Yang X
Cardiff University
School of Engineering

Ji Z
Cardiff University
School of Engineering

ADVANCED MANUFACTURING

Accelerating Multi-step Sparse Reward Reinforcement Learning

After the great successes of deep reinforcement learning (DRL) in recent years, developing methods to speed up DRL algorithms for more complex tasks closer to those in the real world has become increasingly important. In particular, there is a lack of research on long-horizon tasks that contain multiple subtasks or intermediate steps and can only provide sparse rewards at task completion point. This paper suggests to 1) use human priors to decompose a task and provide abstract demonstrations – the correct sequences of steps to guide exploration and learning, and 2) adjust the exploration parameters adaptively according to the online performances of the policy. The proposed ideas are implemented on three popular DRL algorithms, and experimental results on gridworld and manipulation tasks prove the concept and effectiveness of the proposed techniques.

Keywords:
Deep reinforcement learning, abstract demonstration, adaptive exploration, sparse reward, long-horizon manipulation.

Corresponding author:
JiZ1@cardiff.ac.uk

X. Yang and Z. Ji, 'Accelerating Multi-step Sparse Reward Reinforcement Learning', *Proceedings of the Cardiff University Engineering Research Conference 2023*, Cardiff, UK, pp. 86-90.

doi.org/10.18573/conf1.u

INTRODUCTION

Deep reinforcement learning (DRL) has achieved important progresses in the field of recommendation systems, computer games, navigation, economics, etc. [1]. However, DRL algorithms still struggle to learn tasks with long horizon, multiple intermediate steps, and sparse task completion rewards. In the real world, many robotic manipulation tasks exhibit such characteristics. For example, for the block-pushing task shown in Fig. 1, the robot needs to learn to open the chest before it can learn to push the block into the chest. With only a sparse reward signal based on task completion, such a task is hopeless for state-of-the-art continuous control DRL algorithms. This paper seeks to improve the performances of DRL in such tasks.

Fig. 1. Visualisation of a multi-step pushing task.

For such long horizon tasks, classic methods have provided successful examples of adopting manual task decomposition. For example, the popular task and motion planning (TAMP) methods employ domain languages to describe subtasks and skills, and search in the hybrid space of discrete subtasks and continuous manipulator motions for a solution for a long-horizon manipulation problem [2]. These methods, however, require the access to an accurate dynamic model of the world, which is one of the bottlenecks that limits their applications. On the other hand, this paper focuses on model-free DRL methods [1], which is freed from the assumption of having an accurate system dynamic model. The downside, however, is the difficulty of training DRL agents with a long task horizon and a sparse reward function.

In previous research, human demonstrations, in the forms of kinesthetic motion trajectories, have been one of the important options to help DRL in such difficult scenarios [3,4]. However, these methods are not scalable due to the difficulty of collecting them. In response, this paper adopts an abstract form of demonstrations that consist of the correct sequences of the task steps to be learnt and achieved, given the access to a task decomposition scheme. Specifically, abstract demonstrations 1) do not encode human biases into the robot motions and 2) are much easier to collect compared to kinesthetic teaching. The idea is plausible because the short motion trajectories in between two subtasks can be learnt efficiently with hindsight experience replay (HER) even in the face of sparse reward signals [5].

Another issue that slows down learning is related to the design of the exploration strategy of DRL agents. In particular, most exploration strategies of DRL agents at the current stage is task-agnostic [6]. They are therefore not exactly suitable for multi-step tasks, in which the later task steps depend heavily on the former ones. For example, stacking the fourth block would require the previous blocks to be stacked well. As a result, the default setting of using a strategy that explores constantly or decays the randomness in a task-agnostic way [6] will have difficulty dealing with multi-step tasks.

In response to this limitation, this work suggests to adapt the exploration parameters of a DRL agent in accordance with its online performances of each task step. The main idea of this adaptive exploration strategy is to reduce unnecessary exploration when the target subtask has been well-learnt.

