.06
SkiMo	0.75 ± 0.09	0.62 ± 0.05	0.47 ± 0.03	0.58 ± 0.14	0.34 ± 0.06	0.62 ± 0.11	0.56 ± 0.11	0.57 ± 0.08	0.21 ± 0.04	0.39 ± 0.13
Policy Sequencing	0.89 ± 0.08	0.82 ± 0.09	0.77 ± 0.12	0.63 ± 0.28	0.45 ± 0.18	0.61 ± 0.14	0.70 ± 0.16	0.53 ± 0.11	0.36 ± 0.09	0.44 ± 0.09
T-STAR	0.92 ± 0.02	0.90 ± 0.04	0.89 ± 0.04	0.90 ± 0.07	0.71 ± 0.21	0.77 ± 0.09	0.85 ± 0.09	0.68 ± 0.13	0.48 ± 0.08	0.58 ± 0.10
SCaR w/o Bi	0.93 ± 0.04	0.92 ± 0.02	0.91 ± 0.01	0.93 ± 0.02	0.80 ± 0.10	0.79 ± 0.02	0.88 ± 0.05	0.75 ± 0.08	0.57 ± 0.14	0.66 ± 0.09
SCaR w/o AES	0.95 ± 0.03	**0.94** ± 0.03	0.93 ± 0.02	0.95 ± 0.04	0.85 ± 0.06	0.80 ± 0.03	0.91 ± 0.05	0.77 ± 0.07	0.61 ± 0.13	0.74 ± 0.05
SCaR (Ours)	**0.98** ± 0.02	**0.96** ± 0.04	**0.95** ± 0.03	**0.97** ± 0.03	**0.92** ± 0.05	**0.88** ± 0.02	**0.94** ± 0.03 (12% ↑)	**0.84** ± 0.16	**0.73** ± 0.17	**0.78** ± 0.12 (18% ↑)

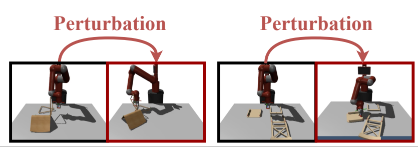
rewards), and Fixed-RL-IL (fixed RL and IL reward weights) baselines fail to maintain good sub-task success rates as the number of sub-task stages increases. This result well validates that our proposed adaptive weighted reward function based on AES regularization enhances *intra-skill dependencies* for multi-stage sub-task learning and brings effectiveness and stability.

Long-horizon Execution Performance We then demonstrate the performance of SCaR in performing 8 long-horizon tasks in IKEA furniture assembly and kitchen organization. Table 1 shows the mean and standard deviation for these 8 tasks across 200 testing episodes with 5 different seeds. The PPO and GAIL baselines show minimal success on tasks with 4 and 5 sub-tasks, indicating the difficulty of learning complex multi-stage tasks solely from reward signals or expert demonstrations. The fixed RL-IL baseline, although improved compared to PPO and GAIL, mostly completed only one sub-task, which highlights the limitations of using fixed RL and IL reward weights in long-horizon tasks. While SkiMo achieves better success rates than model-free methods by building dynamic skill models, its performance remains inconsitent on long-horizon tasks due to its scratch learning nature. The performance of these scratch baselines demonstrates the importance of effective staged sub-task learning for long-horizon tasks. The results in Table 1 further highlight the superiority of the SCaR framework. By reinforcing *intra- and inter-skill dependencies*, task success rates are considerably higher than previous skill chaining approaches such as Policy Sequencing and T-STAR, which primarily address uni-directional inter-skill dependencies. Compared to T-STAR, SCaR increases average success rates by more than 12% on six furniture assembly tasks and 18% on two kitchen organization tasks.[3].

5.3 Robustness to Perturbations

Perturbation tests are conducted to evaluate the robustness of skill chaining for two furniture assembly tasks. As shown in the top figure of Table 2, for the *chair_bernhard* task, the perturbation involves applying external joint torque to the robotic arm, moving the chair back before assembling the second support. For the *chair_ingolf* task, the perturbation is applied by exerting external torque on the robotic arms, causing them to move slightly before mounting the assembled chair seat to the chair back. The results in Table 2 highlight the detrimental impact of environmental perturbations on the success rates of baseline methods during the execution of multiple sub-task skills. Methods like Policy Sequencing and T-STAR, which focus solely on inter-skill dependencies through uni-directional regularization, struggle to complete tasks after perturbations. In contrast, SCaR, demonstrates more robust performance even under unseen perturbations. These results further support the advantages of our proposed *dual regularization* for stable skill chaining on long-horizon manipulation tasks.

Table 2: Comparison of the robustness of skill chaining in perturbed environments.



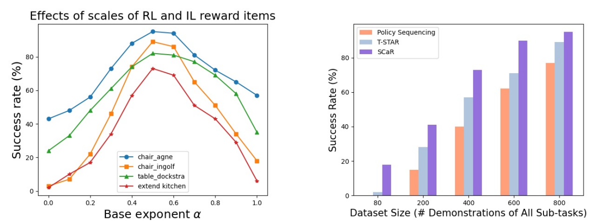
	chair_bernhard		*chair_ingolf*	
Method	**No Perturb**	**Perturb**	**No Perturb**	**Perturb**
Policy Sequencing	0.82 ± 0.09	0.51 ± 0.04	0.77 ± 0.12	0.50 ± 0.10
T-STAR	0.90 ± 0.04	0.60 ± 0.08	0.89 ± 0.04	0.59 ± 0.04
SCaR w/o Bi	0.92 ± 0.02	0.65 ± 0.11	0.91 ± 0.01	0.63 ± 0.05
SCaR w/o AES	0.94 ± 0.03	0.74 ± 0.09	0.93 ± 0.02	0.71 ± 0.07
SCaR (Ours)	**0.96** ± **0.04**	**0.85** ± **0.11**	**0.95** ± **0.03**	**0.80** ± **0.13**

[3]The overall increase is somewhat modest due to averaging the success rates of the 2, 3, and 4 sub-tasks and the 4 and 5 sub-tasks, respectively.

5.4 Ablations and Analysis

We perform ablation studies to explore the important factors that affect the performance of SCaR.

Modular Ablation We investigate how the adaptive sub-task skill learning and bi-directional adversarial learning impact skill chaining through SCaR w/o Bi and SCaR w/o AES. As shown in Table 1, without bi-directional regularization, SCaR w/o Bi experiences significant performance drops in tasks with more than two sub-tasks but still outperforms T-STAR. This is because SCaR w/o Bi maintains the adaptive scheduling of AES during sub-task skill learning, underscoring the importance of focusing on the *intra-skill dependencies* between successive actions. Similarly, the absence of AES regularization reduces SCaR w/o AES's performance, though it still maintains stable outcomes. This underscores the importance of reinforcing *inter-skill dependencies* on long-horizon tasks and reaffirms the contribution of bi-directional regularization. As shown in Table 2, SCaR w/o Bi, though slightly more robust than T-STAR due to the presence of AES, still faces challenges in adapting to perturbations and maintaining stable skill chaining because of its uni-directional fine-tuning limitations. SCaR w/o AES manages to maintain a certain level of performance stability under perturbations, thanks to bi-directional regularization, which ensures the bi-directional alignment of initial and terminal states between skills. The results show that the pre-trained skills via AES exhibit enhanced *intra-skill dependencies* within sub-tasks, and bi-directional regularization ensures stable long-horizon execution, even in the presence of perturbations, by reinforcing *inter-skill dependencies*.



(a) (b)

Figure 6: Ablation experiments. Best viewed zoomed.

Parametric Ablation We further investigate the impact of different scales of RL and IL reward terms, as well as the size of expert demonstration datasets. The effect of varying the base exponent α on task success rates is tested across four tasks: *chair_agne*, *chair_ingolf*, *table_dockstra*, and *extend kitchen*. As depicted in Fig. 6(a), SCaR achieves the highest success rates in all four tasks when $\alpha = 0.5$, indicating a balance between RL and IL at the beginning of learning. When α becomes smaller, emphasizing IL at the start, performance decreases more steeply. Conversely, as α becomes larger, giving more weight to RL, performance also declines but at a slower rate. We also evaluate the impact of different sizes of expert datasets on three skill chaining methods: Policy Sequencing, T-STAR, and SCaR, specifically in the *chair_ingolf* task. We vary the overall task expert data size from 80, 120, 200, 400, 600, to 800 demos. As shown in Fig. 6(b), the results indicate significant performance improvement when increasing the dataset size from 400 to 800 demos, while the improvement is less pronounced when going from 80 to 120 demos. This demonstrates the importance of the demo dataset size in the effectiveness of data-driven approaches like skill chaining.

