Adversarial Skill Chaining for Long-Horizon Robot Manipulation via Terminal State Regularization

Anonymous Author(s)
Affiliation
Address
email

Abstract: Skill chaining is a promising approach for synthesizing complex behaviors by sequentially combining previously learned skills. Yet, a naive composition of skills fails when a policy encounters a starting state never seen during its training. For successful skill chaining, prior approaches attempt to widen the policy's starting state distribution to encourage successful execution. However, these approaches require larger state distributions to be covered as more polices are sequenced, and thus are limited to short skill sequences. In this paper, we propose to chain multiple policies without excessively large initial state distributions by regularizing the terminal state distributions in an adversarial learning framework. We evaluate our approach on a set of complex long-horizon manipulation tasks of furniture assembly. Our results have shown that our method establishes the first model-free reinforcement learning algorithm to solve these tasks; whereas prior skill chaining approaches fail.

Keywords: Long-Horizon Manipulation, Skill Chaining, Reinforcement Learning, Adversarial Imitation Learning

1 Introduction

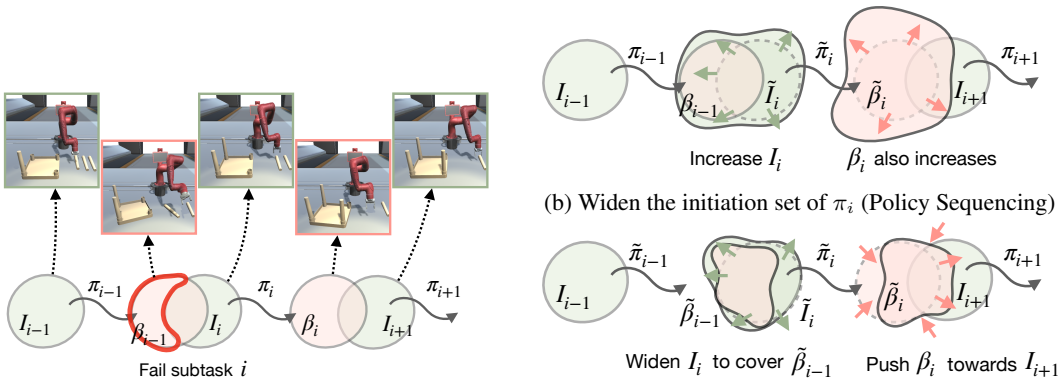
Deep reinforcement learning (RL) presents a promising framework for learning impressive robot behaviors [1–4]. Yet, learning a complex long-horizon task using a single control policy is still challenging mainly due to its high computational costs and the exploration burdens of RL models [5]. A more practical solution is to decompose a whole task into smaller chunks of subtasks, learn a policy for each subtask, and sequentially execute the subtasks to accomplish the entire task [6–9].

However, naively executing one policy after another would fail when the subtask policy encounters a starting state never seen during training [6, 7, 9]. In other words, a terminal state of one subtask may fall outside of the set of starting states that the next subtask policy can handle, and thus fail to accomplish the subtask, as illustrated in Figure 1a. Especially in robot manipulation, complex interactions between a high-DoF robot and multiple objects could lead to a wide range of robot and object configurations, which are infeasible to be covered by a single policy [10]. Therefore, skill chaining with policies with limited capability is not trivial and requires to adapting the policies to be suitable for sequential execution.

To resolve the mismatch between the terminal state distribution (*i.e.* termination set) of one policy and the initial state distribution (*i.e.* initiation set) of the next policy, prior skill chaining approaches have attempted to learn a transition policy that brings an agent to a suitable starting state [7], jointly fine-tune policies to accommodate larger initiation sets that encompass terminal states of the preceding policy [6], or utilize modulated skills for smooth transition between skills [11–14, 9]. Although these approaches can widen the initiation sets to smoothly sequence several subtask policies, it quickly becomes infeasible as the larger initiation set often leads to an even larger termination set, which is cascaded along the chain of policies, as illustrated in Figure 1b.

Instead of enlarging the initiation set to encompass a termination set modelled as a simple Gaussian distribution [6], we propose to keep the termination set small and near the initiation set of the next policy. This can prevent the termination sets from becoming too large to be covered by the subsequent

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.