MATERIALS AND METHODS

This research employs the recent framework of goal-conditioned reinforcement learning (GRL) [5], in which a universal policy $\pi(a|s,g)$ or universal q function $Q^{\pi}(s,g,a)$ is optimised towards the expected maximum discounted cumulated future goal-based rewards $\mathbb{E}_{\pi}[G] = [\sum_{t=0} \gamma^{t} R_{t}(s,g,a)]$, where $\gamma \in [0,1]$ is the discount factor that determines the importance of future rewards. The term *universal* implies that the policy or q function make decisions for a set of goals, instead of a single goal as in standard RL paradigm [7].

The following will introduce 1) the experiment tasks and the implementation of the goal-conditioned Markov decision processes (GMDPs), 2) the main learning algorithms, 3) the abstract demonstration method, and 4) the adaptive exploration method.

Tasks and the GMDP

To examine the effectiveness of the proposed methods, three robotic manipulation tasks developed in simulation are used in the experiments, including the ChestPush, the ChestPick and the BlockStack tasks from the Pybullet Multigoal (PMG) simulation software [8], as shown in Fig. 2. From left to right in Fig. 2, the robot needs to 1) open the chest and push the block into it, 2) open the chest, grasp the block and drop it into the chest, and 3) pick and stack the blocks as indicated by the transparent spheres.

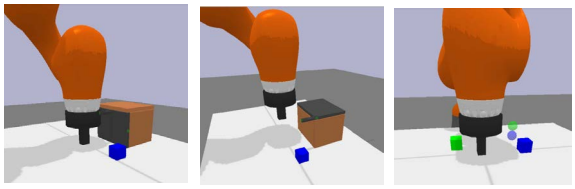

Fig. 2. The experimented tasks.

The GMDP descriptions of the tasks are exactly the same as that presented by in the simulation package [8], except that the representation of the goals is extended. The state of the system consists of the absolute positions and Euler orientations of the blocks, the relative positions and Euler orientations of the blocks w.r.t. the gripper tip, the relative linear and angular velocities of the blocks w.r.t. the gripper tip, the absolute position and velocity of the gripper tip and the finger width. At the beginning of an episode, the algorithm is given a final goal that specifies the desired positions of the blocks and the gripper and the desired width between the fingers. The algorithm gives actions based on the state and goal to move the gripper in the Cartesian space and control its finger width. It is given a reward of 0 when the final goal is achieved, and a reward of -1 otherwise. A desired goal is said to be achieved when its Euclidean distance to the actually achieved goal is less than a threshold .

Reinforcement Learning Algorithms

Two RL algorithms for continuous action space tasks: deep deterministic policy gradient (DDPG) [9] and soft actor critic (SAC) [10] are employed as the base algorithms. A short description of the two algorithms is given below, and the readers are referred to the original papers for further details. Also, as one shall see, the proposed acceleration techniques described in the following subsections are not limited to these two algorithms.

DDPG and SAC are both actor-critic algorithms [9,10]. DDPG seeks to learn a critic network, $q_{\boldsymbol{\omega}}^{\mu}$, that predicts the expected return for a pair of state and action, and a *deterministic* actor network, $\mu_{\boldsymbol{\theta}}$, that takes into the state and predicts the action that maximise the output of the critic network. For the goal-conditioned version, it simply takes the goal as an extra input to the policy and critic. Implementation-wise, the goal vector is concatenated into the input vector. As a deterministic policy agent, DDPG needs a behavioural policy to collect exploration data. In this work, the base exploration used by DDPG is the epsilon-Gaussian (EGa) strategy [5]:

$$\mu_b(a|s,g) = \begin{cases} \mathcal{N}(\mu(s),\sigma), & z \geq \epsilon \\ \mathcal{U}(A), & z < \epsilon \end{cases} \quad (1)$$

where, $\mathcal{N}$ stands for a normal distribution, $\mathcal{U}$ stands for a uniform distribution, $z \sim \mathcal{U}(0,1)$ and $\epsilon \in [0,1]$. In short, the EGa strategy takes a random action with probability ϵ, and take the action output by the learnt policy with zero-mean Gaussian noise with a probability of $1-\epsilon$. The algorithm optimises the following objectives for the critic and the actor networks, respectively:

