2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

SLIM: Skill Learning with Multiple Critics

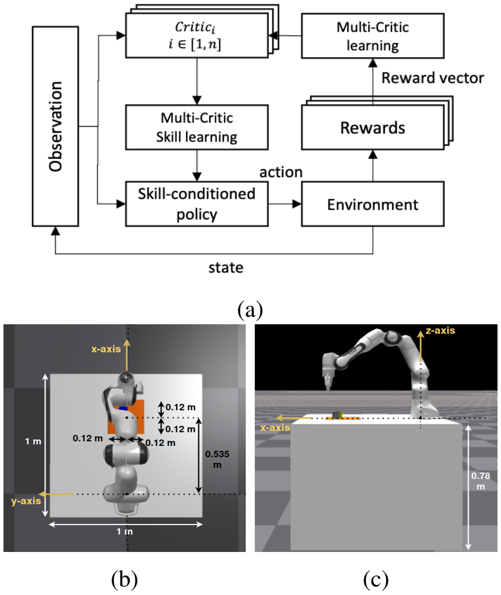
David Emukpere[1], Bingbing Wu[1], Julien Perez[2†], Jean-Michel Renders[1]

Abstract— **Self-supervised skill learning aims to acquire useful behaviors that leverage the underlying dynamics of the environment. Latent variable models, based on mutual information maximization, have been successful in this task but still struggle in the context of robotic manipulation. As it requires impacting a possibly large set of degrees of freedom composing the environment, mutual information maximization fails alone in producing useful and safe manipulation behaviors. Furthermore, tackling this by augmenting skill discovery rewards with additional rewards through a naive combination might fail to produce desired behaviors. To address this limitation, we introduce SLIM, a multi-critic learning approach for skill discovery with a particular focus on robotic manipulation. Our main insight is that utilizing multiple critics in an actor-critic framework to gracefully combine multiple reward functions leads to a significant improvement in latent-variable skill discovery for robotic manipulation while overcoming possible interference occurring among rewards which hinders convergence to useful skills. Furthermore, in the context of tabletop manipulation, we demonstrate the applicability of our novel skill discovery approach to acquire safe and efficient motor primitives in a hierarchical reinforcement learning fashion and leverage them through planning, significantly surpassing baseline approaches for skill discovery.**

I. Introduction

Self-supervised methods for skill discovery have been extensively developed in recent years as they enable robots to acquire reusable and transferable knowledge. This flexibility is crucial in dynamic and unstructured environments where robots encounter variations, uncertainties, and unforeseen events. Instead of engineering explicit rules and behaviors for each individual task, robots can learn from data and experiences, making the learning process scalable, thus improving the efficiency and versatility of robotic systems.

One popular approach to skill discovery utilizes the so-called mutual information maximization objective [1] to derive intrinsic rewards [2]. Commonly, this involves training a latent-variable conditioned policy with reinforcement learning which maximizes the mutual information between the latent variable i.e. *skill*, given as input to the agent, and the agent's state [2], [3]. While this formulation has been shown to enable an embodied agent to discover behaviors with respect to changing its own state, as in locomotion [4], it struggles in situations where we desire to discover skills that affect degrees of freedom composing the state space outside the agent's own state. For example, impacting object states, as is the case in robotic manipulation, would require extensive exploration to discover interaction skills.



(a)

(b) (c)

Fig. 1: **Skill Learning wIth Multiple critics**. Our approach enables the effective combination of multiple objectives for self-supervised skill discovery in robotic manipulation. We learn dedicated critics per intrinsic reward function which is used during policy improvement by taking a weighted combination of their normalized advantages. (a) Schematic diagram (b) Simulation top view (c) Simulation side view.

A simple way to tackle this challenging case might be to augment intrinsic rewards with additional components that encourage such interactions within the environment. For example, one can introduce a reaching bonus to reward the end-effector of the considered manipulator to get close to the objects composing the considered scene. Furthermore, ensuring the safety of skill discovery is another critical problem recently introduced in [5]. Adding this also defines an extra reward component that needs to be carefully combined with other reward terms. In this work, we show that a naive implementation of combining these multiple rewards to obtain meaningful and safe interaction skills typically doesn't work or, in the best case, would require laborious tuning to derive a weighted combination that elicits the desired behavior. To solve this, we introduce a novel multi-critic [6] approach to self-supervised skill discovery that is simple to implement and requires little to no effort to find the right combination of different rewards for safe and effective

[1] NAVER LABS Europe, 6 chemin de Maupertuis, Meylan, 38240, France. email: firstname.lastname@naverlabs.com
[2] EPITA Research Laboratory (LRE), FR-94276 Le Kremlin-Bicêtre, France. email: firstname.lastname@epita.fr
[†] Work done at Naver Labs Europe

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

robotic manipulation skill discovery.