6 Discussion

Limitation and future directions Limitations of our work are that the sub-task division of the long-horizon task is predefined and does not involve visual and semantic processing of objects. Scaling up our framework to address longer-horizon visual manipulation tasks is a direction we aim to investigate in future work. For instance, incorporating a more scalable architecture [48] along with large-scale pre-training on large datasets [49, 50] would be an interesting direction. Another compelling direction is applying our framework to actual robotic furniture assembly tasks, beyond just staged robotic pick-and-place tasks. Building a real-world deployment environment for furniture assembly and being able to guarantee full insertion of each furniture module are huge challenges.

Conclusion In this paper, we introduce SCaR, a novel skill chaining framework that ensures smooth and stable execution of long-horizon robotic manipulation tasks via dual regularization within and between sub-task skills. Extensive experiments demonstrate that the SCaR framework achieves better task success rates than the baseline methods in both simulated and real-robot manipulation tasks, while being robust against perturbations. We hope this work will inspire future research to further explore the potential of skill chaining for long-horizon robotic manipulation.

References

[1] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[2] Francisco Suárez-Ruiz and Quang-Cuong Pham. A framework for fine robotic assembly. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 421–426. IEEE, 2016.

[3] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *Robotics: Science and Systems XIV*, 2018.

[4] Divye Jain, Andrew Li, Shivam Singhal, Aravind Rajeswaran, Vikash Kumar, and Emanuel Todorov. Learning deep visuomotor policies for dexterous hand manipulation. In *2019 international conference on robotics and automation (ICRA)*, pages 3636–3643. IEEE, 2019.

[5] George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3):360–375, 2012.

[6] Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pages 3418–3428. PMLR, 2019.

[7] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[8] Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. What matters for on-policy deep actor-critic methods? a large-scale study. In *International conference on learning representations*, 2020.

[9] Yuyang Liu, Weijun Dong, Yingdong Hu, Chuan Wen, Zhao-Heng Yin, Chongjie Zhang, and Yang Gao. Imitation learning from observation with automatic discount scheduling. *arXiv preprint arXiv:2310.07433*, 2023.

[10] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.

[11] George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in neural information processing systems*, 22, 2009.

[12] Alexander Clegg, Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(6):1–10, 2018.

[13] Youngwoon Lee, Shao-Hua Sun, Sriram Somasundaram, Edward S Hu, and Joseph J Lim. Composing complex skills by learning transition policies. In *International Conference on Learning Representations*, 2018.

[14] Youngwoon Lee, Jingyun Yang, and Joseph J Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International conference on learning representations*, 2019.

[15] Youngwoon Lee, Joseph J Lim, Anima Anandkumar, and Yuke Zhu. Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization. In *Conference on Robot Learning (CoRL 2022)*, pages 406–416. PMLR, 2022.

[16] Jiayuan Gu, Devendra Singh Chaplot, Hao Su, and Jitendra Malik. Multi-skill mobile manipulation for object rearrangement. In *The Eleventh International Conference on Learning Representations*, 2022.

[17] Yuanpei Chen, Chen Wang, Li Fei-Fei, and Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. In *Conference on Robot Learning*, pages 3809–3829. PMLR, 2023.

[18] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[19] Jürgen Schmidhuber. *Towards compositional learning with dynamic neural networks*. Inst. für Informatik, 1990.

[20] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[21] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[22] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

[23] Visak CV Kumar, Sehoon Ha, and C Karen Liu. Expanding motor skills using relay networks. In *Conference on Robot Learning*, pages 744–756. PMLR, 2018.

[24] Yuchen Lu, Yikang Shen, Siyuan Zhou, Aaron Courville, Joshua B Tenenbaum, and Chuang Gan. Learning task decomposition with ordered memory policy network. In *International Conference on Learning Representations*, 2020.

[25] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023.

[26] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[27] Junhyuk Oh, Satinder Singh, Honglak Lee, and Pushmeet Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. In *International Conference on Machine Learning*, pages 2661–2670. PMLR, 2017.

[28] Josh Merel, Arun Ahuja, Vu Pham, Saran Tunyasuvunakool, Siqi Liu, Dhruva Tirumala, Nicolas Heess, and Greg Wayne. Hierarchical visuomotor control of humanoids. In *International Conference on Learning Representations*, 2018.

[29] Chen Wang, Danfei Xu, and Li Fei-Fei. Generalizable task planning through representation pretraining. *IEEE Robotics and Automation Letters*, 7(3):8299–8306, 2022.

[30] Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Akhil Bagaria and George Konidaris. Option discovery using deep skill chaining. In *International Conference on Learning Representations*, 2019.

[32] Zixuan Chen, Ze Ji, Shuyang Liu, Jing Huo, Yiyu Chen, and Yang Gao. Cognizing and imitating robotic skills via a dual cognition-action architecture. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2204–2206, 2024.

[33] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation*, pages 763–768. IEEE, 2009.

[34] Jens Kober, Jan Peters, Jens Kober, and Jan Peters. Movement templates for learning of hitting and batting. *Learning Motor Skills: From Algorithms to Robot Experiments*, pages 69–82, 2014.

[35] Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.

[36] Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

[37] Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pages 2905–2925. PMLR, 2023.

[38] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[39] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[40] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

[41] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[42] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. How to train your dragan. *arXiv preprint arXiv:1705.07215*, 2(4), 2017.

[43] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[44] Youngwoon Lee, Edward S Hu, and Joseph J Lim. Ikea furniture assembly environment for long-horizon complex manipulation tasks. In *2021 ieee international conference on robotics and automation (icra)*, pages 6343–6349. IEEE, 2021.

[45] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[46] Lucy Xiaoyang Shi, Joseph J Lim, and Youngwoon Lee. Skill-based model-based reinforcement learning. In *Conference on Robot Learning*, pages 2262–2272. PMLR, 2023.

[47] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[48] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021.

[49] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, 2019.

[50] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[51] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, pages 1025–1037. PMLR, 2020.

[52] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[53] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023.

[54] 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.

[55] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Technical Appendix

A Pseudo-code

Pseudo-code for adaptive sub-task skill learning and bi-directional adversarial learning are shown in Algorithm 1 and Algorithm 2 respectively. We highlight the key differences between our method and the most relevant T-STAR with a gray background.

A.1 Adaptive Sub-task Skill Learning

As shown in Algorithm 1, the innovation of the sub-task skill learning scheme we propose, compared to previous methods, consists of two parts: 1) We use a more stable weighted reward function for policy learning of sub-task skills, as shown in Eq. 1 and Eq. 3 in the main paper. 2) We introduce AES regularization constraints into this weighted reward function to periodically adaptively schedule the scale of the two reward terms, as shown in line 11-14 of Algorithm 1, allowing the robot to fully explore and learn from both the environment and the expert behaviors.

Algorithm 1 Adaptive Sub-task Skill Learning.
Key differences to T-STAR [15] in gray.

1: **Require:** expert demonstrations $\mathbb{D}_1^E, \ldots, \mathbb{D}_K^E$, sub-task MDPs $\mathcal{M}_1, \ldots, \mathcal{M}_K$
2: Initialize sub-task policies $\pi_\theta^1, \ldots, \pi_\theta^K$, least-squares GAIL discriminator $f_\phi^1, \ldots, f_\phi^K$.
3: Initialize imitation progress recognizer Φ with $\mathbb{D}^E$, balance discount factor $\lambda_{\mathrm{RL}} \leftarrow \alpha$, $\lambda_{\mathrm{IL}} \leftarrow 1-\alpha$.
4: **for** each sub-task $i = 1, \ldots, K$ **do**
5:   **for** episode = $1, 2, \ldots, N$ **do**
6:     Rollout trajectories $\tau = (s_1, a_1, r_1^{\mathrm{Env}}, \ldots, s_T)$ with π_θ^i
7:     // WEIGHTED REWARD FUNCTION
8:     Compute balanced reward $\{r_1, \ldots, r_{T-1}\} \leftarrow \lambda_{\mathrm{RL}} r^{\mathrm{Env}} + \lambda_{\mathrm{IL}} r^{\mathrm{Pred}}$
9:     Update f_ϕ^i with τ and $\tau^E \sim \mathbb{D}_i^E$ using Eq. 3
10:     Update π_θ^i with the rewarded trajectories $\{s_1, a_1, r_1, \ldots, s_T\}$
11:     // ADAPTIVE EQUILIBRIUM SCHEDULING REGULARIZATION
12:     Update imitation progress recognizer Φ with τ and $\tau^E \sim \mathbb{D}_i^E$
13:     Query Φ about the current imitation progress p
14:     Update balance discount factor $\lambda_{\mathrm{RL}}, \lambda_{\mathrm{IL}} \leftarrow \varphi_\lambda(p)$
15:   **end for**
16: **end for**

A.2 Bi-directional Adversarial Learning

As shown in Algorithm 2, the innovation of the bi-directional adversarial learning mechanism consists of two parts: 1) We propose a bi-directional regularization which is trained by two balanced bi-directional constraints to better chain sequential skills, as shown in line 16-17 of Algorithm 2. 2) We also employ the adaptive sub-skill learning scheme during the bi-directional adversarial learning process in order to ensure inter-skill alignment while enabling the sub-task skills to be adaptively adjusted to task changes during fine-tuning as well, as shown in line 10-12 of Algorithm 2.