$$J_{DDPG}(\boldsymbol{\theta}') \approx \mathbb{E}_{(s,g)\sim D,\ a=\mu_{\boldsymbol{\theta}'}(s,g)}\left[q_{\boldsymbol{\omega}'}^{\mu}(s,g,a)\right] \quad (2)$$

$$J_{DDPG}(\boldsymbol{\omega}') \approx \mathbb{E}_{(s,g,a,r,s')\sim D}\left[\frac{1}{2}\left(y - q_{\boldsymbol{\omega}'}^{\mu}(s,g,a)\right)\right] \quad (3)$$

where, $y = r + \gamma \cdot q_{\boldsymbol{\omega}''}^{\pi}\left(s', g, a'|_{a'=\pi_{\boldsymbol{\theta}''}(s',g)}\right)$ is the target Q value computed by the target critic network with weights, and are the parameters of the main policy and Q networks, stands for the replay buffer.

The SAC algorithm is also an actor-critic algorithm, but instead employs a stochastic actor [10]. In short, SAC uses a neural network to predict the mean and deviations of a Gaussian actor policy. The main difference with DDPG is twofold.

First, the SAC optimises the critic and actor not only towards the maximum return direction, but also the maximum policy entropy:

$$J_{SAC}(\boldsymbol{\theta}'') \approx \mathbb{E}_{(s,g)\sim D,\ a\sim\pi_{\boldsymbol{\theta}''}(s,g)}[\alpha \log \pi_{\boldsymbol{\theta}''} - q_{\boldsymbol{\omega}''}^{\pi}(s,g,a)] \quad (4)$$

$$J_{SAC}(\boldsymbol{\omega}'') \approx \mathbb{E}_{(s,g,a,r,s')\sim D}\left[\frac{1}{2}\left(y - q_{\boldsymbol{\omega}''}^{\pi}(s,g,a)\right)\right] \quad (5)$$

where, $y = r + \gamma \cdot q_{\boldsymbol{\omega}''}^{\pi}\left(s', g, a'|_{a'=\pi_{\boldsymbol{\theta}''}(s',g)}\right) - \alpha \log \pi_{\boldsymbol{\theta}''}$ is the soft target Q value, $\boldsymbol{\theta}''$ and $\boldsymbol{\omega}''$ are the parameters of the main policy and Q networks, and α is the temperature parameter.

Secondly, the SAC algorithm conducts exploration by sampling from its own policy, whose randomness is determined by the output of the neural network. As the objective partly maximises the entropy of the policy, it can retain a certain degree of exploration without collapsing into a deterministic policy.

These two algorithms are selected because they are well-known representatives of recent model-free DRL algorithms. Also, they both retain the state-of-the-art performances of popular continuous control benchmarks [9,10]. However, they are not tailored for the kind of long-horizon and multi-step tasks considered in this paper, and the following will then introduce the two techniques that can speed up these two agents.

Abstract demonstrations

The idea of abstract demonstrations (AD) is very similar to the human practices of learning to build a Lego house or assemble a furniture using a user manual that specifies a series of key task steps.

AD assumes the access to a task decomposition scheme that produces a set of task steps, each corresponds to a subset of goals in the GRL framework. In this work, the decomposition scheme comes from human priors, but it is certainly interesting to develop automated task decomposition methods in the future. Specifically, the given tasks are decomposed as follows:

- ChestPush: push the door open reach the blue block push the block into the chest.
- ChestPick: push the door open grasp the blue block move to the top of the chest drop the block.
- BlockStack: grasp the first block move to the target position grasp the second block put it on top of the first one.

Instead of providing only the final goal at the start of training, the agent is given the subgoals associated with the subtasks in the correct order. When a subgoal is achieved, the next one according to the demonstrations will be given, until the final goal is reached or the episode runs out of time. A parameter, $\eta = 0.75$ is used to control in how many episodes during training that the agent is demonstrated.

