Intrinsic Reward Decomposition for Soft Robotic Manipulation Tasks

Vladimir Sivtsov[1], Daniil Shkolnik[1], Athanasios Papanikolaou[1], Ivan Markovic[1], Ivan Petrovic[1], Enrica Zereik[2], Fabio Bonsignorio[1]

***Abstract*— Reinforcement learning has shown remarkable potential for autonomous skill acquisition. However, effective exploration-exploitation of possible actions and states remains a fundamental challenge, especially in soft robotic manipulation tasks where the continuous deformation of soft materials causes complex nonlinear dynamics and high-dimensional state spaces. A promising solution to this problem is intrinsic motivation, which allows learning agents to explore the environment more systematically by producing self-supervised reward signals. Existing intrinsic reward techniques, however, frequently approach the whole state space as a single entity, which may reduce their efficacy in robotic manipulation, where interactions can take place between the manipulated item and the robot. In this work, we introduce a new method of intrinsic reward decomposition which focuses exploration on task-relevant interactions. Our method implements a weighted combination of random network distillation rewards derived separately from robot observations and manipulation object states. Experimental results across various manipulation tasks in the soft robotic benchmark show that this attention-inspired decomposition enables more effective discovery of manipulation strategies and significantly enhances performance.**

***Index Terms* — intrinsic motivation, exploration, reinforcement learning, embodied learning, robotics**

I. Introduction

Skill acquisition and learning-based control are crucial in the transformation from model-based control of robots acting in known environments to generalistic embodied agents in the open-ended world with its complex dynamics and uncertainties. Reinforcement learning (RL) has demonstrated great potential in this transformation, making it possible for learning agents to acquire meaningful behaviors and control strategies through trial-and-error interactions with the environment. However, it remains difficult for RL methods to effectively obtain close to optimal action policies for robotic manipulators. In particular, learning agents in complex robotic environments often suffer from inefficient exploration-exploitation, slow convergence, and as a consequence suboptimal performance in manipulation tasks [1]. At the same time, there has been an increasing interest of the research community in soft robots as a result of the desire to fully utilize the flexible materials of the robot's body in order to improve many tasks including manipulation, achieve high compliance with obstacles and intrinsic safety in real-world scenarios, in particular when cooperation with humans is needed. Soft robotic systems are different from traditional stiff robots and pose an extensive number of additional control challenges. For instance, due to their compliance and deformability, it is more difficult to model or predict the dynamic interactions with the environment, and the high dimensionality of their state and action spaces increases even more this complexity [2]. This reduces the efficiency of RL algorithms, making them less successful in obtaining good control policies.

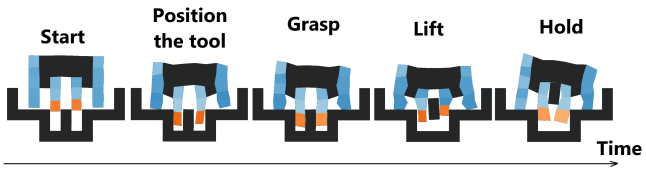


Fig. 1. Showcasing performance on a hard manipulation task that state-of-the-art RL algorithms struggle to solve

Recent advances have introduced various approaches to enhance the adaptability and efficiency of these algorithms. Some strategies focused on the inclusion of more data and the consequent feature-fusion issues, making multi-modal representations guide the learning process [3]. Several approaches introduced new loss function regularization inspired by partly known mathematical models of the system, physics (physics-informed) or information theory terms that encourage required types of behavior [4]. A promising research direction is finding new ways of modifying reward signals that would better guide learning agents toward the desired performance. Reward engineering is a conventional approach; however, it requires extensive engineering and often leads to a lot of trial and error while finding an appropriate reward function. Furthermore, methods based on optimization of these hand-crafted reward functions often find counter-intuitive ways of maximizing the cumulative rewards, which can go against the desired performance (e.g. robots intended to collect as much rubbish as possible in a strive to maximize rewards learn how to rubbish first and then collect their own rubbish i.e. working in a meaningless loop, but maximizing the rewards) [5].