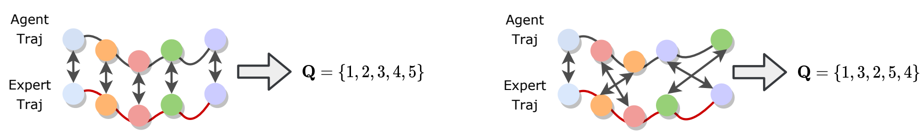
B More Details on AES Regularization

Following the mechanism described in ADS [9], our AES also employs an imitation progress recognizer Φ to monitor the extent to which the agent has assimilated the expert's behaviors. The main idea is to assess the closeness of the pair of trajectories by evaluating the agent-collected trajectory $\tau = (s_0, \ldots, s_T)$ and the expert trajectory $\tau^E = (s_0^E, \ldots, s_T^E)$ through a monotonic state-by-state alignment.

To be specific, Φ receives the agent's collected trajectories τ (line 12 in Algorithm 1) and infers the agent's current imitation progress p, $p \in [0, T)$ (line 13 in Algorithm 1). The construction

Algorithm 2 Bi-directional Adversarial Learning
Key differences to T-STAR [15] in gray.

Require: expert demonstrations $\mathbb{D}_1^E, \ldots, \mathbb{D}_K^E$, sub-task MDPs $\mathcal{M}_1, \ldots, \mathcal{M}_K$, pre-trained sub-task policies $\pi_\theta^1, \ldots, \pi_\theta^K$, pre-trained GAIL discriminator $f_\phi^1, \ldots, f_\phi^K$.
Initialize dual set discriminator $\zeta_\omega^1, \ldots, \zeta_\omega^K$, imitation identifier Φ with $\mathbb{D}^E$, balance discount factor $\lambda_{\text{RL}} \leftarrow \alpha$, $\lambda_{\text{IL}} \leftarrow 1 - \alpha$.
Initialize initial state buffers $\mathcal{B}_I^1, \ldots, \mathcal{B}_I^K$, and terminal state buffers $\mathcal{B}_\beta^1, \ldots, \mathcal{B}_\beta^K$.
for iteration $m = 0, 1, \ldots, M$ **do**
  for each sub-task $i = 1, \ldots, K$ **do**
    Sample s_0 from environment or $\mathcal{B}_\beta^{i-1}$
    Rollout trajectories $\tau = (s_1, a_1, r_1, \ldots, s_T)$ with pre-trained π_θ^i
    if τ is successful **then**
      $\mathcal{B}_I^i \leftarrow \mathcal{B}_I^i \cup s_1, \mathcal{B}_\beta^i \leftarrow \mathcal{B}_\beta^i \cup s_T$
      // ADAPTIVE EQUILIBRIUM SCHEDULING
      Update imitation identifier Φ with τ
      Query Φ about the current imitation progress p
    end if
    Update balance discount factor $\lambda_{\text{RL}}, \lambda_{\text{IL}} \leftarrow \varphi_\lambda(p)$
    Fine-tune f_ϕ^i with τ and $\tau^E \sim \mathbb{D}_i^E$
    // TRAIN BI-DIRECTIONAL DISCRIMINATOR
    Update ζ_ω^i with $s_\beta \sim \mathcal{B}_\beta^{i-1}$ and $s_I \sim \mathcal{B}_I^i$ with $\mathcal{L}_i(\omega) = \frac{1}{2}\mathcal{C}_1 + \frac{1}{2}\mathcal{C}_2$
    // FINE-TUNE WITH DUAL REGULARIZATION
    Update π_θ^i with $r_i(s_t, a_t, s_{t+1}; \phi, \omega)$ using Eq. 7
  end for
end for



(a) (b)

Figure 7: Visualization of the construction of the sequence **Q**. To be more intuitive, we directly represent the minimum cosine similarity with double arrows.

of Φ, with reference to ADS, first requires the construction of a sequence $\mathbf{Q}(q_1, \ldots, q_T)$, where $q_i = \text{argmin}_j c(s_i, s_j^E)$ is the index of the nearest neighbor of s_i in τ^E, c is the cosine similarity. As shown in Fig. 7, If τ and τ^E are exactly the same, then **Q** becomes a strictly increasing sequence (Fig 7(a)). On the contrary, if τ and τ^E characterize some different behaviors, there are some unordered sequences in **Q** (Fig 7(b)).

After constructing **Q**, the progress alignment between τ and τ^E is measured as the length of the longest increasing subsequence (LIS) in **Q**, denoted as $LIS(\tau, \tau^E)$. For instance, if $\mathbf{Q} = \{1, 3, 2, 5, 4\}$ as in Fig 7(b), then its LIS can be $\{1, 3, 5\}$, $\{1, 2, 5\}$, $\{1, 3, 4\}$ or $\{1, 2, 4\}$. The LIS measurement concentrates on the consistency of the macroscopic trends in these trajectories, thereby preventing overfitting to the microscopic features in the observation [9].

Further, if the following inequality Eq. 8 holds, this indicates that at this time step, the agent's imitation of the expert's action is equivalent to the level of the expert's performance, then the agent's imitation progress p will increase by 1:

$$\max_{\grave{\tau}^E \in \mathbb{D}^E} LIS(\tau_{1:p+1}, \grave{\tau}_{1:p+1}^E) \geq \rho \times \min_{\grave{\tau}^E, \acute{\tau}^E \in \mathbb{D}^E} LIS(\grave{\tau}_{1:p+1}^E, \acute{\tau}_{1:p+1}^E), \tag{8}$$

where $\grave{\tau}^E \neq \acute{\tau}^E$, the subscript $1 : p + 1$ denotes the first $p + 1$ steps of the extracted trajectory, and $\rho \in [0, 1]$ controls the stringency of the imitation progress monitoring.

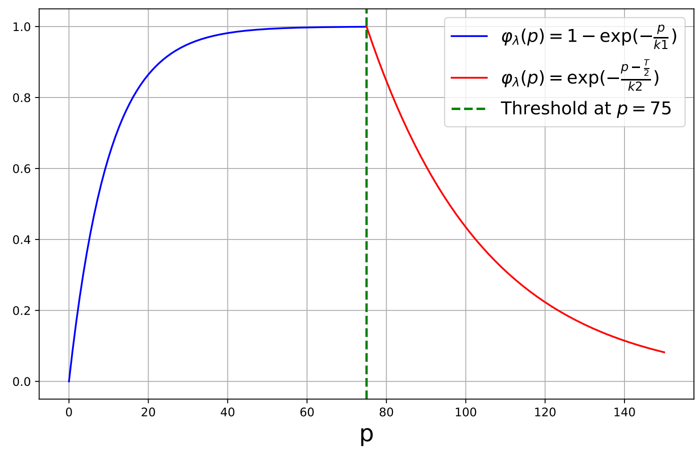


Figure 8: Visualization of the mapping function $\varphi_\lambda(p)$. In this example, we assume that $T = 150$.

After obtaining the current imitation progress p of the agent, AES then adopts a mapping function $\varphi_\lambda(p)$ to schedule the two new balance discount factors λ_{RL} and λ_{IL}. Straightforward idea of setting $\varphi_\lambda(p)$ is that **If p reaches a certain threshold, i.e., the agent is able to imitate the expert's behavior well, then the more the agent tends to imitate the expert's behavior in subsequent training, and vice versa.** Therefore, we set the threshold as $\frac{T}{2}$. If $p \in [0, \frac{T}{2})$, we propose $\varphi_\lambda(p) = 1 - e^{\left(-\frac{p}{k}\right)}$; if $p \in [\frac{T}{2}, T)$, we propose $\varphi_\lambda(p) = e^{\left(-\frac{p-\frac{T}{2}}{k}\right)}$, where k is used to flatten the curve of the mapping function. The mapping function shown in Fig. 8, where $T = 150$. In our experiments, we use different flatten factors for the two stages, where $k1 = 10$ and $k2 = 30$.