(a) Naive execution of π_i

(b) Widen the initiation set of π_i (Policy Sequencing)

(c) Update π_i with terminal state regularization (Ours)

Figure 1: We aim to solve a long-horizon task, *e.g.*, furniture assembly, using independently trained subtask policies. (a) Each subtask policy, π_i, works successfully only on its initiation set (green), $\mathcal{I}_i$, and results in its termination set (pink), β_i; thus, it fails when performed outside of $\mathcal{I}_i$ (red curve). (b) To enable sequencing policies, a subsequent policy π_i needs to widen its initiation set to cover the termination set of the prior policy β_{i-1}. But, this can result in an increase of its termination set β_i, which makes fine-tuning of the following policy π_{i+1} even more challenging. This effect is exacerbated when more policies are chained together. (c) During fine-tuning of a policy π_i, we regularize the terminal state distribution β_i to be close to the initiation set of the next policy $\mathcal{I}_{i+1}$. In contrast to the boundless increase of $\tilde{\beta}_i$ in (b), our approach effectively keeps the required initiation set small over the chain of policies with the terminal state regularization.

policies, especially when executing a long sequence of skills; thus, fine-tuning of subtask polices becomes more sample efficient. As a result, small changes in the initiation sets of subsequent policies are sufficient to successfully execute a series of subtask policies.

To this end, we devise an adversarial learning framework that learns an initiation set discriminator to distinguish the initiation set of the policy that follows from terminal states; and uses it as a regularization for encouraging the terminal states of a policy to be near the initiation set of the next policy. With this terminal state regularization, the pretrained subtask policies are iteratively fine-tuned to solve the subtask with the new initial state distribution while keep the terminal states of a policy small enough to be covered by the initiation set of the subsequent policy. As a result, our model is capable of chaining a sequence of closed-loop policies to accomplish a collection of multi-stage IKEA furniture assembly tasks [15] that require high-dimensional continuous control under contact-rich dynamics.

In summary, our contributions are threefold:

- We propose a novel adversarial skill chaining framework with *Terminal STAte Regularization*, T-STAR, for learning long-horizon and hierarchical manipulation tasks.
- We demonstrate that the proposed method can learn a complex long-horizon robotic manipulation task, *furniture assembly* of two different types of furniture. Our terminal state regularization algorithms improves the success rate of the policy sequencing method from 0% to 55.5% for CHAIR_INGOLF and from 59% to 87% for TABLE_LACK. To our best knowledge, this is the first empirical result of a model-free RL method that solves these furniture assembly tasks without manual engineering.
- We present comprehensive comparisons with prior skill composition methods and qualitative visualizations to analyze our model performances.

2 Related Work

Deep reinforcement learning (RL) for continuous control [16–18] is an active research area. While some complex tasks can be solved either from a sparse reward or a heavily engineered reward, undesired behaviors often emerge [19] when tasks require several different primitive skills. Moreover,

learning such complex tasks becomes computationally impractical as the tasks become long and complicated due to the credit assignment problem and the large exploration space.

Imitation learning aims to reduce this complexity of exploration and the difficulty of learning from reward signal. Behavioral cloning approaches [20–23] greedily imitate the expert policy and therefore suffer from accumulated errors, causing a drift away from states seen in the demonstrations. On the other hand, inverse reinforcement learning [24–26] and adversarial imitation learning approaches [27, 28] encourage the agent to imitate expert trajectories with a learned reward function, which can better handle the compounding errors. Specifically, generative adversarial imitation learning (GAIL) [27] and its variants [28, 29] show improved demonstration efficiency by training a discriminator to distinguish expert versus agent transitions and using the discriminator output as a reward for policy training. Although these imitation learning methods can learn simple locomotion behaviors [27] and a handful of short-horizon manipulation tasks [30, 29], these methods easily overfit to local optima and still suffer from temporal credit assignment and accumulated errors for long-horizon tasks.

Instead of learning entire tasks using a single policy, we can tackle complex tasks by decomposing the tasks into easier and reusable subtasks. Hierarchical reinforcement learning does this by decomposing a task into a sequence of temporally extended macro actions. It often consists of one high-level policy and a set of low-level policies, such as in the options framework [31], in which the high-level policy decides which low-level policy to activate and the chosen low-level policy generates an sequence of atomic actions until the high-level policy switches it to another low-level policy. Options can be discovered without supervision [32–35], meta-learned [36], pre-defined [7, 37–39], or attained through additional supervision signals about task structure [40, 10].

To synthesize complex motor skills with a set of predefined skills, Lee et al. [7] learns to find smooth transitions between subtask policies. This method assumes the subtask policies are fixed, which makes learning such transitions feasible but, at the same time, leads to failure of smooth transition when an external state has to be changed or the end state of the prior subtask policy is too far away from the initiation set of the following subtask policy.