We demonstrate the applicability of our approach for acquiring safe and effective motor primitives in a hierarchical reinforcement learning (HRL) fashion. Then, we leverage them for rearrangement and object trajectory following tasks through planning, surpassing the state-of-the-art baseline approaches for skill discovery.

In summary, our main contributions are:

- We introduce SLIM, a robust multi-critic approach to latent variable skill discovery which enables us to train skill-conditioned policies with useful, diverse, and safe behaviors.
- We perform extensive ablation tests that illustrate the benefits of our approach for skill discovery.
- We evaluate SLIM against the main state-of-the-art approaches of skill discovery in challenging robotic manipulation scenarios.
- We demonstrate the benefit of SLIM for training HRL-based motor primitives used for object-centric trajectory tracking.

II. Related Work

A. Skill Discovery

Numerous skill discovery methods [3], [4], [7], [8], [9], [10], [11], [12] and benchmarks [13] for robotics have been actively studied in recent years due to the perceived fruitfulness of unsupervised pretraining for efficient adaptation to new tasks. In general, while most of these methods have produced impressive results in various robotics domains such as locomotion, navigation and simple manipulation settings, for example with narrow initial state distributions and fixed end-effector orientation, they typically struggle in more challenging manipulation environments. One reason for this is the extended exploration due to wider initial state distributions and larger action space (position *and* orientation) needed to learn consistent interaction with objects. Without these interactions, obtaining intrinsic rewards related to diversity in object manipulation becomes infeasible. In addition, mutual information maximization done without any prior information between the proprioception and exteroception parts of the state definition leads to local minima which leads to agents mostly moving around their embodiment with little environmental impact [14], [15].

There exists a few interesting approaches to mitigating this problem with skill discovery in robotic manipulation. MUSIC [16] takes the approach of partitioning the state space into the agent's state and the surrounding state, then maximizing the mutual information between them. Furthermore [17] investigate combining MUSIC [16] with DADS [3] and multiplicative compositional policy architecture [18] to encourage acquisition of transferable manipulation skills.

While MUSIC-based methods can help agents learn how to interact with objects in their environment, exploration remains challenging. Indeed, assuming the agent doesn't interact with changing parts of its surroundings, the quantity of information to learn from is limited. More recently, controllability-aware skill discovery [15] proposes a framework improving upon distance-maximizing skill discovery [14] that encourages actively seeking "hard-to-achieve" skills, showing impressive capability to acquire useful robotic manipulation skills. In this paper, we approach this problem by augmenting latent-variable skill discovery with additional rewards that improve exploration efficiency, as well as incorporating safety constraints while focusing on the effective combination of multiple rewards with the multi-critic scheme to avoid interference [19] between various reward components.

B. Multi Critic Learning

Multi-critic actor learning [6] tackles the multi-task reinforcement learning problem by employing multiple critics for each task reward function. This approach was shown to minimize possible interference between multiple-task reward signals and allow for stable policy learning in multi-task reinforcement learning. Their approach was studied and motivated by the context of multi-style learning in games. Additionally, the usage of multiple critics has been widely studied in various reinforcement learning contexts, such as for tackling overestimation in value-based reinforcement learning [20], [21], [22], [23], or for stabilizing learning with uncertainty estimation [24], [25], [26]. We differ in our motivation for utilizing multiple critics in this paper, as we are more interested in a multi-objective reinforcement learning viewpoint, particularly in the context of robotic manipulation skill discovery. To the best of our knowledge, we are the first to propose utilizing the multi-critic architecture for skill discovery with multiple objectives or constraints in a robotic manipulation framework and demonstrate its effectiveness.

III. Approach

A. Preliminaries

Skill Discovery encompasses unsupervised approaches to reinforcement learning which enable the acquisition of diverse behaviors of a reinforcement learning agent in its environment without specific task rewards. One main approach to this problem relies on mutual information maximization between a latent variable sampled from a fixed distribution $p(z)$ which encodes *skills* and states visited by a policy conditioned on this skill [2], [4]. This is usually achieved by variational information maximization [1], by maximizing the following bound:

$$\begin{aligned} \max I(s;z) &= H(z) - H(z|s) \\ &\geq \mathbb{E}_{(s,z)}[\log q_\eta(z|s)], \end{aligned} \tag{1}$$

where $z \sim p(z)$ represents skills, s represents states from an agent's trajectory $\tau = (s_0, ..., s_T)$, and $q_\eta(z|s)$ is a discriminator network approximating the posterior distribution of skills given states.