Adaptive exploration

To adapt the exploration parameters, the adaptive exploration (AE) method first keeps a record of the online performances on each subtask, and then uses the performance to scale the exploration parameters. The following will explain the implementations on the DDPG and SAC agents.

For convenient usage, the average success rates of the agent over 30 testing episodes on each subtask are used as the performance metric, denoted as a n-dimensional vector, $\boldsymbol{S}$, where N is the number of subtasks. This evaluation run is performed after every training epoch (800 episodes). To ensure a smoothly changing behaviour of the performance record, the Polyak average of the success rate is used instead of the arithmetic mean:

$$\boldsymbol{S}^{-} \leftarrow (1-\tau_S)\cdot \boldsymbol{S}^{-} + \tau_S \cdot \boldsymbol{S} \quad (6)$$

where $\boldsymbol{S}^{-}$ is the Polyak averaged success rate vector and $\tau_S = 0.3$ is the update ratio.

To update the DDPG exploration strategy, Eq.1 is used with an individual ϵ^n and σ^n for the n-th step. All of them starts with $\boldsymbol{\epsilon_0} = 0.2$ and $\boldsymbol{\sigma_0} = 0.05$. After each evaluation run, these two parameters are updated as follows:

$$\boldsymbol{\epsilon} \leftarrow \boldsymbol{\epsilon}_0 \cdot (1 - \boldsymbol{S^-}), \boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}_0 \cdot (1 - \boldsymbol{S^-}) \quad (7)$$

For the SAC agent, the evaluation success rates are used to scale the deviations predicted by the actor neural network for each subtask after an evaluation run:

$$\boldsymbol{\sigma}^{SAC} \leftarrow \widehat{\boldsymbol{\sigma}}^{SAC} \cdot (1 - \boldsymbol{S^-}) \quad (8)$$

where $\widehat{\boldsymbol{\sigma}}^{SAC}$ is predicted by the actor network.

RESULTS

To examine the effectiveness of the proposed ideas, ablative experiments are conducted with the three robotic manipulation tasks. All codes are available on GitHub: https://github.com/IanYangChina/A-2-paper-code.

As shown in Fig. 3, the DDPG and SAC agents are both run on the three tasks in their original forms (Vanilla), aided with abstract demonstrations (AD), and aided with both methods (ADAE). The first row is the performances of the DDPG agents, and the second row is for the SAC agents. For each row, the success rates correspond to the three tasks from left to right: ChestPush, ChestPick and BlockStack.

In Fig. 3, both agents have substantial improvements on the three tasks with the help of the abstract demonstrations (compare the blue and green lines). As the task becomes more difficult (from left to right), the improvements over the vanilla algorithms become more obvious. This suggests that providing abstract demonstrations can substantially accelerate multi-step sparse reward reinforcement learning.

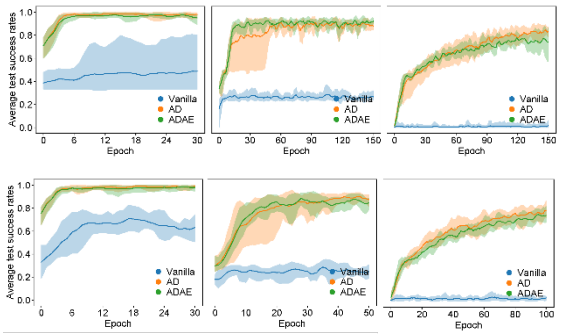

Fig. 3. Test success rates.

Secondly, the adaptive exploration has gained less obvious performance improvements in addition to the abstract demonstrations (compare the orange and green lines). However, it does reduce the variances of the success rates, indicating a more stable training processes.

DISCUSSION

The proposed abstract demonstrations and adaptive exploration methods have been proved in the result section to improve the performance of two popular DRL algorithms in long-horizon, multi-step and sparse reward continuous control tasks.