On the other hand, Clegg et al. [6] proposes to sequentially improve subtask policies to cover the terminal states of previous subtask policies. This method can work if the terminal state distribution is small enough to be covered by the following subtask policies. However, the terminal state distribution of the prior policy can become extremely large as there is no regularization on its terminal states. To resolve this issue, our method encourages each subtask policy to be compatible with the following subtask policies by penalizing the boundless expansion of the termination set.

Furniture assembly is a challenging robotics task requiring reliable 3D perception, high-level planning, and sophisticated control. Prior works [41–43] tackle this problem by manually programming the high-level plan and learning only a subset of low-level controls. In this paper, we propose to tackle the furniture assembly task in simulation [15] by learning all the low-level control skills and then effectively sequencing these skills.

3 Approach

In this paper, we aim to address the problem of chaining multiple policies for long-horizon complex manipulation tasks. Sequentially executing skills often fails when a policy encounters a starting state (*i.e.* a terminal state of the preceding policy) never seen during its training. The subsequent policy can learn to address these new states, but this can end up requiring even larger state distributions to be covered by the following policies. To smoothly chain multiple subtasks without requiring boundlessly large initiation sets of subtask policies, we introduce a novel adversarial skill chaining framework that constrains the terminal state distribution, as illustrated in Figure 2. Our approach first learns each subtask using subtask rewards and demonstrations (Section 3.2); it then adjusts the subtask policies to effectively chain them to complete the whole task via the terminal state regularization (Section 3.3).

3.1 Preliminaries

We formulate our learning problem as a Markov Decision Process [44] defined through a tuple $(\mathcal{S}, \mathcal{A}, R, P, \rho_0, \gamma)$ for the state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $R(s, a, s')$, transition distribution $P(s'|s, a)$, initial state distribution ρ_0, and discount factor γ. We define a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maps from states to actions and correspondingly moves an agent to a new state according

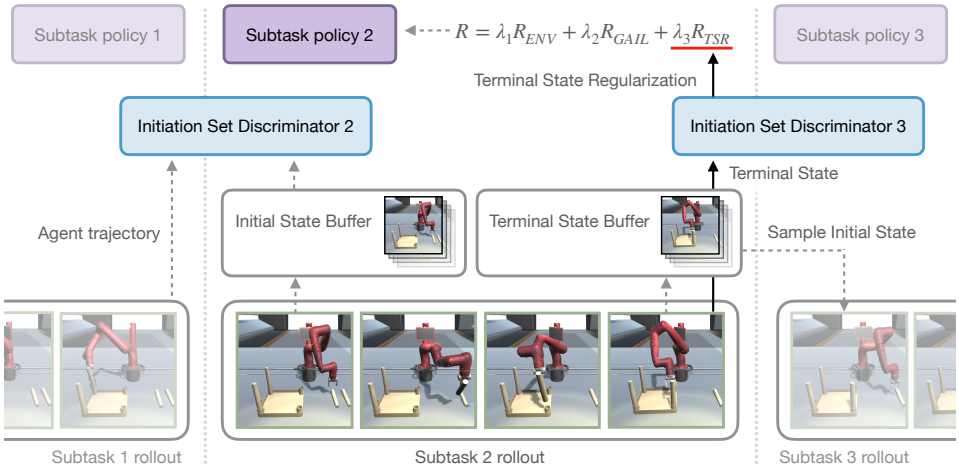


Figure 2: Our adversarial skill chaining framework regularizes the terminal state distribution to be close to the initiation set of the subsequent subtask. The *initiation set discriminator* models the initiation set distribution by discerning the initiation set and states in agent trajectories, while the policy learns to reach states close to the initiation set by augmenting the reward with the discriminator output, dubbed *terminal state regularization*. Our method jointly trains all policies and initiation set discriminators, pushing the termination set close to the initiation set of the next policy, which leads to smaller changes required for the policies that follow, especially effective in a long chain of skills.

to the transition probabilities. The policy is trained to maximize the expected sum of discounted rewards $\mathbb{E}_{(s,a)\sim\pi}\left[\sum_{t=0}^{T}\gamma^t R(s_t, a_t)\right]$ where T is the episode horizon. Each policy comes with an initiation set $\mathcal{I} \subset \mathcal{S}$ and termination set $\beta \subset \mathcal{S}$, where the initiation set $\mathcal{I}$ contains all initial states that lead to successful execution of the policy and the termination set β consists of all final states of successful executions. In addition to the reward function, we assume the learner receives a fixed set of expert demonstrations, $\mathcal{D}^e = \{\tau_1^e, \ldots, \tau_N^e\}$, where a demonstration is a sequence of state-action pairs, $\tau_j^e = (s_0, a_0, \ldots, s_{T_j-1}, a_{T_j-1}, s_{T_j})$.