Then λ_{RL} and λ_{IL} are scheduled to be :

$$\begin{cases} \lambda_{\mathrm{RL}} = \alpha^{tanh(\frac{p}{k})}, \lambda_{\mathrm{IL}} = 1 - \alpha^{tanh(\frac{p}{k})}. & \text{if } p \in [0, \frac{T}{2}) \\ \lambda_{\mathrm{IL}} = \alpha^{tanh(\frac{k}{p})}, \lambda_{\mathrm{RL}} = 1 - \alpha^{tanh(\frac{k}{p})}. & \text{if } p \in [\frac{T}{2}, T) \end{cases} \tag{9}$$

As can be seen from Eq. 9, Fig 8 and Fig. 9, the scale of λ_{RL} is scheduled to be larger than λ_{IL} when p does not reach the imitation process threshold, but this gap gets smaller and smaller as p gets larger. When p reaches the threshold $\frac{T}{2}$, the scale of λ_{IL} is scheduled to be larger than λ_{RL}, while the scale of λ_{IL} increases as the agent imitates better.

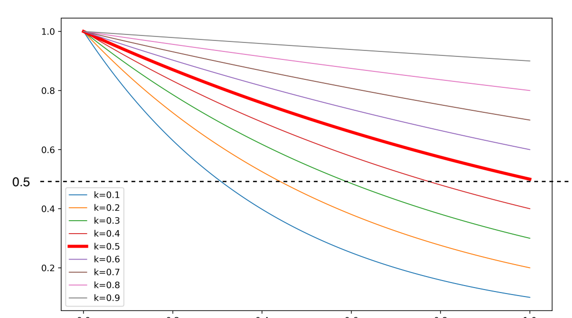


Figure 9: $\alpha^{\varphi_\lambda(p)}$ based on the variation of different α sizes in $\varphi_\lambda(p) \in [0, 1]$. We use $\alpha = 0.5$ as the base in our experiments.

Thus, if p is larger and reaches a threshold step, i.e., the agent is able to imitate the expert's behavior well, then the more the agent tends to imitate the expert's behavior in subsequent training, and vice versa. The entire process is adaptively scheduled based on Φ periodic monitoring of the agent's imitation process. Consequently, the RL and IL components of sub-task skill learning can be adaptively scheduled and regularized through AES, effectively enhancing *intra-skill dependencies* between sequential actions.

C Sub-task Skills

In our simulation experiments, we use sequences of sub-tasks defined internally by the environment [44, 45] as task decomposition sub-tasks. Here we list these sequential skills to emphasize the difficulty of long-horizon tasks. Each skill takes a 3D position as the input g_*.

IKEA Furniture Assembly:

Chair_agne (2 sub-task skills): Assemble stool leg 0 to target position $g_*^0 \rightarrow$ Assemble stool leg 1 to target position g_*^1

Chair_bernhard (2 sub-task skills): Assemble support leg 0 to target position $g_*^0 \rightarrow$ Assemble support leg 1 to target position g_*^1

Table_dockstra (3 sub-task skills): Assemble table leg 0 to target position $g_*^0 \rightarrow$ Assemble table leg 1 to target position $g_*^1 \rightarrow$ Assemble table top to target position g_*^3

Chair_ingolf (4 sub-task skills): Assemble chair support 0 to target position $g_*^0 \rightarrow$ Assemble chair support 1 to target position $g_*^1 \rightarrow$ Assemble front leg 0 to target position $g_*^3 \rightarrow$ Assemble front leg 1 to target position g_*^4

Table_lack (4 sub-task skills): Assemble table leg 0 to target position $g_*^0 \rightarrow$ Assemble table leg 1 to target position $g_*^1 \rightarrow$ Assemble table leg 2 to target position $g_*^3 \rightarrow$ Assemble table leg 3 to target position g_*^4

Toy_table (4 sub-task skills): Assemble table leg 0 insert to target position $g_*^0 \rightarrow$ Assemble table leg 1 insert to target position $g_*^1 \rightarrow$ Assemble table leg 2 insert to target position $g_*^3 \rightarrow$ Assemble table leg 3 insert to target position g_*^4

Kitchen Organization:

Kitchen (4 sub-task skills): Turn on the microwave to target position $g_*^0 \rightarrow$ Move the kettle to target position $g_*^1 \rightarrow$ Turn on the stove (rotate the stove button to target position g_*^2) $\rightarrow$ Turn on the light (rotate the light button to target position g_*^3)

Extended Kitchen (5 sub-task skills): Turn on the microwave to target position $g_*^0 \rightarrow$ Turn on the stove (rotate the stove button to target position g_*^1) $\rightarrow$ Turn on the light (rotate the light button to target position g_*^2) $\rightarrow$ Slide the cabinet to the right target position $g_*^3 \rightarrow$ Open the cabinet to target position g_*^4

D More Quantitative Results

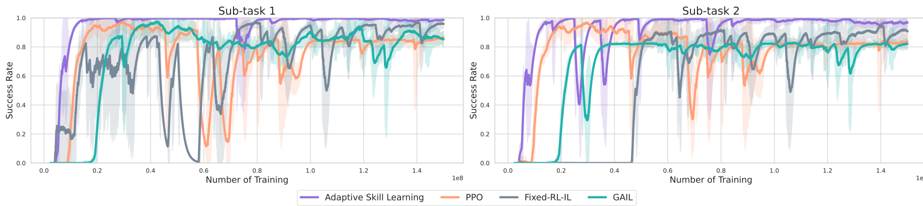
We present the training curves with different skill learning methods for sub-task skills in *chair_ingolf* task, and we further present the evaluation performance of the pre-trained skills with different methods across sub-tasks in the other 6 long-horizon simulation tasks. Also, we test the algorithms trained from scratch in the presence of perturbations to further illustrate the importance of the execution of sub-tasks on long-horizon tasks.

Additionally, the main paper does not delve into the loss function $\mathcal{L}_i(\omega)$ concerning the different scales of the bi-directional constraints in bi-directional adversarial training. Therefore, we conduct further ablation experiments to examine the impact of different scales of the two constraints in the bi-directional discriminator.

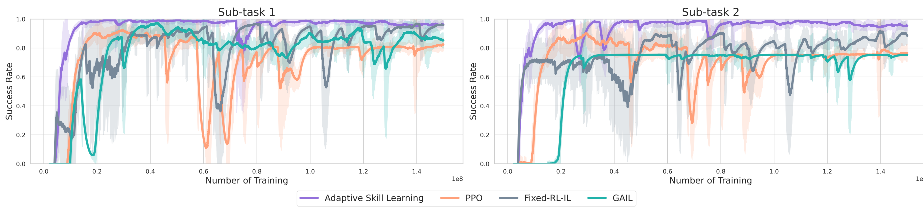
D.1 Sub-task Skill Learning Performance

D.1.1 Training performance

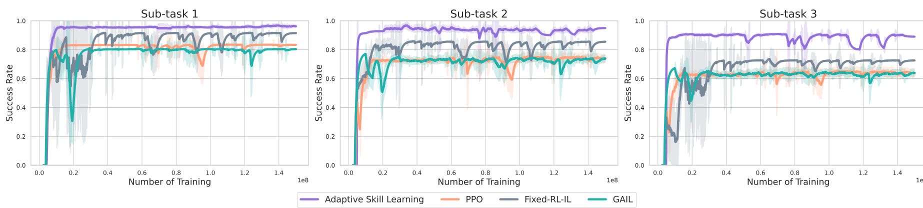
Fig. 10 shows the sub-task skill training curves in IKEA furniture assembly tasks. All methods are trained in each sub-task with 5 random seeds, 15M environment steps. As can be seen, the sub-task skill training based on PPO (learning only from environmental feedback), GAIL (learning only from expert demonstrations) and Fixed-RL-IL learning from a fixed scale of environmental feedback and expert demonstration) cannot maintain stability and exhibits significant training performance degradation as the sub-task stage increases. In contrast, the sub-task skill training process using our proposed adaptive sub-skill learning scheme has always been relatively stable and better performing.