To improve mutual information based skill discovery for learning dynamic skills, Lispchitz-constrained unsupervised skill discovery (LSD) [14] proposes the maximization of the objective:

$$J_{\text{LSD}} = \mathbb{E}_{\tau,z}\big[\phi(s_T) - \phi(s_0)\big]^{\text{T}} z$$
$$\text{s.t. } \|\phi(s_T) - \phi(s_0)\| \leq \|s_T - s_0\|. \tag{2}$$

This objective effectively encourages maximal displacement in a learned state representation space $\phi(s)$, constrained by actual state space displacement with a 1-Lipschitz constant, while ensuring diversity by aligning displacement in representation space with latent skill vectors. In [14], J_{LSD} is decomposed using a telescoping sum $\mathbb{E}_{\tau,z}\big[\sum_{t=0}^{t=T-1} \phi(s_{t+1}) - \phi(s_t)\big]^{\text{T}} z$ to derive per-transition rewards for a skill-conditioned policy $\pi(a|s,z)$, and the Lipschitz constraint was implemented using Spectral Normalization [27]. ϕ and π are jointly learned using stochastic gradient descent and reinforcement learning.

Multi-critic Actor Learning is a model-free reinforcement learning approach targeting composite reward function (*i.e.* function with multiple reward components). The approach uses multiple critics, one per reward component, in combination with a single actor in an actor-critic reinforcement learning paradigm introduced in [6]. Practically, for an actor-critic algorithm using a policy gradient optimization approach, the optimization objective is:

$$J_\pi \propto \mathbb{E}_{\tau,\pi}\big[\log \pi(a|s) \sum_i \omega_i A_i\big], \tag{3}$$

where A_i represents advantage functions for each reward component and ω_i represents weights used to combine these signals. In the seminal multi-critic approach, the authors' primary focus was on learning one critic at a time during updates. They achieved this by utilizing sparse encoding for the weights based on the task being learned. However, they also conducted preliminary experiments to showcase the effectiveness of using equal weights, demonstrating the ability to interpolate between tasks. We build on this approach in our method and adapt it to self-supervised skill discovery.

B. Method

We consider a Markov Decision Process [28] augmented with a skill latent space in the domain of robotic manipulation $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{Z} \rangle$, where $\mathcal{S}$ is the state space with state vectors $s \in \mathbb{R}^{42}$ containing robot joint positions, robot joint velocities, object pose, end-effector pose, object linear velocity, object angular velocity, end-effector linear velocity and end-effector angular velocity. We note here that we consider both cartesian positions *and* orientations in object and end-effector poses. $\mathcal{A}$ is the action space of actions $a \in \mathbb{R}^7$ split into two parts: $a_{\text{arm}} \in [-1,1]^6$ corresponding to normalised delta pose of the robot's end-effector in Cartesian space which is converted to joint torques using operational space control (OSC) [29], and $a_{\text{gripper}} \in \{0,1\}$ a Boolean action to open or close the gripper. $\mathcal{P}$ is the transition function defining our environment dynamics, $\mathcal{R}$ is the reward function, and $\mathcal{Z}$ is a continuous latent space representing skills. To enable the discovery of meaningful and safe interaction skills, we define $\mathcal{R}$ as a composite reward function consisting of the following reward components:

$$r_{\text{reach}} = \frac{1}{\left\|\text{ee\_pos}_t - \text{targ\_pos}_t\right\|_2^2 + \epsilon}, \tag{4}$$

where targ_pos is a pre-specified position of interest, for example, an object's position, ee_pos is the robot's end effector position, and ϵ is a threshold for numerical stability,

$$r_{\text{discovery}} = \big(\phi(s_{t+1}) - \phi(s_t)\big)^T z_t, \tag{5}$$

where we follow the formulation in LSD [14] that decomposes the trajectory level reward into per transition rewards using a telescoping sum,

$$r_{\text{safety}} = -\mathcal{I}(s_t), \tag{6}$$

where $\mathcal{I}$ is a safety indicator function over the states encoding predefined safety constraints which are agent and environment dependent. In the context of robotic manipulation, such constraints involve joint positions, joint velocities, self-collision avoidance, end-effector velocity, and workspace limits. In the experimental section, we detail the necessity of these three components to develop a viable skill-conditioned policy for contact-rich manipulation scenarios.