Inspired by how humans can be driven by curiosity, i.e., not only relying on encouragement from the environment, intrinsic reward methods were proposed [6]. They are added to the extrinsic rewards from the environment and encourage learning agents to explore it in different ways, helping to avoid local optimum in exploration/exploitation. A general

This work was supported by the H2020 project AIFORS under Grant Agreement No 952275.

[1]University of Zagreb Faculty of Electrical Engineering and Computing, Unska ulica 3, Zagreb, 10000, Croatia, email: {name.lastname}@fer.unizg.hr

[2]Italian National Research Council, Institute of Marine Engineering (CNR-INM), 16149, Genoa, Italy

The code is available at the paper's website.

class of intrinsic reward functions, using characteristics of the type of objectives that the robotic system should achieve, is proposed in [7]. A method for adaptive balancing intrinsic and extrinsic rewards is developed in [8]. Subsequently, work in [9] shows a comprehensive comparison of intrinsically motivated learning autonomous machines based on goal-conditioned RL.

In this work, we address the challenge of the exploration-exploitation tradeoff in reinforcement learning for soft robotic manipulation and propose a novel intrinsic reward decomposition method to focus exploration on task-relevant interactions. We implement an attention-like weighted combination of random network distillation rewards derived separately from robot observations and manipulation object states. The proposed method was evaluated in various soft robotic manipulation tasks (showcased in Fig. 1) against the state-of-the-art exploration algorithms, showing that our method enables an effective discovery of manipulation strategies and significantly enhances performance.

The main contributions of this work are:

- Formulation of a novel intrinsic reward decomposition method that differentiates between robot and manipulation object states.
- Validation of the method on soft robotic manipulation benchmark and highlighting a significant performance improvement in comparison to state-of-the-art intrinsic rewards approaches.
- Investigation through comprehensive ablation study examining how different weighting schemes affect learning efficiency and task performance.

II. Related Work

Learning-based soft robot control is often formulated via RL as a Markov Decision Process (MDP), defined by the tuple (S, A, P, R, γ), where S is a state space, A is an action space, P is the transition probability function, R is a reward function and $\gamma \in [0, 1]$ is a discount factor. At each time step t, the agent interacts with the environment by observing state $s_t \in S$, selecting action $a_t \in A$, transitioning to state s_{t+1} and receiving reward r_t. The goal is to learn a policy π that maximizes the discounted expected reward $\mathrm{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t \right]$. It has been repeatedly shown that defining an appropriate reward function R, especially for soft robots, can be challenging [10], [11]. While standard RL approaches demonstrated promising results in terms of performance in many robotic tasks, their effectiveness is largely dependent on reward function definition and efficient exploration technique.

In order to balance exploration and exploitation, RL has made extensive use of different techniques. A basic approach is random exploration, where an agent incorporates additive randomness in actions. Some more advanced approaches make modifications to the loss function to encourage exploration; for example, entropy-based methods increase action diversity by adding an entropy-maximization term into the loss function [12]. The effect of adjustment of modification of the quadratic Q-learning loss function is studied in [13]. While these methods work well in many settings, they require altering the loss function, which can introduce instabilities in the learning process and can be computationally expensive.

A different approach leverages intrinsic motivation to guide exploration. Intrinsic and curiosity-driven methods change the reward function by adding terms that encourage exploration. The generation of self-supervised reward signals based on curiosity, novelty, etc., encourages learning robotic agents to explore in a more systematic way. Some count-based methods utilize the frequency of attending different states to derive an exploration bonus. For instance, exploration via elliptical episodic bonuses (E3B) [14] uses elliptical bonuses with continuous state representations and defines an intrinsic reward based on the position of the current state's embedding relative to an ellipse fitted on embeddings of previous states within the episode. Random network distillation (RND) [15] uses a predictor network that is trained on the objective of mimicking a fixed randomly initialized network with the prediction error serving as intrinsic reward for novel states. Never Give Up (NGU) and PseudoCounts [16] are based on episodic novelty with k-nearest neighbours that reward the learning agent for revisiting all states in its environment. Generalized state-dependent exploration (gSDE) [17] directly makes the exploration noise within one episode a function of the state so that the exploration is more consistent and smoother. A system leveraging autoencoders in the prediction of intrinsic rewards for enhanced navigation of robots through its state-space representation toward high-rewards areas is considered in [18].