3.2 Learning Subtask Policies

To solve a complicated long-horizon task (*e.g.* assembling a table), we decompose the task into smaller subtasks (*e.g.* assembling a table leg to a table top), learn a policy π_θ^i for each subtask $\mathcal{M}_i$, and chain these subtask policies. Learning each subtask solely from reward signals is still challenging in robot manipulation due to a huge state space of robot and objects to explore, and complex robot control and dynamics. Thus, for efficient exploration, we use an adversarial imitation learning approach, GAIL [27], which encourages the agent to stay near the expert trajectories using a learned reward by discerning expert and agent behaviors. Together with reinforcement learning, the policy can efficiently learn to solve the subtask even on states not covered by the demonstrations.

More specifically, we train each subtask policy π_θ^i on R^i following the reward formulation proposed in Zhu et al. [45], Peng et al. [46], which uses the weighted sum of the environment and GAIL rewards:

$$R^i(s_t, a_t, s_{t+1}; \phi) = \lambda_1 R_{ENV}(s_t, a_t, s_{t+1}) + \lambda_2 R_{GAIL}(s_t, a_t; \phi), \tag{1}$$

where $R_{GAIL}(s_t, a_t; \phi) = 1 - 0.25 \cdot [f_\phi^i(s_t, a_t) - 1]^2$ is the predicted reward by the GAIL discriminator f_ϕ^i [46], and λ_1 and λ_2 are hyperparameters that balance between the reinforcement learning and adversarial imitation learning objectives.

Due to computational limitations, training a subtask policy for all possible initial states is impractical, and hence it can cause failure on states unseen during training. Instead of indefinitely increasing the initiation set, we first train the policy on a limited set of initial states (*e.g.* predefined initial states with small noise), and later fine-tune the policy on the set of initial states required for skill chaining as described in the following section.

Algorithm 1 T-STAR: Skill chaining via terminal state regularization

Require: Expert demonstrations $\mathcal{D}^e = \{\tau_1^e, \ldots, \tau_N^e\}$, subtask MDPs $\mathcal{M}_1, \ldots, \mathcal{M}_K$

Initialize weights of subtask policies $\pi_\theta^1, \ldots, \pi_\theta^K$, GAIL discriminators $f_\phi^1, \ldots, f_\phi^K$, and initiation set discriminators $D_\omega^1, \ldots, D_\omega^K$
Initialize initial state buffers $\mathcal{B}_\mathcal{I}^1, \ldots, \mathcal{B}_\mathcal{I}^K$, and terminal state buffers $\mathcal{B}_\beta^1, \ldots, \mathcal{B}_\beta^K$
for each subtask $i = 1, ..., K$ **do**
  Rollout trajectories $\tau = (s_0, a_0, r_0, \ldots, s_T)$ with π_θ^i
  Update f_ϕ^i with τ and $\tau^e \sim \mathcal{D}^e$
  Update π_θ^i using $R^i(s_t, a_t, s_{t+1}; \phi)$ in Equation (1) ▷ Train subtask policy
end for
for $m = 0, 1, ..., M$ **do**
  for each subtask $i = 1, ..., K$ **do**
    Sample s_0 from environment or $\mathcal{B}_\beta^{i-1}$
    Rollout trajectories $\tau = (s_0, a_0, r_0, \ldots, s_T)$ with π_θ^i
    if τ is successful **then**
      $\mathcal{B}_\mathcal{I}^i \leftarrow \mathcal{B}_\mathcal{I}^i \cup s_0$ ▷ Collect initial states of successful trajectories
      $\mathcal{B}_\beta^i \leftarrow \mathcal{B}_\beta^i \cup s_T$ ▷ Collect terminal states of successful trajectories
    end if
    Update f_ϕ^i with τ and $\tau^e \sim \mathcal{D}^e$ ▷ Fine-tune GAIL discriminator
    Update π_θ^i using $R^i(s_t, a_t, s_{t+1}; \phi, \omega)$ in Equation (4) ▷ Fine-tune subtask policy with terminal state regularization
    Update D_ω^i with $s_\beta \sim \tau$ and $s_\mathcal{I} \sim \mathcal{B}_\mathcal{I}^i$ ▷ Train initiation set discriminator
  end for
end for

3.3 Policy Chaining with Terminal State Regularization

Once subtask polices are acquired, one can sequentially execute the subtask policies to complete more complex tasks. However, naively executing the polices one-by-one would fail since the policies are not trained to be smoothly connected. As can be seen in Figure 1a, independently trained subtask policies only work on a limited range of initial states. Therefore, the execution of a policy π_i fails on the terminal states of the preceding policy β_{i-1} outside of its initiation set $\mathcal{I}_i$.