We propose to use a multi-critic actor learning architecture [6] with three critics for the above reward functions to learn a latent variable skill-conditioned policy $\pi(a|s,z)$ using PPO [30]. Fig. 1a illustrates our method and, as far as our knowledge goes, this proposition hasn't been considered in the context of skill discovery. Specifically, we propose to utilize a fully separate multi-network architecture as it was shown to perform better in [6]. One key component in our implementation is that we learn the value functions for each reward function using their respective reward scales but perform a batch normalization of the advantages computed from each critic before combining them with weights for actor learning. This scheme has the advantages of (i) fostering unperturbed critic learning per reward component and (ii) easing the burden of choosing appropriate weights to ensure contributions from each reward component are well-balanced when updating the policy. Practically, we use equal weights to combine the normalized advantages. In addition, we follow the skill composition scheme from [5] by selecting a sequence of skills to execute in each episode which encourages learning safe skill composition. The full algorithm is detailed in Algorithm 1.

IV. Experiments

In the context of robotic manipulation, we aim to answer the following questions: (Q1) Does SLIM discover *more meaningful* skills than state-of-the-art skill discovery methods? (Q2) Does SLIM enable *effective combination of multiple rewards* for skill discovery? (Q3) Do skills discovered by SLIM lead to *improved learning speed* on downstream tasks? (Q4) Can skills discovered by SLIM be sequenced to perform *complex* downstream tasks?

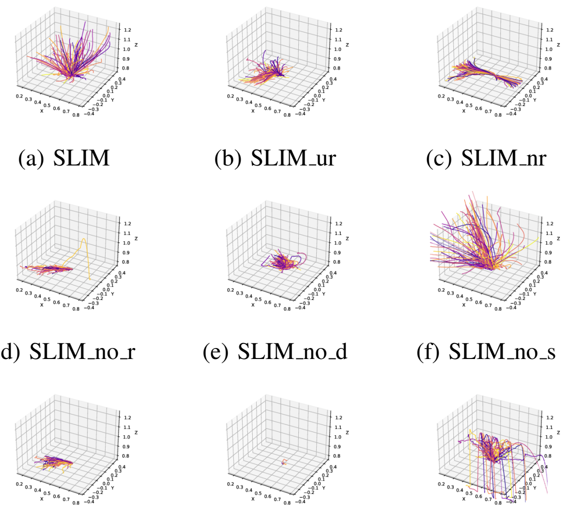
Algorithm 1 Skill Learning with Multiple Critics

```
Require: Reward functions $r_i$, Critics $V_i$, Policy $\pi_\theta$, state
    representation function $\phi$, normalization function $\nu$
repeat
    Sample sequence of skills $(z_1, ..., z_n)$ for rollouts
    Collect trajectories using $\pi_\theta$ and $(z_1, ..., z_n)$
    Update $\phi$ with rollout data to maximize Eq. (2)
    for $i \in \{\text{reach}, \text{discovery}, \text{safety}\}$ do
        Update $V_i$: $\min_{\forall (s^t, z^t) \in \tau} \|V_i(s^t, z^t) - \sum_{j=t}^{j=T} r_i^j\|$
        Compute advantage $A_i$ with GAE [31]
    end for
    for $i \in \{\text{reach}, \text{discovery}, \text{safety}\}$ do
        Batch normalize advantages $A_i$: $A_i \leftarrow \nu(A_i)$
    end for
    Update $\pi_\theta$ with PPO, using Eq. (3) with $\omega_i = 1$
until convergence
```

To answer our experimental questions, we proceed in four steps. First, we evaluate sampled rollouts of skill discovery methods to assess their respective diversity in object interaction and safety. Second, we evaluate the capabilities to train safe and efficient motor primitives with Hierarchical Reinforcement Learning (HRL) [32]. In detail, we evaluate our approach on position and orientation matching, which implicitly involves behaviors like reaching, grasping, pushing, and displacing. Third, we leverage our HRL-trained motor primitives with a planner to validate our approach for safe object-centric trajectory tracking. Finally, we extend our trajectory tracking evaluation for multiple object manipulation.