Ensemble-based approaches employ multiple models or learners. Specifically, in [19] Q-decomposition was proposed consisting of building complex agents from simpler ones that have their own reward function with RL algorithm. Disagreement [20] leverages an ensemble of models trying to predict the next state and rewards the agent based on the variance in their predictions, driving the agent to explore areas where the models disagree, indicating uncertainty. A hybrid architecture where a value function is fitted for each part of the decomposed reward function was presented in [21]. Juozapaitis et al. [22] used reward decomposition to help explain the action preferences of learning agents via minimal sufficient explanations. The reasoning abilities of foundation large language models can also be employed to greatly help in increasing the probability of the learning agents visiting promising states [23], [24].

III. Proposed Method

In this section, we present our algorithm for intrinsic reward decomposition in RL for soft robotic manipulation. The proposed approach considers a class of manipulation problems where the observation consists of states of the robot and states corresponding to the manipulation object (e.g. angles, velocities, etc.). This observation vector can be measured directly from inherent sensors via proprioception or indirectly via computer-vision frameworks or other additional perception systems.

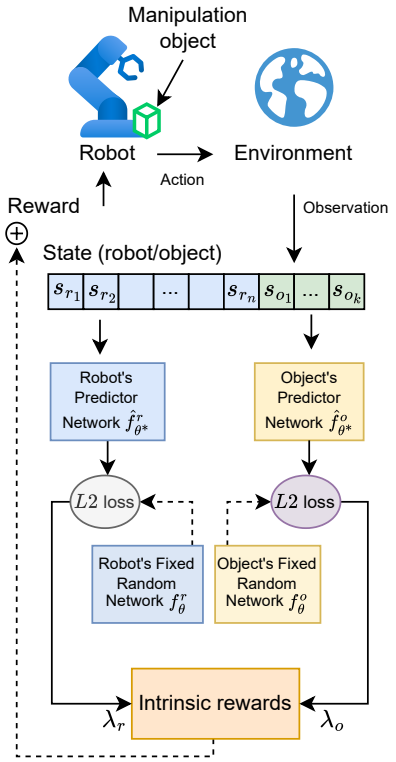


Fig. 2. Proposed intrinsic reward decomposition method. Intrinsic rewards are computed separately for states of the robot and states of the object of manipulation. The weighted sum is added to rewards from the environment during training

Typically, intrinsic reward methods consider the whole observation vector in the calculation of the intrinsic rewards. However, it limits the ability of these methods to differentiate between the states of the robot and the states of the manipulation object. The overall representation of the environment in one state and the calculation of intrinsic rewards based on this inherently averages the modes of the self-exploration from the point of view of the robot and exploration of manipulation in terms of variations of movement after the contact. At the beginning of the learning process, depending on the number of states representing the robot and manipulation object, the exploration in the smaller action space makes the reward space more uniform and not leading to the improvement.

Moreover, in the general case, the manipulation object does not necessarily have to be static at the beginning. Many real-world manipulation problems deal with dynamic objects that can move in the environment due to the existence of some external forces or environmental conditions. In this case, the agent in conventional exploration approaches would constantly encounter new observations and this would make it receive a high intrinsic motivation reward, which it should not, as the exploration does not happen because of purposeful actions of the robot. In this work, both cases of static and dynamic objects are considered as discussed in Section IV.

To address the limitations of existing curiosity and intrinsic motivation strategies, we propose a novel approach for enhancing soft robotic manipulation performance. The summary of the proposed method is shown in Fig. 2. Our method decomposes the state space into two separate components: the robot state $s_r \in S_r$ and the object state $s_o \in S_o$. The variables included in the state depend on the environment and typically include positions/velocities of point masses and orientation, as described in Evogym documentation/supplemental material [25]. The intuition behind this decomposition is to improve performance in manipulation tasks by explicitly separating the s_r and s_o and performing exploration in their corresponding distinct state spaces. After this decomposition, we leverage random network distillation to generate intrinsic rewards separately for each component.