For successful sequential execution of π_{i-1} and π_i, the terminal states of the preceding policy should be included in the initiation set of the subsequent policy, $\beta_{i-1} \subset \mathcal{I}_i$. This can be achieved either by widening the initiation set of the subsequent policy or by shifting the terminal state distribution of the preceding policy. Clegg et al. [6] iteratively increases the initiation sets of subtask policies, which is sufficient in tasks with a limited range of terminal states. However, in robot manipulation, the set of valid terminal states can be huge with freely located objects. For example, while successfully assembling the first table leg to the table top, a robot can mess up the workplace by moving or throwing other parts away. This issue is cascaded along the chain of policies, which leads to the boundlessly large initiation set required for policies.

Therefore, a policy needs to not only increase the initiation set (*e.g.* assemble the table leg in diverse configurations) but also regularize the termination set (*e.g.* keep the workplace organized) to be bounded and close to the initiation set of the subsequent policies. To this end, we devise an adversarial framework which jointly trains both an *initiation set discriminator*, $D_\omega^i(s_t)$, to distinguish the terminal states of the preceding policy and the initiation set of the corresponding policy, and a policy to reach the initiation set of the subsequent policy with the guidance of the initiation set discriminator. We train the initiation set discriminator for each policy to minimize the following objective:

$$\mathcal{L}_i(\omega) = \mathbb{E}_{s_\mathcal{I} \sim \mathcal{I}_i}\left[D_\omega^i(s_\mathcal{I}) - 1\right]^2 + \mathbb{E}_{s_T \sim \beta_{i-1}}\left[D_\omega^i(s_T)\right]^2. \tag{2}$$

With the initiation set discriminator, we regularize the terminal state distribution of the policy by encouraging the policy to reach a terminal state close to the initiation set of the following policy. The *terminal state regularization* can be formulated as following:

$$R_{TSR}^i(s; \omega) = \begin{cases} D_\omega^{i+1}(s) & \text{if } s \text{ is terminal and successful} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Then, we can rewrite the reward function with the terminal state regularization:

$$R^i(s_t, a_t, s_{t+1}; \phi, \omega) = \lambda_1 R_{ENV}(s_t, a_t, s_{t+1}) + \lambda_2 R_{GAIL}(s_t, a_t; \phi) + \lambda_3 R^i_{TSR}(s_{t+1}; \omega), \quad (4)$$

where λ_3 is a weighting factor for the terminal state regularization. The first two terms of this reward function guide a policy to accomplish the subtask while the terminal state regularization term forces the termination set to be closer to the initiation set of the following policy.

With this reward function incorporating the terminal state regularization, subtask policies and GAIL discriminators can be updated. Once subtask policies are updated, we collect terminal states and initiation sets with the updated policies, and train the initiation set discriminators. We alternate these procedures to smoothly chain subtask policies, as summarized in Algorithm 1. We mark changes of our algorithm with respect to Clegg et al. [6] in red in Algorithm 1.

4 Experiments

In this paper, we propose a skill chaining approach with the terminal state regularization, which encourages to match the terminal state distribution of the prior policy with suitable starting states of the following policy. Each subtask first learns its own policy and initiation set discriminator. Then, our approach iteratively fine-tunes all subtask policies and initiation set discriminators to adapt to changes in contiguous skills. Through our experiments, we aim to verify our hypothesis that the policy sequencing fails due to unbounded terminal states, which is cascaded along the sequence of skills, and demonstrate the effectiveness of our framework on learning a long sequence of complex manipulation skills.

4.1 Baselines

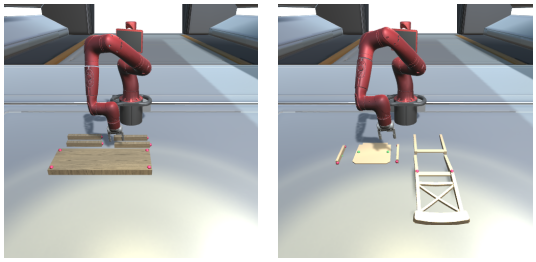
We compare our method to the state-of-the-art prior works in reinforcement learning, imitation learning, and skill composition, which are listed below:

- **BC** [20] fits a policy to the demonstration actions with supervised learning.
- **PPO** [47] is a model-free on-policy RL algorithm that learns a policy solely from the environment reward.
- **GAIL** [27] is an adversarial imitation learning approach with a discriminator trained to distinguish expert and agent state-action pairs (s, a).
- **GAIL + PPO** uses both the environment and GAIL reward as described in Equation (1) to optimize a policy.
- **SPiRL** [48] is an hierarchical RL approach that learns fixed-length skills and skill prior from the dataset, and then learns a downstream task using skill-prior-regularized RL.
- **Policy Sequencing** [6] first trains a policy for each subtask independently and finetunes the policies to cover the terminal states of the previous policy.
- **T-STAR (Ours)** learns subtask policies simultaneously, leading them to be smoothly connected using the terminal state regularization.