Setup We use a tabletop manipulation environment modeled in the IsaacGym simulator [33] illustrated in Fig. 1b and Fig. 1c. The environment includes a Franka Emika Panda robot, a table, and a 5-cm cube. The robot is mounted on the table and is always initialized in a fixed configuration shown in the side view image in Fig. 1c. Meanwhile, the object is initialized at a randomly sampled position within an initialization area of dimension 24 x 24 cm, illustrated with the orange square in Fig. 1b. The object's orientation is also initialized randomly using a uniform distribution over axis angle rotations. Compared to tabletop manipulation setups studied in previous works [16], [17], [14], [15], our setup is more challenging as our initial object poses are sampled from a wider distribution and our action space is larger[1]. In our experiments, we leverage the high level of parallelism enabled by IsaacGym by running 5000 instances of our environment in parallel. Furthermore, for all experiments, we use a 6-D von Mises-Fisher distribution as the fixed prior skill distribution. Intuitively, these skills correspond to representing position and orientation displacements.

Baselines We use the following skill discovery methods as baselines: DIAYN [4], LSD [14] and SASD [5]. DIAYN and LSD are chosen to serve as commonly used and cited latent variable skill discovery methods. SASD serves as a baseline that introduces the safe skill discovery formalism and tackles both objectives of skill discovery and safety.



(a) SLIM (b) SLIM_ur (c) SLIM_nr

(d) SLIM_no_r (e) SLIM_no_d (f) SLIM_no_s

(g) SASD (h) DIAYN (i) LSD

Fig. 2: **Skill trajectories for SLIM, SLIM ablations, and baselines**. SLIM outperforms baselines in terms of grasping consistency and the diversity of the cube's displacement. The baselines do not learn to pick up the cube. While SLIM ablations show different levels of object interaction with both picking and pushing behaviors emerging, only SLIM learns interactive, diverse and safe displacement manipulations

A. *Qualitative evaluation*

For our qualitative evaluation in Fig. 2 we plot color-coded 3-D object trajectories over 200 environment steps with the skill-conditioned polcies for 100 randomly sampled skill vectors z .

SLIM vs. Baselines: From Fig. 2 we observe that LSD learns to push the object but quite unsafely as the object gets knocked off the table frequently. On the other hand, SASD learns safe pushing behaviors but fails to grasp and lift, while DIAYN hardly interacts with the object. SLIM outperforms both baselines as even though they both learn some form of pushing, they all fail to learn grasping and lifting in many directions.

SLIM vs. SLIM ablations: To better understand the effectiveness of the proposed method, we compare SLIM to various ablated versions. These ablations can be grouped in two groups. The first group considers using the same reward functions as SLIM, Eq. (4), Eq. (5), Eq. (6), but combined into a single reward function (by simple summation) and hence a single critic. Note that this provides all the same reward signals used in SLIM except we perform the weighted combination of the rewards before learning a single critic. Our first ablation, henceforth called **SLIM_unnormalized_rewards**

[1] These works usually consider the Fetch robotics manipulation environments [34] where the gripper orientation is fixed and object initial orientations are also fixed to be aligned with the gripper. As such the actions only control 3-D cartesian displacements while we control 6-D position and orientation displacements.

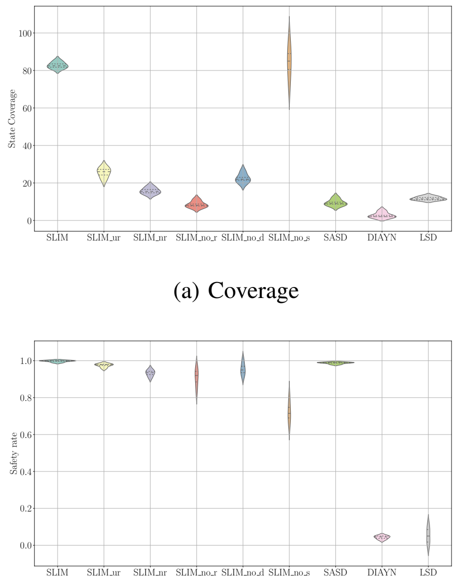
(a.k.a SLIM_ur) consists of summing up all rewards. In Fig. 2b, the skill policy rollouts with this method shows some grasping and lifting behavior is learned but the trajectories are less diverse than in SLIM which uses multiple critics. Furthermore, to prevent different reward scales to be determinants of performance differences, we define a second ablation version called **SLIM_normalized_rewards** (a.k.a SLIM_nr). Here, similar to SLIM, we apply normalization to ensure all reward signals are on similar scales before combining them into a single reward function. The trajectories from this version are visualized in Fig. 2c showing less diversity than SLIM and mostly failing to learn grasping and picking.