Random network distillation consists of two networks: a fixed randomly initialized target network $f_{\theta^*} : \mathbb{R}^m \supset S_r \to \mathbb{R}^n$ and a predictor network $\hat{f}_\theta : \mathbb{R}^m \supset S_r \to \mathbb{R}^n$. The fixed random network is not updated during the training to provide a stable target that is learned by the predictor network. When a learning agent encounters an unseen observation, the prediction network is likely to have high error in the prediction of the output of the fixed network. This error acts as an intrinsic reward, encouraging enhanced exploration. To ensure that the agent does not start with any prior biases, a random orthogonal initialization is used, where weight matrices are orthogonal to each other. Given a state $s_r \in S_r$ the target network generates a fixed feature representation $v = f_{\theta*}(s_r) \in \mathbb{R}^n$. The intrinsic reward i_r is given by the prediction error:

$$i_r := \|f_\theta(s_r) - \hat{f}_{\theta^*}(s_r)\|^2.$$

The predictor network learns to predict the output of the target network

$$\theta = \underset{\theta}{\operatorname{argmin}} \|f_\theta(s_r) - \hat{f}_{\theta^*}(s_r)\|^2 \tag{1}$$

via Adam [26] optimization algorithm. The same applies to the object state component s_o to get the intrinsic reward i_o. If the agent encounters a new or rarely visited state, it is expected that its predictor network will have a high error. As the agent revisits states, the predictor improves and the intrinsic reward decreases. In order to ensure that the intrinsic reward and observations are on a consistent scale, normalization is used. More precisely, the intrinsic reward is divided by the running estimate of the standard deviation of the intrinsic rewards $\hat{i} = \frac{i}{\hat{\sigma}}$. To normalize the observation, the running mean is subtracted and the result is divided by the running standard deviation component-wise:

$$\hat{s}_j = \frac{s_j - \hat{\mu}_j}{\hat{\sigma}_j}. \tag{2}$$

The normalized observation is then clipped to the interval $[-a, a]$: $\hat{s} := \text{clip}(s, -a, a)$. Finally, we combine the intrinsic reward i_r with the intrinsic reward i_o using a weighted sum to create a composite intrinsic reward signal:

$$i_t = \lambda_r \cdot i_r + \lambda_o \cdot i_o, \tag{3}$$

for some weights $\lambda_r, \lambda_o \in \mathbb{R}_{\geq 0}$. The total reward at time t is the sum of composite intrinsic reward i_t and extrinsic

TABLE I

EVALUATION OF THE LEARNED ACTION POLICIES WITH DIFFERENT INTRINSIC REWARD STRATEGIES AT LEARNING OF SOFT ROBOTIC MANIPULATION (MEAN VALUE AND STANDARD DEVIATION, THE BEST VALUES BY MEAN REWARDS ARE UNDERLINED)

Algorithm	Pusher-v0	Carrier-v0	Lifter-v0	Catcher-v0	Thrower-v0
gSDE [17]	1.02 ± 0.75	0.48 ± 0.79	-0.59 ± 0.34	-4.30 ± 1.96	1.47 ± 0.35
Vanilla PPO [27]	6.59 ± 0.38	6.76 ± 0.90	$\underline{1.61 \pm 0.77}$	-3.32 ± 1.51	1.09 ± 0.85
RND [15]	5.66 ± 0.78	2.88 ± 2.13	-0.80 ± 0.28	-3.52 ± 1.77	1.90 ± 0.79
E3B [14]	0.04 ± 0.23	-0.06 ± 0.72	0.23 ± 0.85	-4.22 ± 0.62	0.33 ± 0.34
PseudoCounts [16]	2.54 ± 1.33	2.22 ± 1.52	1.05 ± 1.01	-3.19 ± 2.03	0.36 ± 0.37
Disagreement [20]	0.26 ± 1.25	-0.41 ± 1.03	-1.04 ± 1.01	-4.84 ± 1.06	1.38 ± 0.57
NGU [16]	2.99 ± 0.58	2.38 ± 1.89	-0.36 ± 0.39	-4.49 ± 0.66	0.58 ± 0.36
Ours	$\underline{\mathbf{6.86 \pm 0.77}}$	$\underline{\mathbf{7.27 \pm 0.71}}$	$\mathbf{-0.02 \pm 1.15}$	$\underline{\mathbf{-2.68 \pm 2.04}}$	$\underline{\mathbf{2.02 \pm 0.95}}$