4.2 Tasks

We test our method and baselines with two furniture models, TABLE_LACK and CHIAR_INGOLF, from the IKEA furniture assembly environment [15] as illustrated in Figure 3:

- TABLE_LACK: Four table legs need to be picked up and aligned to the corners of the table top.
- CHAIR_INGOLF: Two chair supports and front legs need to be attached to the chair seat. Then, the chair seat needs to be attached to the chair back while avoiding collision to each other.



(a) TABLE_LACK (b) CHAIR_INGOLF

Figure 3: Two different furniture assembly tasks, TABLE_LACK and CHIAR_INGOLF.

In our experiments, we define a subtask as assembling one part to another; thus, we have *four* subtasks for each task. Subtasks are independently trained on the initial states sampled from the environment with random noise ranging from $[-2\text{cm}, 2\text{cm}]$ and $[-3°, 3°]$ in the (x, y)-plane. Moreover, this subtask decomposition is given, *i.e.*, the environment can be initialized for each subtask and the agent is informed when the subtask is completed.

For the robotic agent, we use the 7-DoF Rethink Sawyer robot operated via joint velocity control. For imitation learning, we collected 200 demonstrations for each furniture part assembly with a programmatic assembly policy. Each demonstration for single-part assembly consists of around 200-900 steps long due to the long-horizon nature of the task.

The observation space includes robot observations (29 dim), object observations (35 dim), and task phase information (8 dim). The object observations contain the positions (3 dim) and quaternions (4 dim) of all five furniture pieces in the scene. Once two parts are attached, the corresponding subtask is completed and the robot arm moves back to its initial pose in the center of the workplace.

4.3 Results

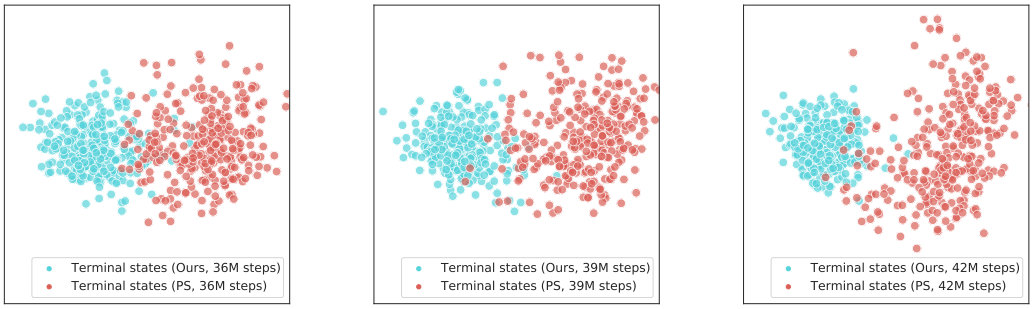
The results in Table 1 show the average progress of the furniture assembly tasks. Since each task consists of assembling furniture parts four times, completing one furniture part assembly amounts for progress of 0.25. Even though the environment has small noises in furniture initialization, both BC [20] and GAIL [27] baselines move near furniture pieces but struggle at picking up even one furniture piece. This shows the limitation of BC and GAIL in dealing with compounding errors in long-horizon tasks due to the large state space and continuous action space. Similarly, the hierarchical skill-based learning approach, SPiRL, also struggles at learning picking up a single furniture piece. This can be due to the insufficient amount of data to cover a long sequence of skills on the large state space, *e.g.*, many freely located objects. Moreover, due to the difficulty of exploration, the model-free RL baseline, PPO, rarely learns to assemble more than one part. On the other hand, the GAIL + PPO baseline can consistently learn one-part assembly, but cannot learn to assemble further parts due to the exploration challenge and temporal credit assignment problem. Note that we train baseline algorithms with over 100M environment interactions (5M for off-policy SPiRL).

Table 1: Average progress of the furniture assembly task. Each subtask amounts for 0.25 progress. Hence, 1 represents successful execution of all four subtasks while 0 means the agent does not achieve any subtask. Our method learns to complete all four subtasks in sequence and outperforms the policy sequencing baseline as well as the standard reinforcement learning and imitation learning methods.