The second ablation group considers subset combinations of the reward functions namely: **SLIM_no_reach** (a.k.a. SLIM_no_r): using $r_{\text{discovery}}$ and r_{safety}, **SLIM_no_discovery** (a.k.a SLIM_no_d): using r_{reach} and r_{safety}, and **SLIM_no_safety** (a.k.a SLIM_no_s): using r_{reach} and $r_{\text{discovery}}$. **SLIM_no_reach** in Fig. 2d shows the effect of combining safety with LSD is very similar to SASD. We observe the safety reward constrains LSD from knocking objects off the table but with limited diversity of object displacements. On the other hand, **SLIM_no_discovery** in Fig. 2e shows safe object manipulations with some grasping, pushing and lifting, but quite limited diversity due to the missing discovery reward component. Finally, for **SLIM_no_safety** in Fig. 2f, we observe that the robot learns to displace the object in multiple directions showing a very effective combination of reaching and distance maximization discovery rewards similar to SLIM, but is unconstrained by the safety component hence it learns to over-extend the robot in order to maximize the discovery component leading to unsafe robot configurations.

Overall, our ablations show the importance of each component to obtain diverse yet interactive and safe manipulation behaviors. We observe that the three reward components are necessary and complementary to achieve our desired behaviors. In the first group of ablations, we clearly observe the difficulty with combining these three components using normalized or unnormalized sums due to possible interferences between reward signal while learning a skill-conditioned policy. Utilizing the multi-critic architecture with dedicated criticis per reward component helps to alleviate this problem and stabilize learning. Furthermore, we show with the second ablation group that while the multi-critic scheme helps with combining reward components, an omission of any of the three rewards, Eq. (4), Eq. (5), Eq. (6), hampers the overall result.

B. Quantitative evaluation

Coverage and Safety We evaluate coverage and safety for SLIM, SLIM ablations, and baselines. Coverage is measured over a 50 x 50 x 50 cm centered region discretized into 125 units of 10cm cubes. We evaluate by rolling out 100 trajectories per method repeated for 4 seeds and visualize the mean and standard deviation of the number of cubes covered by the object during the rollouts. Safety is measured as the ratio of safe states according to the indicator function in Eq. (6) encountered during the rollouts. Both measures are visualized as violin plots shown in Fig. 3. We observe that SLIM matches the strongest safety baseline SASD while being significantly superior in coverage to baselines.



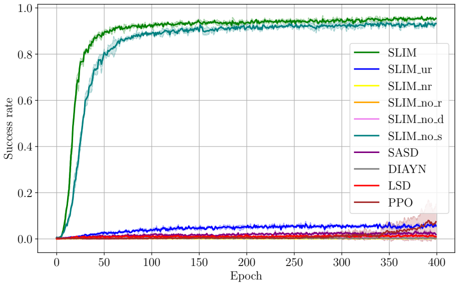
(a) Coverage

(b) Safety

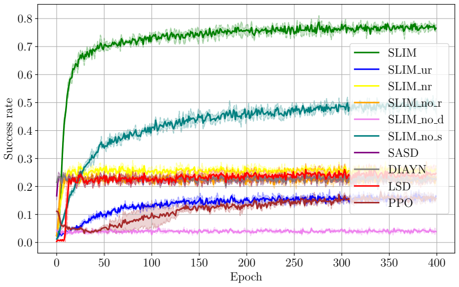
Fig. 3: **Coverage and Safety**. Coverage is the number of boxes discretizing the workspace covered by the object. Safety is the ratio of safe states encountered during random skill rollouts.

Skill utility for downstream tasks To assess the utility of discovered skills across all skill discovery methods introduced above, we train a hierarchical controller with HRL above the skill-conditioned policies to solve downstream robotic manipulation tasks. Specifically, we evaluate our approach to the tasks of position-matching and orientation-matching. We chose these two tasks because they correspond to the prime competencies required in robotic manipulation for re-arrangement type tasks. Furthermore, they illustrate how well the full range of skills learned over object position and orientation displacements can be leveraged. We compare SLIM to our baselines, SLIM ablations, and reinforcement learning from scratch with PPO. From Fig. 4 we observe that only skills learned by SLIM (and SLIM_no_safety) can be leveraged by the hierarchical controller to solve both tasks with vastly improved sample efficiency.