Algorithm 1 Learning algorithm with the proposed reward decomposition

Initialize policy π_θ and value function V_ϕ
Initialize predictor network for robot states f_ξ and target network for robot states f_{ξ^*}
Initialize predictor network for object states f_ζ, randomly initialize target network for object states f_{ζ^*}
$T \leftarrow$ total time steps
$\tau \leftarrow$ batch size
$u_f \leftarrow$ update frequency
for episode $= 1, ..., N$ **do**
  Sample initial state $s_0 \sim \rho(s_0)$
  for $t = 0, ..., T$ **do**
    Sample action $a_t \sim \pi_\theta(a_t \mid s_t)$, execute it and get s_{t+1}, e_t
    From s_{t+1} select robot state s_r and object state s_o
    Compute intrinsic reward for robot state $i_t^r = \|f_{\xi^*}(s_{t+1}^r) - f_\xi(s_{t+1}^r)\|^2$
    Compute intrinsic reward for object state $i_t^o = \|f_{\zeta^*}(s_{t+1}^o) - f_\zeta(s_{t+1}^o)\|^2$
    Compute composite intrinsic reward $i_t = \lambda_r \cdot i_t^r + \lambda_o \cdot i_t^o$
    Compute total reward $r_t = e_t + i_t$
  end for
  if episode $\%$ u_f == 0 **then**
    Update normalization parameters
    Update π_θ and V_ϕ using $\{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T-1}$
    Update robot predictor network weights ξ to minimize $||f_{\xi^*}(s_{t+1}^r) - f_\xi(s_{t+1}^r)||^2$ and
    Update object predictor network weights ζ to minimize $||f_{\zeta^*}(s_{t+1}^o) - f_\zeta(s_{t+1}^o)||^2$
  end if
end for

reward e_t from the environment:

$$R_t = i_t + e_t.$$

We leverage this approach within the proximal policy optimization algorithm [27]. The following objective function is used to update policy parameters θ:

$$J(\theta) = \mathrm{E}_{\pi_\theta}\left[\log \pi_\theta(a_t|s_t) \cdot \hat{A}^{\pi_\theta}(s_t, a_t)\right], \quad (4)$$

where $\hat{A}^{\pi_\theta}(s_t, a_t)$ is the advantage function estimate. The policy π_θ is a neural network (multi-layer perception as described in [28]) with parameters θ, which predicts action a given observation s during the MDP interaction loop described in Section II. The objective $J(\theta)$ is maximized via the Adam algorithm [26]. To address the instabilities caused by the non-linearity of the network π_θ in training where small changes in parameters lead to large changes in the performance, the clipped surrogate objective is introduced so that the new policy is not too different from the old one:

$$L^{\mathrm{CLIP}}(\theta) = \mathrm{E}_{\pi_\theta}\left[\min\left(\hat{r}_t(\theta) \cdot \hat{A}_t, \mathrm{clip}(\hat{r}_t(\theta), 1-\epsilon, 1+\epsilon) \cdot \hat{A}_t\right)\right], \quad (5)$$

where:

- ϵ is hyperparameter that defines the clip range,
- $\hat{r}_t(\theta) := \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$ is the ratio of the new policy's probability to the old policy's probability for action a_t in state s_t, indicating the divergence between $\pi_{\theta_{\mathrm{old}}}$ and π_θ.
- $\hat{A}_t := \hat{A}^{\pi_{\mathrm{old}}}(s_t, a_t)$ estimates how much better an action a_t in state s_t was compared to the average in terms of reward.

By optimizing objective (5), it is ensured that the agent is choosing the safest possible policy update. Combining the elements discussed above, the proposed decomposition method is presented in Algorithm 1.

IV. EXPERIMENTAL RESULTS

We validated the proposed method in simulation on the soft robotic benchmark EvoGym [25] against state-of-the-art exploration methods. Table I shows the results of the learned action policy evaluation. In order to check the performance of the method and baselines, a variety of challenging high-dimensional manipulation tasks were employed. Specifically, we used 5 tasks of different levels of difficulty according to the documentation of the benchmark (Pusher, Carrier – easy, Thrower – medium, Lifter, Catcher – hard). As mentioned in Section IV, we validated the method on two kinds of tasks: tasks where the manipulation object was initialized statically and tasks where it was dynamic. The object is static and does not move at the beginning of the episode in Lifter-v0 and Pusher-v0 tasks. The dynamic initialization where the object moves at the beginning is implemented in tasks Carrier-v0 (the object is initialized above the robot and slides into its gripper), Catcher-v0 (the object falls from high altitude and exhibits significant rotational movement while falling), and Thrower-v0 (the object is initialized just above the robot which causes a slight movement at the beginning).