	TABLE_LACK	CHAIR_INGOLF
BC [20]	0.0	0.0
PPO [47]	0.16	0.25
GAIL [27]	0.0	0.0
GAIL + PPO [45]	0.25	0.25
SPiRL [48]	0.05	0.02
Policy Sequencing [6]	0.82	0.70
T-STAR (Ours)	**0.95**	**0.80**

By utilizing pre-trained subtask policies with GAIL + PPO, skill chaining approaches could achieve improved performance compared to the single-policy baselines. We train the policy sequencing baseline and our method for additional 200M environment steps (50M steps for each subtask). The policy sequencing baseline [6] achieves 0.82 and 0.70 average task progress, whereas our method achieves 0.95 and 0.80 average task progress on TABLE_LACK and CHAIR_INGOLF, respectively. The performance gain of our method comes from the reduced discrepancy between the termination set and the initiation set of the next subtask thanks to the terminal state regularization.

We can observe this performance gain even clearer in the success rates. Our terminal state regularization improves the success rate of policy sequencing from 0% to 55.5% for CHAIR_INGOLF and from 59% to 87% for TABLE_LACK. We observe that with more skills to be chained, the success rate of the newly chained skill decreases, especially in CHAIR_INGOLF; the policy sequencing baseline learns to complete the first three subtasks, but fails to learn the last subtask due to excessively large and shifted initial state distribution, *i.e.*, terminal states of the preceding subtask. For more experimental results, refer to appendix, Section B.



(a) 36M steps (b) 39M steps (c) 42M steps

Figure 4: To demonstrate the benefit of our terminal state regularization, we visualize the changes in termination sets over training of the third subtask policy on CHAIR_INGOLF. We plot each terminal state by projecting its object configuration into 2D space using PCA. Through 36M to 42M training steps, both the policy sequencing baseline [6] and our method successfully learn to cover most terminal states from the second subtask. However, without regularization, the policy sequencing method (red) shows the increasing size of the termination set (*e.g.* spread over horizontally at 39M and vertically at 42M steps) as more initial states are covered by the policy. In contrast, in our approach (blue), the terminal state distribution is bounded, which shows that the terminal state regularization can effectively prevent the terminal state distribution diverging. This bounded termination set makes learning of the following skills efficient, and thus helps chaining a long sequence of skills.

4.4 Qualitative Results

To analyze the effect of the proposed terminal state regularization, we visualize the changes in termination sets over training. We first collect 300 terminal states of the third subtask policy on CHAIR_INGOLF both for our method and the policy sequencing baseline for 36M, 39M, and 42M training steps. Then, we apply PCA on the object state information in the terminal states and use the first two principal components to reduce the data dimension.

Figure 4 shows that our method effectively constrains the terminal state distribution of a subtask policy. At 36M training steps, the policy cannot solve the third subtask due to the shifted initial state distribution by the second subtask policy. With additional training, at 42M training steps, the policy learns to solve the subtask on newly added initial states for both methods. From 36M to 42M steps of training, the termination set of the policy sequencing baseline (red) spreads out horizontally after 39M training steps and vertically after 42M steps. For successful skill chaining, this wide termination set has to be covered by the following policy, requiring a large change in the policy and potentially causing an even larger termination set. In contrast, the terminal state distribution of our method (blue) does not excessively increase but actually shrinks in this experiment, which leads to successful execution and efficient adaptation of subsequent subtask policies.

5 Conclusion

We propose T-STAR, a novel adversarial skill chaining framework that addresses the problem of the increasing size of initiation sets required for executing a long chain of manipulation skills. To prevent excessively large initiation sets to be learned, we regularize the terminal state distribution of a subtask policy to be close to the initiation set of the following subtask policy. Through terminal state regularization, our approach jointly trains all subtask policies to ensure that the final state of one policy is a good initial state for the policy that follows. We demonstrate the effectiveness of our approach on a challenging long-horizon manipulation task, furniture assembly, where prior skill chaining approaches fail. These results are promising and motivate future work on chaining more skills with diverse skill combinations to tackle complex long-horizon problems. Another interesting research direction is eliminating subtask supervision required in our work and discovering subtask decomposition from large data in a unsupervised learning manner. Finally, transferring our framework to real robot systems, which involves learning a vision-based policy, improving sample efficiency, and closing simulation-to-real gaps, is our definite future work.

References

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

[2] F. Suárez-Ruiz and Q.-C. Pham. A framework for fine robotic assembly. In *IEEE International Conference on Robotics and Automation*, pages 421–426, 2016.

[3] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems*, 2018.