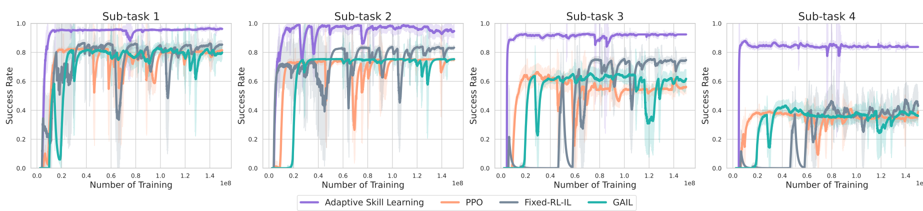
(a) chair_agne



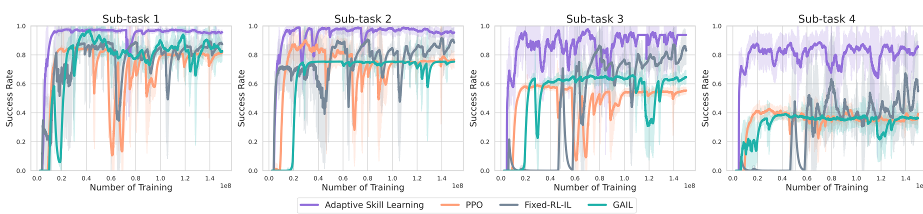
(b) chair_bernhard



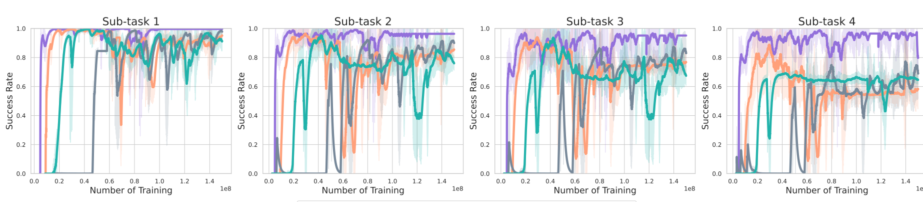
(c) table_dockstra



(d) table_lack

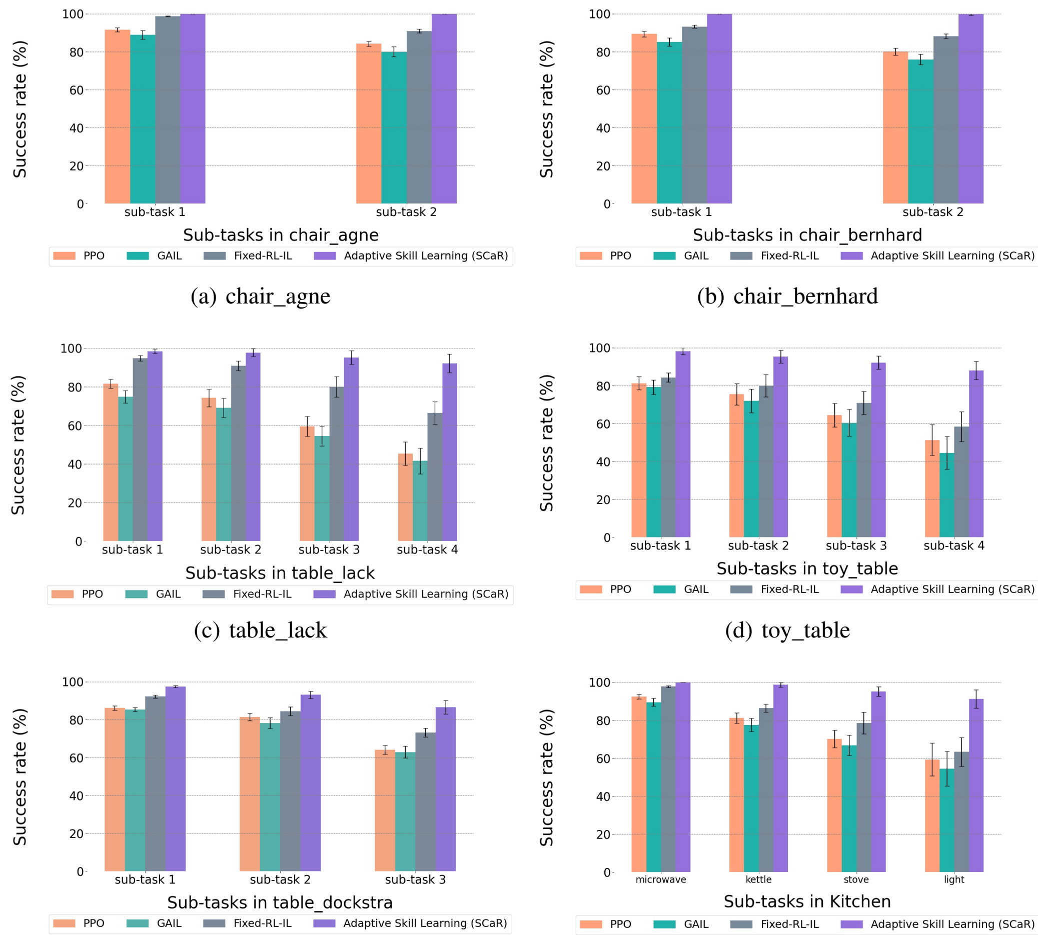


(e) toy_table



(f) chair_ingolf

Figure 10: Training curves for sub-task skills in IKEA furniture assembly tasks. The y-axis represents the success rate of the sub-task.



(a) chair_agne (b) chair_bernhard (c) table_lack (d) toy_table (e) table_dockstra (f) Kitchen

Figure 11: Evaluation Performance Comparison of Sub-task Skill Learning.

D.1.2 More evaluation performance

As shown in Fig. 11, in *chair_agne*, *chair_bernhard*, *table_lacktoy_table*, *table_dockstra*, and Kitchen tasks, even with the increase of objects in the environment - and the increase of unpredictable perturbations - our proposed adaptive skill learning learns better sub-task skills. In contrast, the PPO, GAIL, and Fixed-RL-IL baselines fail to maintain well-learning sub-task skills.

These results further corroborate that our proposed AES regularization can reinforce *inter-step dependencies* to the sequential actions within each sub-task skill, and thus pre-train better sub-task skills for long-horizon tasks.

D.2 Robustness to Perturbations

We test the algorithms trained from scratch in the presence of perturbations. As shown in Table 3, algorithms trained from scratch fail to successfully complete the task when environment perturbations occur during execution. This further illustrates the importance of dividing sub-tasks for multi-stage execution on long-horizon manipulation tasks that are contact-rich and subject to unanticipated perturbations. It also supports the significance of our work on long-horizon robotic manipulation tasks.

Table 3: Success rates of completing the two sub-tasks *chair_bernhard* and four sub-tasks *chair_ingolf* in stationary and perturbed environments.

	chair_bernhard		*chair_ingolf*	
Method	**No Perturb**	**Perturb**	**No Perturb**	**Perturb**
PPO (Scratch RL)	0.42 $\pm$ 0.12	0.01 $\pm$ 0.00	0.14 $\pm$ 0.03	0.00 $\pm$ 0.00
GAIL (Scratch IL)	0.23 $\pm$ 0.02	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
Fixed-RL-IL	0.53 $\pm$ 0.07	0.05 $\pm$ 0.00	0.22 $\pm$ 0.00	0.00 $\pm$ 0.00
SkiMo	0.62 $\pm$ 0.05	0.10 $\pm$ 0.00	0.47 $\pm$ 0.03	0.00 $\pm$ 0.00
Policy Sequencing	0.82 $\pm$ 0.09	0.51 $\pm$ 0.04	0.77 $\pm$ 0.12	0.50 $\pm$ 0.10
T-STAR	0.90 $\pm$ 0.04	0.60 $\pm$ 0.08	0.89 $\pm$ 0.04	0.59 $\pm$ 0.04
SCaR w/o Bi	0.92 $\pm$ 0.02	0.65 $\pm$ 0.11	0.91 $\pm$ 0.01	0.63 $\pm$ 0.05
SCaR w/o AES	0.94 $\pm$ 0.03	0.74 $\pm$ 0.09	0.93 $\pm$ 0.02	0.71 $\pm$ 0.07
SCaR (Ours)	**0.96** $\pm$ **0.04**	**0.85** $\pm$ **0.11**	**0.95** $\pm$ **0.03**	**0.80** $\pm$ **0.13**

D.3 Further Ablation

We set the loss function for the bi-directional discriminator in the main paper as $\mathcal{L}_i(\omega) = \frac{1}{2}\mathcal{C}_1 + \frac{1}{2}\mathcal{C}_2$., where the bi-directional constraints $\mathcal{C}_1$,$\mathcal{C}_2$ are defined as:

$$\begin{aligned}
&\textbf{next initial} \rightarrow \textbf{previous terminal:} \quad \mathcal{C}_1 = \mathbb{E}_{s_\mathcal{I} \sim \mathcal{I}_i}[\zeta_\omega^i(s_\mathcal{I}) - 1]^2 + \mathbb{E}_{s_T \sim \beta_{i-1}}[\zeta_\omega^i(s_T)]^2 \\
&\textbf{previous terminal} \rightarrow \textbf{next initial:} \quad \mathcal{C}_2 = \mathbb{E}_{s_T \sim \beta_i}[\zeta_\omega^i(s_T) - 1]^2 + \mathbb{E}_{s_\mathcal{I} \sim \mathcal{I}_{i+1}}[\zeta_\omega^i(s_\mathcal{I})]^2
\end{aligned} \tag{10}$$

The first constraint $\mathcal{C}_1$ trains the policy to have the initial states approach the terminal states of the previous policy, while the second constraint $\mathcal{C}_2$ trains the policy to have the terminal states close to the initial states of the next policy. In the experiments, these two constraints have the same scale in the training process of the bi-directional discriminator.