Safe object-trajectory following Next, we investigate the ability to use SLIM for safe object-trajectory tracking, which offers more usability than HRL alone for solving downstream tasks. We demonstrate how our skill-based HRL policy, when used as a motor primitive, can be useful. Additionally, we examine the impact of errors in this context across six different types of trajectories. As shown in Fig. 5, all trajectories are described using five ordered points defined in Cartesian space. We roll out the position-matching HRL policy trained



(a) Position matching



(b) Orientation matching

Fig. 4: **Performance on downstream tasks**. We evaluate our approach with the position-matching and orientation-matching tasks. SLIM enables improved sample efficiency across all downstream tasks

in the previous section to follow a trajectory by sequentially selecting the points in order as position-matching goals for the policy. We evaluate using the following metrics: (i) *Overall success* (%) indicates if the trajectory is followed successfully by reaching all the points above a given distance threshold of 5cm, (ii) *Maximum distance* (m) indicates the maximum distance between the object and the current waypoint at all phases in the trajectory, (iii) *Points success* indicates the total number of points successfully approached in the trajectory, and (iv) *Safety rate* (%) indicates the ratio of safe states encounter over the trajectory.

TABLE I: Safe object-trajectory following and multi-object rearrangement using SLIM-based motor primitives

Trajectory	Overall success	Max distance	Points success	Safety rate
1	100	0.04 ± 0.00	5 ± 0.00	100
2	100	0.04 ± 0.00	5 ± 0.00	100
3	80	0.05 ± 0.03	4.5 ± 1.20	99.97
4	60	0.07 ± 0.04	4.4 ± 0.91	100
5	80	0.12 ± 0.20	4.2 ± 1.66	100
6	80	0.05 ± 0.02	4.7 ± 0.64	100
Line	100	0.034 ± 0.009	3.0 ± 0.00	100
Pyramid	80	0.048 ± 0.014	2.8 ± 0.60	99.98

Our findings, as displayed in Table I, suggest that SLIM-based motor primitives can serve within planning algorithms, offering an inherent level of safety. This opens the door to executing complex trajectories over single or multiple objects while ensuring arbitrary safety criteria.

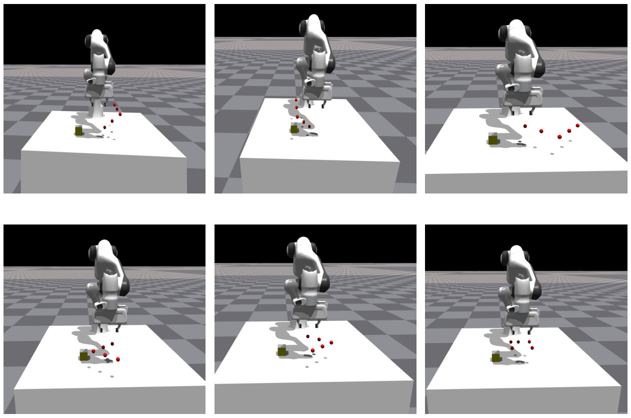


Fig. 5: **Safe trajectory following**. We evaluate our HRL policies trained over SLIM as motor primitives for safe trajectory following. The six trajectories evaluated are shown in order from top left to bottom right

Multi-object manipulation Finally, we take our trajectory-following task one step further by evaluating the ability to solve complex downstream tasks involving multiple objects. Specifically, we evaluate the same planning-based trajectory following approach but to re-arrange a set of three cubes into various configurations namely: (a) Line: where we align the cubes to the horizontal axis, and (b) Pyramid: where we form a base with two cubes and place the third cube over this base. For each cube, we plan a trajectory to reach the end pose in the desired configuration and sequentially execute the trajectory following. We evaluate using the same metrics introduced above and the results are also shown in Table I. Points success for this case refers to the number of cubes correctly placed in their final pose for the desired configuration.

V. Conclusion

In this paper, we have introduced SLIM, a novel approach to skill discovery tailored to the challenges of robotic manipulation. We empirically demonstrated that by integrating multiple critics and associated reward functions, the resulting skill-conditioned policy acquires safe and diverse manipulation skills that can be leveraged for downstream tasks using hierarchical reinforcement learning and planning. One limitation of our approach is that we assume an easy-to-design and reasonably generic bonus reward function to help with encouraging object interaction. A natural extension for future work is to replace this bonus with another intrinsic reward function that serves the same purpose of easing exploration. Likewise, we plan to further study the interference between multiple rewards, which necessitates such an approach. Additionally, exploring improved compositions of the advantages used in the policy gradient would be an interesting avenue for investigation. Lastly, sim2real deployment of our learned skill policies and assessing the benefits of applying our approach in other fields such as locomotion and navigation holds potential for fruitful exploration.

References

[1] D. Barber and F. V. Agakov, "The im algorithm: a variational approach to information maximization," in *NIPS*, 2003.

[2] K. Gregor, D. J. Rezende, and D. Wierstra, "Variational intrinsic control," *International Conference on Learning Representations*, 2016.