TABLE IV

ABLATION STUDY SHOWING THE EFFECT OF VARYING PARAMETERS λ_o AND λ_r (MEAN VALUE AND STANDARD DEVIATION)

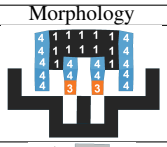
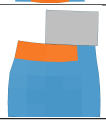
Parameters (λ_o, λ_r)	**Pusher-v0**	**Carrier-v0**	**Lifter-v0**	**Catcher-v0**	**Thrower-v0**
(0.95, 0.05)	6.48 ± 0.71	7.07 ± 1.51	$\underline{0.19 \pm 1.10}$	-3.67 ± 1.62	$\underline{2.16 \pm 1.16}$
(0.9, 0.1)	$\underline{7.18 \pm 0.81}$	7.27 ± 0.71	-0.02 ± 1.15	-2.68 ± 2.04	2.02 ± 0.95
(0.8, 0.2)	6.46 ± 0.67	$\underline{7.38 \pm 0.74}$	-0.45 ± 0.91	-4.00 ± 1.66	1.76 ± 0.90
(0.7, 0.3)	6.32 ± 0.54	5.74 ± 2.40	-0.15 ± 1.05	-3.45 ± 1.24	1.73 ± 1.01
(0.6, 0.4)	5.63 ± 1.54	5.87 ± 2.34	-0.85 ± 0.48	$\underline{-2.39 \pm 1.97}$	1.85 ± 0.63

For our purposes of testing the learned performance of soft robots on tasks, we manually designed intuitive robot morphologies in order to have an idea of how they should perform in a near-optimum way. All the robots are modular, and the building blocks are squares of different colors that have a certain function (blue or orange – actuators that can expand vertically or horizontally, grey – soft non-actuated blocks, black – rigid non-actuated blocks) as showcased in the first row of Table II.

The dimensionality of control depends on the number of actuators in the robot, and for our robots, it varies from 10 to 25. The actions are continuous and normalized to the range $[-1, 1]$, where -1 stands for maximum shrunk and 1 for maximum expansion. The complete list of robot morphologies, environments, and their descriptions is shown in Table II. In order to test the performance, we employ stable baselines 3 [28] as a reliable implementation of proximal policy optimization (PPO) [27] and use different intrinsic rewards strategies. The parameters of the PPO algorithm remain unchanged for different intrinsic reward strategies to ensure a fair comparison.

TABLE II

MORPHOLOGIES OF ROBOTS IN DIFFERENT PROBLEMS

Morphology	Environment	Task description
	Lifter-v0	Robot has to take the black object (1×2) out of a hole (3×2) vertically as high as possible.
	Carrier-v0	The robot should carry the grey object (3×2) as far as possible in the positive x-direction.
	Pusher-v0	The robot should push the grey object (3×2) as far as possible in the positive x-direction
	Catcher-v0	The robot should catch a fast-moving and rotating grey object (3×2).
	Thrower-v0	The robot should throw the grey object (3×2) as far as possible in the positive x-direction

For the reliable implementations of baseline exploration algorithms, we use the RLeXplore toolkit [29]. We run a total of 550 end-to-end learning experiments - 10 experiments for a fixed number of 1000 000 steps with different random initialization for each algorithm and robot, and report the mean value and standard deviation. The experiments were performed on a server mounting two GPUs NVIDIA RTX™ 6000 Ada Generation 64GB and 128GB RAM. The intrinsic rewards are only added during training, while during evaluation, only extrinsic rewards from the environment are considered to ensure a fair comparison.