We wonder whether different scales of these two terms would lead to different performances, and for this reason, we conduct further parametric ablation experiments to explore this. Specifically, we define the scale parameter of the first term $\mathcal{C}_1$ as d_1, and the second term $\mathcal{C}_2$ as $d_2 = 1 - d_1$, and set **0.1, 0.3, 0.5, 0.7, 0.9** for d_1 respectively for comparison experiments. We test the effect of different scales of bi-directional adversarial training items d_1 and d_2 on the success rate of SCaR in each of the four tasks: *chair_agne*, *chair_ingolf*, *table_dockstra*, and extend kitchen. As shown in Fig. 12, the experimental result is also in line with our intuition that when the ratio of the two terms **initial** $\rightarrow$ **previous terminal** and **terminal** $\rightarrow$ **next initial** is the same, the performance is the best among the four tasks, whereas when the more imbalanced the scale of the two terms is, the worse the performance is.

This ablation result further demonstrate our statement in Sec. 4.3 in the main paper: **The purpose of the bi-directional discriminator is to establish a balanced mapping relationship between the initial states and terminal states to ensure the coherence and stability of the policy.** If the constraint in one direction (e.g., from initial states to terminal states) is stronger than the constraint in the other direction (e.g., from terminal states to initial states), the information transmission becomes asymmetric. This asymmetry results in better training in one direction and insufficient training in the other, thereby affecting overall performance.

E More Qualitative Results

Fig 13 shows the qualitative comparison of skill chaining methods. Their animated versions can be found on our project website.

F Real-Robot Long-Horizon Manipulation via Sim-to-Real Transfer

Real-robot Experiment Setup We also evaluate the skill chaining performance of real-robot for solving simple yet intuitive real-world long-horizon manipulation. We set up two types of desktop-

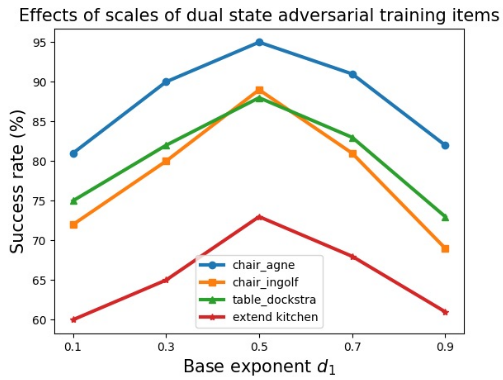
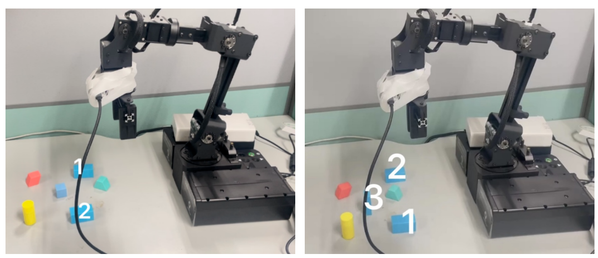


Figure 12: Impact on skill chaining performance of different scales of bi-directional constraints in SCaR.

Table 4: Skill chaining performance of real-world long-horizon robotic manipulation tasks.



Method	Success rate
T-STAR	70% (2 sub-tasks) / 50% (3 sub-tasks)
SCaR	**90%** (2 sub-tasks) / **70%** (3 sub-tasks)

level long-horizon manipulation tasks. The robotic arm needs to pick-and-place 2 and 3 blue squares in sequence, as shown in the top figures in Table 4.

We built the corresponding task environment using the gazebo simulation that accompanies the K1 robot[4], and collect 50 demonstrations of grasping skills for each square for training. With camera calibration, we deploy agents trained under simulation in a real robot desktop task to solve 2-square as well as 3-square pick-and-place tasks without the need for adaptation processes. We conduct experiments with the Sagittarius K1 and use MoveIt2 library based on ROS 2 framework for controlling the arm. We use RGB observations from RealSense D435i camera on the wrist of the robotic arm.

Results For evaluation, we measure the success rate across 10 randomized square positions for each task. As shown in Table. 4, SCaR can solve the two long-horizon tasks and outperforms T-STAR baseline. Fig. 14 and Fig. 15 show the qualitative results of successful skill chaining in the 2 and 3-blue-square pick-and-place tasks using SCaR. Video demonstrations are available at our webpage: `https://tinyurl.com/4333d6np`.

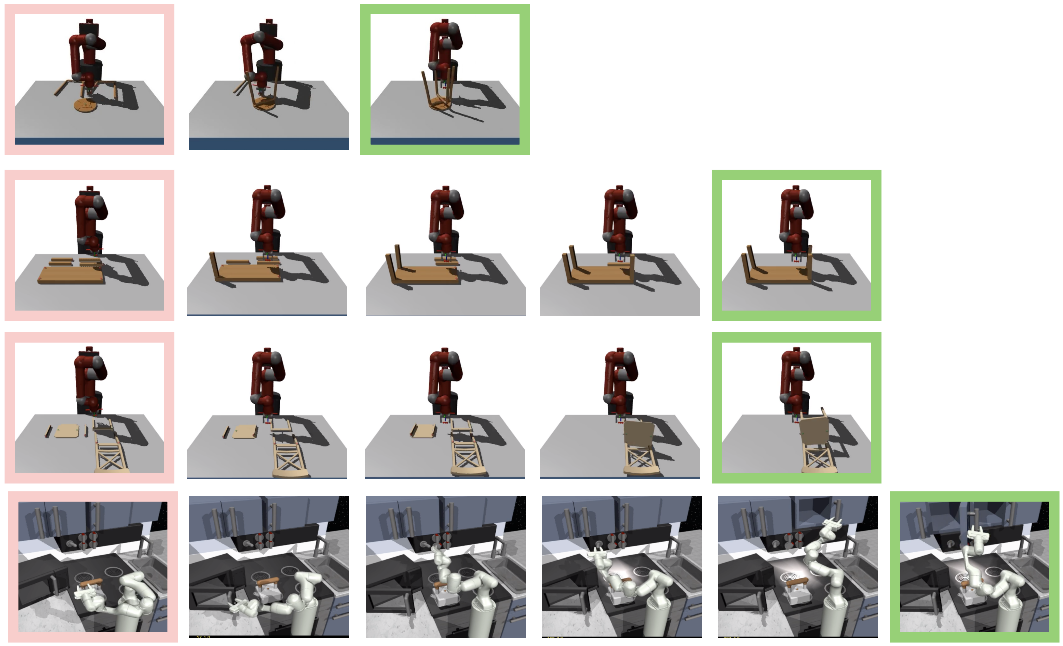
G Environment Details

G.1 IKEA Furniture Assembly

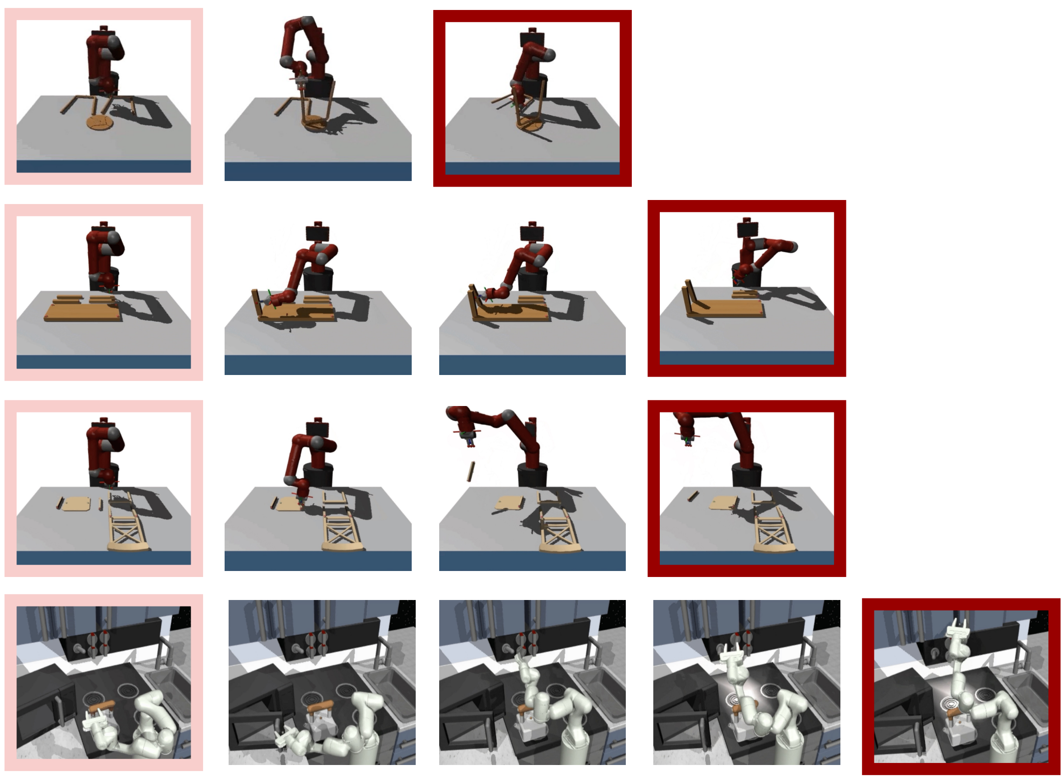
We choose six tasks, *chair_agne*, *chair_bernhard*, *chair_ingolf*, *toy_table*, *table_lack* and *table_bjorkudden* from the IKEA furniture assembly environment[5] [44] as the focal points of our

[4] `https://github.com/NXROBO/sagittarius_ws`

[5] `https://github.com/clvrai/furniture`



(a) SCaR - Successful



(b) T-STAR - Failed

Figure 13: Qualitative results of successful skill chaining performance with SCaR and failed skill chaining performance with T-STAR. More qualitative results can be found on our project website `https://tinyurl.com/4333d6np`.

experiments, as shown in Fig. 17. Our chosen robotic platform is the 7-DoF Rethink Sawyer robot, and we control it using joint velocity commands.