[4] D. Jain, A. Li, S. Singhal, A. Rajeswaran, V. Kumar, and E. Todorov. Learning deep visuomotor policies for dexterous hand manipulation. In *IEEE International Conference on Robotics and Automation*, pages 3636–3643. IEEE, 2019.

[5] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, and O. Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nIAxjsniDzg.

[6] A. Clegg, W. Yu, J. Tan, C. K. Liu, and G. Turk. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(6):1–10, 2018.

[7] Y. Lee, S.-H. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim. Composing complex skills by learning transition policies. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rygrBhC5tQ.

[8] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. In *Neural Information Processing Systems*, 2019.

[9] Y. Lee, J. Yang, and J. J. Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International Conference on Learning Representations*, 2020.

[10] D. Ghosh, A. Singh, A. Rajeswaran, V. Kumar, and S. Levine. Divide-and-conquer reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJwelMbR-.

[11] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *IEEE International Conference on Robotics and Automation*, pages 763–768, 2009.

[12] J. Kober, K. Mülling, O. Krömer, C. H. Lampert, B. Schölkopf, and J. Peters. Movement templates for learning of hitting and batting. In *IEEE International Conference on Robotics and Automation*, pages 853–858, 2010.

[13] K. Mülling, J. Kober, O. Kroemer, and J. Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.

[14] K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rk07ZXZRb.

[15] Y. Lee, E. S. Hu, and J. J. Lim. IKEA furniture assembly environment for long-horizon complex manipulation tasks. In *IEEE International Conference on Robotics and Automation*, 2021. URL https://clvrai.com/furniture.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[17] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.

[19] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Van de Wiele, V. Mnih, N. Heess, and J. T. Springenberg. Learning by playing-solving sparse reward tasks from scratch. *arXiv preprint arXiv:1802.10567*, 2018.

[20] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, pages 305–313, 1989.

[21] S. Schaal. Learning from demonstration. In *Advances in Neural Information Processing Systems*, pages 1040–1046, 1997.

[22] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *IEEE International Conference on Robotics and Automation*, pages 512–519. IEEE, 2016.

[23] D. Pathak*, P. Mahmoudieh*, M. Luo*, P. Agrawal*, D. Chen, F. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell. Zero-shot visual imitation. In *International Conference on Learning Representations*, 2018.

[24] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, volume 1, page 2, 2000.

[25] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, 2004.

[26] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Association for the Advancement of Artificial Intelligence*, volume 8, pages 1433–1438, 2008.

[27] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[28] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkHywl-A-.

[29] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

[30] K. Zolna, S. Reed, A. Novikov, S. G. Colmenarej, D. Budden, S. Cabi, M. Denil, N. de Freitas, and Z. Wang. Task-relevant adversarial imitation learning. In *Conference on Robot Learning*, 2020.

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

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

[33] P.-L. Bacon, J. Harb, and D. Precup. The option-critic architecture. In *Association for the Advancement of Artificial Intelligence*, pages 1726–1734, 2017.

[34] O. Nachum, S. S. Gu, H. Lee, and S. Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3303–3313, 2018.

[35] A. Levy, G. Konidaris, R. Platt, and K. Saenko. Learning multi-level hierarchies with hindsight. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryzECoAcY7.

[36] K. Frans, J. Ho, X. Chen, P. Abbeel, and J. Schulman. Meta learning shared hierarchies. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyX0IeWAW.

[37] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 3675–3683, 2016.

[38] J. Oh, S. Singh, H. Lee, and P. Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. In *International Conference on Learning Representations*, pages 2661–2670, 2017.

[39] J. Merel, A. Ahuja, V. Pham, S. Tunyasuvunakool, S. Liu, D. Tirumala, N. Heess, and G. Wayne. Hierarchical visuomotor control of humanoids. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJfYvo09Y7.

[40] J. Andreas, D. Klein, and S. Levine. Modular multitask reinforcement learning with policy sketches. In *International Conference on Machine Learning*, pages 166–175, 2017.

[41] S. Niekum, S. Chitta, A. G. Barto, B. Marthi, and S. Osentoski. Incremental semantically grounded learning from demonstration. In *Robotics: Science and Systems*, 2013.

[42] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *IEEE International Conference on Robotics and Automation*, pages 855–862, 2013.

[43] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham. Can robots assemble an ikea chair? *Science Robotics*, 3(17): eaat6385, 2018.

[44] R. S. Sutton. *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 1984.

[45] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, and N. Heess. Reinforcement and imitation learning for diverse visuomotor skills. In *Robotics: Science and Systems*, 2018.

[46] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *arXiv preprint arXiv:2104.02180*, 2021.

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

[48] K. Pertsch, Y. Lee, and J. J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning*, 2020.