While in this work we focus on the validation of the performance in a simulated environment to validate our hypotheses as a proof of concept, there is a strong potential for transition to physical soft robotic systems in real-world applications. Modular soft robotic systems similar to the one we simulate can be obtained via soft lithography for the production of hollow cubes and magnets for connecting them [30]. Also, a fixed morphology can be created via molding, providing a relatively small simulation to reality gap for silicone-based soft robots, as shown in [31].

A. Ablation Experiments

In order to investigate the influence of each component of the decomposed intrinsic reward on overall task performance, we analyze how the choice of parameters λ_r and λ_o affects the learning process.

We evaluated five distinct parameter configurations and, for each of them, performed 10 training runs with different random initialization. For each environment, there's a different extrinsic reward function, as defined in [25], that significantly affects the learning process. We report the summary of the results in Table IV as a mean reward with a standard deviation per environment. Higher values of parameter λ_o were chosen to shift attention from the exploration of the new states of the robot to the exploration of the new states of the object. The consistent presence of differences in the performance across different parameter settings confirms our fundamental hypothesis that the proposed decomposition of intrinsic rewards meaningfully influences learned behavior, as it is possible to find coefficients λ that improve the performance. Generally, these coefficients depend on the task, and in our problems, higher values of λ_o yielded better results. Moreover, the observed performance patterns suggest that the relationship between the choice of weights for components and task performance in the general case is nonlinear. However, higher values of λ_o mostly lead to higher rewards in the performance. For instance, in Lifer-v0 the results reveal a monotonic increase in performance when the attention shifts towards the manipulation object's intrinsic rewards from 0.7 to 0.95. Similarly, in Pusher-v0 the performance improves monotonically when λ_0 changes from

0.6 to 0.9. Thus, the adjustments in the weighting scheme favoring either the robot or the object component lead to systematic changes in the learned behavior.

Overall, these findings contribute to the understanding of intrinsic motivation in reinforcement learning for soft robotic manipulation and provide practical insights for the implementation of intrinsic reward decomposition in complex manipulation scenarios involving soft robots and deformable objects.

V. Conclusions

In this work, we addressed the crucial challenge of the exploration-exploitation trade-off in reinforcement learning for soft robotic manipulation where high-dimensional state spaces and complex nonlinear dynamics complicate effective skill acquisition. We developed a novel intrinsic attention-inspired reward decomposition method that separates robot states from manipulation object states. The proposed method efficiently guides learning agents towards task-relevant interactions by weighted aggregation of random network distillation intrinsic rewards. Experimental validation on a soft robotic manipulation benchmark indicates that our method outperforms state-of-the-art intrinsic motivation algorithms. Additionally, the comprehensive ablation study provides insights into how various weighting schemes affect task performance and learning efficiency. Overall, the results highlight the potential of attention-inspired intrinsic rewards and curiosity-driven methods for more effective skill acquisition and performance enhancement.

Future research directions include the extension of these techniques to the challenging setting of co-optimization of robot's morphology and control, and developing multi-task and generalist control policies, potentially with the use of large foundational models and deep physics-informed and information geometry methods for enhanced guidance toward high-rewarding regions.

References

[1] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of machine learning research*, vol. 22, no. 30, pp. 1–82, 2021.

[2] C. Liang, X. Gao, K. Wu, and Z. Pan, "Learning reduced-order soft robot controller," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 574–581, IEEE, 2023.

[3] J. Ma, F. Wu, Y. Chen, X. Ji, and Y. Ding, "Effective multimodal reinforcement learning with modality alignment and importance enhancement," *arXiv preprint arXiv:2302.09318*, 2023.

[4] A. Ramesh and B. Ravindran, "Physics-informed model-based reinforcement learning," in *Learning for Dynamics and Control Conference*, pp. 26–37, PMLR, 2023.

[5] D. Gupta, Y. Chandak, S. Jordan, P. S. Thomas, and B. C da Silva, "Behavior alignment via reward function optimization," *Advances in Neural Information Processing Systems*, vol. 36, pp. 52759–52791, 2023.

[6] N. Chentanez, A. Barto, and S. Singh, "Intrinsically motivated reinforcement learning," *Advances in neural information processing systems*, vol. 17, 2004.

[7] P. Dhakan, K. Merrick, I. RaÃ±Ã³, and N. Siddique, "Intrinsic rewards for maintenance, approach, avoidance, and achievement goal types," *Frontiers in Neurorobotics*, vol. 12, Oct. 2018.