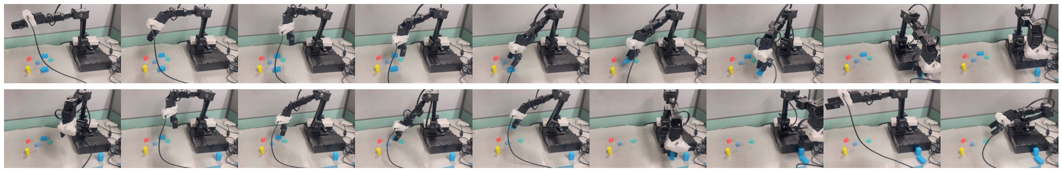


Figure 14: Visualization of the successful skill chaining in the 2-blue-square pick-and-place tasks using SCaR.

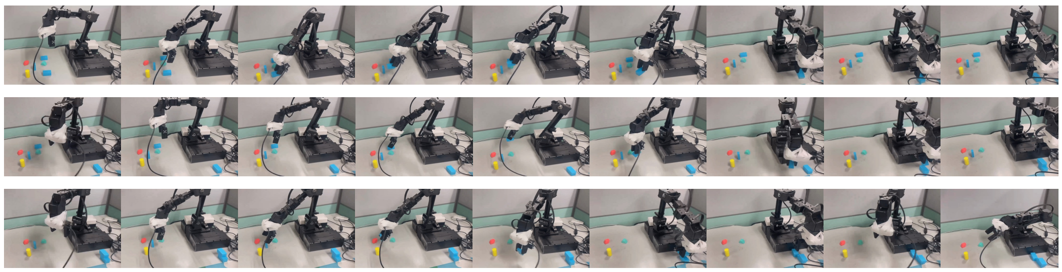


Figure 15: Visualization of the successful skill chaining in the 3-blue-square pick-and-place tasks using SCaR.

Observation Space The observation space comprises three key components: robot observations (29 dimensions), object observations (35 dimensions), and task phase information (8 dimensions). Robot observations encompass robot joint angles (7 dimensions), joint velocities (7 dimensions), gripper state (2 dimensions), gripper position (3 dimensions), gripper quaternion (4 dimensions), gripper velocity (3 dimensions), and gripper angular velocity (3 dimensions). Object observations include the positions (3 dimensions) and quaternions (4 dimensions) of all five furniture pieces in the scene. Task information, an 8-dimensional one-hot encoding, represents the current phase, including actions like reaching, grasping, lifting, moving, and aligning.

Action space The action space includes arm movement, gripper control, and the connect action, which can vary based on different control modes: 6D end-effector space control using inverse kinematics, joint velocity control, and joint torque control.

In the context of reinforcement learning (RL), we utilize a heavily shaped multi-phase dense reward obtained from the IKEA Furniture Assembly Environment [44].

Environmental Reward Function The IKEA furniture assembly environmental reward function is a multi-phase reward defined with respect to a pair of furniture parts to attach (e.g., a table leg and a table top) and the corresponding manually annotated way-points, such as a target gripping point g for each part. The reward function for a pair of furniture parts consists of eight different phases as follows:

- **Initial phase:** The robot has to reconfigure its arm pose to an appropriate pose $\mathbf{p}_{\text{init}}$ for grasping a new furniture part. The reward is proportional to the negative distance between the end-effector $\mathbf{p}_{\text{eff}}$ and $\mathbf{p}_{\text{init}}$.
- **Reach phase:** The robot reaches above a target furniture part. The reward is proportional to the negative distance between the end-effector $\mathbf{p}_{\text{eff}}$ and a point $\mathbf{p}_{\text{reach}}$ 5 cm above the gripping point g.
- **Lower phase:** The gripper is lowered onto the target part. The phase reward is proportional to the negative distance between $\mathbf{p}_{\text{eff}}$ and the target gripping points.
- **Grasp phase:** The robot learns to grasp the target part. The reward is given if the gripper contacts the part, and is proportional to the force exerted by the grippers.
- **Lift phase:** The robot lifts the gripped part up to $\mathbf{p}_{\text{lift}}$. The reward is proportional to the negative distance between the gripped part $\mathbf{p}_{\text{part}}$ and the target point $\mathbf{p}_{\text{lift}}$.

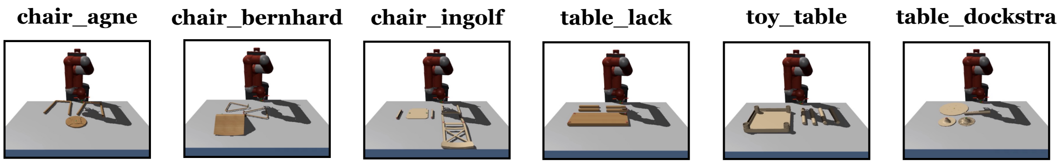


Figure 16: **IKEA Furniture Assembly Environment for Long-Horizon Complex Manipulation Tasks.**

- **Align phase:** The robot roughly rotates the gripped part before moving it. The reward is proportional to the cosine similarity between up vectors $\mathbf{u}_A$, $\mathbf{u}_B$ and forward vectors $\mathbf{f}_A$, $\mathbf{f}_B$ of the two connectors.
- **Move phase:** The robot moves and aligns the gripped part to another part. The reward is proportional to the negative distance between the connector of the gripped part and a point $\mathbf{p}_{\text{move\_to}}$ 5 cm above the connector of another part, and the cosine similarity between two connector up vectors, $\mathbf{u}_A$ and $\mathbf{u}_B$, and forward vectors $\mathbf{f}_A$ and $\mathbf{f}_B$. Note that all connectors are labeled with aligned up vectors and forward vectors.
- **Fine-grained move phase:** The robot must finely align two connectors until attached. The same reward is used as the **move phase** with a higher coefficient, making the reward more sensitive to small changes. In addition, when the part is connectable, a reward is provided based on the activation of the connect action $a[\text{connect}]$.

Upon completion of each phase, completion rewards are given to encourage the agent to move on to the next phase. In addition to stage-based rewards, control penalties, stabilizing wrist pose rewards, and grasping rewards (i.e., opening the grasping hand only during the initial, arrival, and lower stages) are provided throughout the process. If the robot releases the grasped object, the phase ends early and a negative reward is provided. Phase completion depends on the robot and part configurations satisfying distance and angle constraints with respect to the goal configuration. After all stages are completed, the stage resets to the initial stage. This process repeats until all parts are connected.

Demonstration Collection For imitation learning (IL), we gathered 200 demonstrations for each furniture part assembly using a programmatic assembly policy. Each demonstration for single-part assembly typically spans 150 steps, reflecting the overall task's inherently long-horizon nature.

Sub-tasks In our experiments, we define a sub-task as the process of assembling one part to another. Thus, the *chair_agne* and *chair_bernhard* tasks have two distinct sub-tasks, *table_dockstra* has three distinct sub-tasks, and *chair_ingolf*, *table_lack*, and *toy_table* have four distinct sub-tasks. These sub-tasks are trained independently, with their initial state sampled from the environment and random noise introduced in the [-2cm, 2cm] and [-3°, 3°] ranges of the (x, y) plane. Importantly, the decomposition of the sub-tasks is pre-determined, which means that the environment is initialized for each sub-task, and the agent receives a notification when a sub-task is successfully completed. Once the two components are firmly connected, the corresponding sub-task is considered completed and the robotic arm is guided back to its initial pose, i.e., at the center of the workspace.

Assembly Difficulty The difficulty of modeling furniture depends largely on the shape of the furniture. For example, the *toy_table* task with cylindrical legs is more difficult to grasp, whereas the *table_lack* task with rectangular legs is easier to grasp. Chairs are generally more difficult to assemble because of their irregular shape (e.g., seat and back). This is the reason why the success rates of the *toy_table* and*chair_ingolf* tasks are lower than the success rates of *table_lack*.