[3] A. Sharma, S. S. Gu, S. Levine, V. Kumar, and K. Hausman, "Dynamics-aware unsupervised discovery of skills," *ArXiv*, vol. abs/1907.01657, 2019.

[4] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *ArXiv*, vol. abs/1802.06070, 2018.

[5] S. Kim, J. Kwon, T. Lee, Y. Park, and J. Perez, "Safety-aware unsupervised skill discovery," in *2023 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[6] S. Mysore, G. Cheng, Y. Zhao, K. Saenko, and M. Wu, "Multi-critic actor learning: Teaching rl policies to act with style," in *International Conference on Learning Representations*, 2022.

[7] L. Lee, B. Eysenbach, E. Parisotto, E. P. Xing, S. Levine, and R. Salakhutdinov, "Efficient exploration via state marginal matching," *ArXiv*, 2019.

[8] V. Campos, A. R. Trott, C. Xiong, R. Socher, X. G. i Nieto, and J. Torres, "Explore, discover and learn: Unsupervised discovery of state-covering skills," in *International Conference on Machine Learning*, 2020.

[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] H. Liu and P. Abbeel, "Behavior from the void: Unsupervised active pre-training," *Neural Information Processing Systems*, 2021.

[11] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," *IEEE Robotics and Automation Letters*, vol. 7, pp. 4126–4133, 2021.

[12] M. Laskin, H. Liu, X. B. Peng, D. Yarats, A. Rajeswaran, and P. Abbeel, "Unsupervised Reinforcement Learning with Contrastive Intrinsic Control," *Neural Information Processing Systems*, 2022.

[13] M. Laskin, D. Yarats, H. Liu, K. Lee, A. Zhan, K. Lu, C. Cang, L. Pinto, and P. Abbeel, "Urlb: Unsupervised reinforcement learning benchmark," *ArXiv*, vol. abs/2110.15191, 2021.

[14] S. Park, J. Choi, J. Kim, H. Lee, and G. Kim, "Lipschitz-constrained unsupervised skill discovery," *International Conference on Learning Representations*, 2022.

[15] S. Park, K. Lee, Y. Lee, and P. Abbeel, "Controllability-aware unsupervised skill discovery," *International Conference on Machine Learning*, 2023.

[16] R. Zhao, Y. Gao, P. Abbeel, V. Tresp, and W. Xu, "Mutual information state intrinsic control," *International Conference on Learning Representations*, 2021.

[17] D. Cho, J. Kim, and H. J. Kim, "Unsupervised reinforcement learning for transferable manipulation skill discovery," *IEEE Robotics and Automation Letters*, vol. 7, pp. 7455–7462, 2022.

[18] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine, "Mcp: Learning composable hierarchical control with multiplicative compositional policies," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019.

[19] G. Shi, Q. Li, W. Zhang, J. Chen, and X.-M. Wu, "Recon: Reducing conflicting gradients from the root for multi-task learning," *ArXiv*, vol. abs/2302.11289, 2023.

[20] H. Hasselt, "Double q-learning," in *Advances in Neural Information Processing Systems*, 2010.

[21] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, 2018.

[22] Q. Lan, Y. Pan, A. Fyshe, and M. White, "Maxmin q-learning: Controlling the estimation bias of q-learning," *International Conference on Learning Representations*, 2020.

[23] X. Chen, C. Wang, Z. Zhou, and K. W. Ross, "Randomized ensembled double q-learning: Learning fast without a model," *International Conference on Learning Representations*, 2021.

[24] K. Lee, M. Laskin, A. Srinivas, and P. Abbeel, "Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning," *International Conference on Machine Learning*, 2021.

[25] Y. Wu, S. Zhai, N. Srivastava, J. M. Susskind, J. Zhang, R. Salakhutdinov, and H. Goh, "Uncertainty weighted actor-critic for offline reinforcement learning," in *International Conference on Machine Learning*, 2021.

[26] Y. Lee, A. Szot, S.-H. Sun, and J. J. Lim, "Generalizable imitation learning from observation via inferring goal proximity," in *Neural Information Processing Systems*, 2021.

[27] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *International Conference on Learning Representations*, 2018.

[28] M. L. Puterman, "Markov decision processes: Discrete stochastic dynamic programming," in *Wiley Series in Probability and Statistics*, 1994.

[29] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics Autom.*, vol. 3, pp. 43–53, 1987.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[31] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *International Conference on Learning Representations*, 2016.

[32] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dynamic Systems*, vol. 13, pp. 41–77, 2003.

[33] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," *ArXiv*, vol. abs/2108.10470, 2021.

[34] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *ArXiv*, 2018.