[8] Z. Zheng, J. Oh, and S. Singh, "On learning intrinsic rewards for policy gradient methods," *Advances in neural information processing systems*, vol. 31, 2018.

[9] C. Colas, T. Karch, O. Sigaud, and P.-Y. Oudeyer, "Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey," *Journal of Artificial Intelligence Research*, vol. 74, pp. 1159–1199, 2022.

[10] S. Bhagat, H. Banerjee, Z. T. H. Tse, and H. Ren, "Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges," *Robotics*, vol. 8, p. 4, 2019.

[11] M. Vecerík, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. M. O. Heess, T. Rothörl, T. Lampe, and M. A. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *ArXiv*, vol. abs/1707.08817, 2017.

[12] B. Eysenbach and S. Levine, "Maximum entropy rl (provably) solves some robust rl problems," *arXiv preprint arXiv:2103.06257*, 2021.

[13] V. Zangirolami and M. Borrotti, "Dealing with uncertainty: Balancing exploration and exploitation in deep recurrent reinforcement learning," *Knowledge-Based Systems*, vol. 293, p. 111663, 2024.

[14] M. Henaff, R. Raileanu, M. Jiang, and T. Rocktäschel, "Exploration via elliptical episodic bonuses," *Advances in Neural Information Processing Systems*, vol. 35, pp. 37631–37646, 2022.

[15] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.

[16] A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, *et al.*, "Never give up: Learning directed exploration strategies," *arXiv preprint arXiv:2002.06038*, 2020.

[17] A. Raffin, J. Kober, and F. Stulp, "Smooth exploration for robotic reinforcement learning," in *Conference on robot learning*, pp. 1634–1644, PMLR, 2022.

[18] M. Kubovčík, I. Dirgová Luptáková, and J. Pospíchal, "Signal novelty detection as an intrinsic reward for robotics," *Sensors*, vol. 23, no. 8, p. 3985, 2023.

[19] S. J. Russell and A. Zimdars, "Q-decomposition for reinforcement learning agents," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 656–663, 2003.

[20] D. Pathak, D. Gandhi, and A. Gupta, "Self-supervised exploration via disagreement," in *International conference on machine learning*, pp. 5062–5071, PMLR, 2019.

[21] H. Van Seijen, M. Fatemi, J. Romoff, R. Laroche, T. Barnes, and J. Tsang, "Hybrid reward architecture for reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[22] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, "Explainable reinforcement learning via reward decomposition," in *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[23] R. Ma, J. Luijkx, Z. Ajanovic, and J. Kober, "Explorllm: Guiding exploration in reinforcement learning with large language models," *arXiv preprint arXiv:2403.09583*, 2024.

[24] Z. Liu, X. Yang, Z. Liu, Y. Xia, W. Jiang, Y. Zhang, L. Li, G. Fan, L. Song, and B. Jiang, "Knowing what not to do: Leverage language model insights for action space pruning in multi-agent reinforcement learning," *arXiv preprint arXiv:2405.16854*, 2024.

[25] J. Bhatia, H. Jackson, Y. Tian, J. Xu, and W. Matusik, "Evolution gym: A large-scale benchmark for evolving soft robots," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2201–2214, 2021.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[28] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[29] M. Yuan, R. C. Castanyer, B. Li, X. Jin, G. Berseth, and W. Zeng, "Rlexplore: Accelerating research in intrinsically-motivated reinforcement learning," *arXiv preprint arXiv:2405.19548*, 2024.

[30] A. Vergara, Y.-s. Lau, R.-F. Mendoza-Garcia, and J. C. Zagal, "Soft modular robotic cubes: toward replicating morphogenetic movements of the embryo," *PloS one*, vol. 12, no. 1, p. e0169179, 2017.

[31] S. Kriegman, A. M. Nasab, D. Shah, H. Steele, G. Branin, M. Levin, J. Bongard, and R. Kramer-Bottiglio, "Scalable sim-to-real transfer of soft robot designs," in *2020 3rd IEEE international conference on soft robotics (RoboSoft)*, pp. 359–366, IEEE, 2020.