G.2 Kitchen Organization

We use the Franka Kitchen tasks in D4RL [45] and refer to the experimental setup in SkiMo [46] for the sub-task extensions. Including the following two tasks: **Kitchen task** and **Extended Kitchen task**, as shown in Fig. 17.

Kitchen The 7-DoF Franka Emika Panda robot arm is tasked with performing four sequential sub-tasks: *Turn on the microwave - Move the kettle - Turn on the stove - Turn on the lights*.

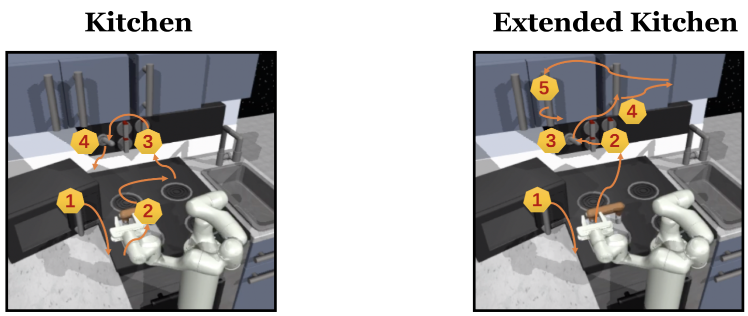


Figure 17: **Kitchen Organization Environment for Long-Horizon Complex Manipulation Tasks.**

Extended Kitchen The environment and task-agnostic data used in this experiment are consistent with those employed in the **Kitchen** scenario. However, we introduce a different set of sub-tasks for this experiment, namely: *Turn on the microwave - Turn on the stove - Turn on the lights - Slide the cabinets to the right - Open the cabinets*, as depicted in Fig. 17 (right). It's worth noting that this sequence of tasks is not aligned with the sub-task transition probabilities observed in the task-agnostic data, posing a challenge for exploration based on prior data.

Observation Space The agent operates within a 30-dimensional observation space, which includes an 11-dimensional robot proprioceptive state and 19-dimensional object states. This modified observation space removes a constant 30-dimensional goal state found in the original environment.

Action Space The agent's action space consists of 9 dimensions, encompassing 7-dimensional joint velocity control and 2-dimensional gripper velocity control.

Environmental Reward Function In terms of the environmental rewards, the agent receives a reward of +1 for each completed sub-task. The total episode length is set to 280 steps, and an episode concludes once all sub-tasks are successfully accomplished. The initial state is initialized with slight noise introduced in each state dimension.

Demonstration Collection For imitation learning, we collect 200 demonstrations per sub-task with reference to the dataset in [51] that obtained through teleoperation. This dataset covers interactions with all seven manipulatable objects within the environment.

H Network Architecture

For a fair comparison, our method and the benchmark methods use the same network structure. The policy network and the critic network consist of two layers of 128 and 256 hidden units fully connected with ReLU nonlinear properties, respectively. The output layer of the actor network outputs an action distribution, which consists of the mean and standard deviation of a Gaussian distribution. The critic network outputs only one critic value. The discriminator of GAIL [39] and the bi-directional discriminator of our proposed approach use a two-layer fully connected network with 256 hidden units. The outputs of these discriminators are clipped between [0, 1], following the least-square GAIL proposed by [40].

I Training Details

I.1 Computing Resources

Our method and all baselines were implemented using PyTorch [52]. All experiments were carried out on workstations equipped with Intel(R) Xeon(R) Gold 5218 CPUs and NVIDIA GeForce RTX 3080 2 GPUs. Pre-training of each sub-task skill policy in SCaR (150M time steps) took about 10 hours. Testing and evaluation of skill chaining for the entire long-horizon task, approximately 10 to 15 hours, depending on the difficulty of the task. Training of the skill dynamics model in SkiMo [46] took approximately 24 hours (100M steps), and PPO [47], GAIL [39], and Fix-RL-IL were slower (about 48 hours) because they all train the entire long-horizon task from scratch, with 450M time steps for the overall long-horizon task.

I.2 Algorithm Implementation Details

We report the hyperparameters used in our experiments in Table 5.

Table 5: Hyperparameters used in our experiments.

Hyperparameter	**Value**
Rollout Size	1024
Learning Rate	0.0003
Learning Rate Decay	Linear decay
Mini-batch Size	128
Discount Factor	0.99
Entropy Coefficient	0.003
Reward Scale	0.05
State Normalization	True
Discriminator learning rate	$1e^{-4}$
Sub-task training steps	150000000
# Workers	20
# Epochs per Update	10
Base exponent for balancing α	0.5
$k1$ (used to flatten the mapping function during $p \in [0, \frac{T}{2}))$	10
$k2$ (used to flatten the mapping function during $p \in [\frac{T}{2}, T))$	30
Weighting factor λ_{Bi}	10000
ρ (for imitation progress recognizer Φ)	0.9
Penalty coefficient η^{gp}	10

For the baseline implementations, we use the official code for PPO [47], GAIL [39], Fixed-RL-IL [40], SkiMo [46], Policy Sequencing [12] and T-STAR [15]. The table below (Table 6) compares key components of **SCaR** with model-based, model-free and skill-based baselines and ablated methods, where *joint training* indicates whether or not reinforcement learning combined with imitation learning is used for training.

PPO [47] Any reinforcement learning algorithm can be used for policy optimization, in this paper we choose to use Proximal Policy Optimization (PPO) and use the default hyperparameters of PPO [47].

GAIL [39] In this paper we choose to use Generative Adversarial Imitation Learning (GAIL) [39] as the learning algorithm for imitation learning and use the default hyperparameters of GAIL [39]. We specifically use an agent states s to discriminate agent and expert trajectories, instead of state-action pairs (s, a).

Fixed-RL-IL [12] We adopt the AMP [40] solution combining environmental rewards and least square GAIL with $\lambda_{\text{RL}} = \lambda_{\text{IL}} = 0.5$. For implementation details of least square GAIL training and GAIL rewards, see original paper [40].

SkiMo [46] We use the official implementation of the original paper and use the hyperparameters suggested in the official implementation.

Policy Sequencing [12] We employ the official implementation and the hyperparameters provided by [15].

T-STAR [15] We use the official implementation of the original paper and use the hyperparameters suggested in the official implementation [15].

SCaR (ours) We refer to T-STAR and use $\lambda_{\text{Re}} = 10000$ for bi-directional regularization. We take 50% of the initial state samples from the start environment of each policy, 50% of the terminal state samples at the end, and 50% of the initial state buffer and 50% of the terminal state buffer from the previous skill, respectively.

Table 6: Comparison to prior work and ablated methods.

Method	Model-based	Skill-based	Scratch training	Joint training
PPO [47] and GAIL [39]	✗	✗	✓	✗
Fixed-RL-IL [40]	✗	✗	✓	✓
SkiMo [46]	✓	✓	✓	✓
Policy Sequencing [12]	✓	✓	✗	✓
T-STAR [15]	✗	✓	✗	✓
SCaR (Ours) and SCaR w/o Bi and SCaR w/o AES	✓	✓	✗	✓

J Potential negative impacts

Since our method is currently limited to applications in simulated environments and simple desktop-level robot manipulation, it is not expected to have a significant negative impact on society. However, privacy concerns may arise if our method is applied to real-world long time-series tasks with mobility, as imitation learning agents used in applications such as autonomous driving [53] or real-time control [54, 55] require large amounts of data that often contain controversial information. In addition, the imitation learning policy is a challenge because it imitates a specified demonstration that may include bad behavior. If the expert demonstration includes some nefarious behaviors (e.g., training data for a mobile manipulation task includes behaviors that may be violent towards pedestrians), then the policy may have a significant negative impact on the user. To address this issue, future directions should focus on developing agents with safety adaptations in addition to improving performance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose a new skill chaining framework for long time-series robotic manipulation tasks that improves overall task completion performance by providing dual regularization for intra- and inter-skill dependencies. We hope this work will inspire future research to further explore the potential of skill chaining for long-horizon robotic manipulation.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the last section of the main paper: limitations mainly exist in that 1) the sub-tasks in our framework are predefined, 2) we did not test our method on a more challenging real robot furniture assembly task due to limited hardware.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide further explanation in the Appendix to explain the assumptions presented in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We further describe the network architecture, training details, dataset, and the open source codebase on which the method is based in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: All simulation environments, datasets, and open source code libraries that can reproduce our method have been described in the Appendix.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe partial details in the main paper and provide further details in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our experiments perform means and standard deviations for the five seed results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We illustrate the computational resources and the time required for the experiments in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We elaborate on these in the final section of the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper does not release data or models with high risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited or provided URLs to all the code, data, and models used in the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not do crowdsourcing experiments and research on human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.