Compared to previous studies which use motion-level trajectories as demonstrations [3,4], abstract demonstrations are much easier to collect and implement. However, there is no free lunch. It requires a given task decomposition scheme. In the future, developing automated task decomposition and subgoal discovery methods are promising directions. Also, abstraction demonstrations can be used along with kinesthetic demonstrations, when the motions in between subtasks are still too hard to learn by pure exploration.

There is not much research that paid attention to task-oriented exploration strategy design [6]. In fact, the idea is compatible to use with many task-agnostic exploration strategies, as shown by the DDPG and SAC implementations in this work. For example, one can use a success rate to scale the noise injection rate in [11] or the random action probability of the popular epsilon-greedy method [7].

In addition to what were mentioned, more efforts are demanded to implement and evaluate the proposed methods on more realistic tasks, especially tasks with complex and noisy observations. For example, the representation and generation of subtasks and high-dimensional goals [12]. Such studies will also empower techniques in other areas of robotic research.

Acknowledgments
Xintong thanks the China Scholarship Council for the financial support during his PhD career (No. 201908440400).

Conflicts of interest
The authors declare no conflict of interest.

REFERENCES

[1] A. Lazaridis, A. Fachantidis, and I. Vlahavas, 'Deep Reinforcement Learning: A State-of-the-Art Walkthrough', *jair*, vol. 69, pp. 1421–1471, Dec. 2020. doi.org/10.1613/jair.1.12412

[2] C. R. Garrett e*t al.*, 'Integrated Task and Motion Planning', *Annu Rev Control Robot Auton Syst*, vol. 4, no. 1, pp. 265–293, May 2021. doi.org/10.1146/annurev-control-091420-084139

[3] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, 'Recent Advances in Robot Learning from Demonstration', *Annu Rev Control Robot Auton Syst*, vol. 3, no. 1, pp. 297–330, May 2020. doi.org/10.1146/annurev-control-100819-063206

[4] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, 'Overcoming Exploration in Reinforcement Learning with Demonstrations', in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD: IEEE, May 2018, pp. 6292–6299. doi.org/10.1109/ICRA.2018.8463162

[5] M. Andrychowicz e*t al.*, 'Hindsight experience replay [C]', *NeurIPS*, Long Beach, UK, 4-9 Dec. 2017. doi.org/10.48550/arxiv.1707.01495

[6] P. Ladosz, L. Weng, M. Kim, and H. Oh, 'Exploration in deep reinforcement learning: A survey', *Information Fusion*, vol. 85, pp. 1–22, Sep. 2022. doi.org/10.1016/j.inffus.2022.03.003

[7] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. in Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018.

[8] X. Yang, Z. Ji, J. Wu, and Y.-K. Lai, 'An open-source multi-goal reinforcement learning environment for robotic manipulation with pybullet [C]', *Towards Autonomous Robotic Systems*. Springer, Cham, Lincoln, UK. 8-10 Sep. 2021. doi.org/10.1007/978-3-030-89177-0_2

[9] T.P. Lillicrap e*t al.*, 'Continuous control with deep reinforcement learning' [C] *ICLR*, San Juan, Puerto Rico, 2-4 May 2016. doi.org/10.48550/arXiv.1509.02971

[10] T. Haarnoja e*t al.*, 'Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor', [C] *ICML*, Stockholm, Sweden, 10-15 Jul. 2018. doi.org/10.48550/arXiv.1801.01290

[11] M. Fortunato e*t al.*, 'Noisy networks for exploration' [C] *ICLR*, Vancouver, Canada, 30 Apr.–3 May 2018. doi.org/10.48550/arXiv.1706.10295

[12] S. Pateria, B. Subagdja, A.-H. Tan, and C. Quek, 'End-to-End Hierarchical Reinforcement Learning With Integrated Subgoal Discovery', *IEEE Trans Neural Netw Learning Syst*, vol. 33, no. 12, pp. 7778–7790, Dec. 2022. doi.org/10.1109/TNNLS.2021.3087733

Cynhadledd Ymchwil Peirianneg Caerdydd
Cardiff Engineering Research Conference

E. Spezi and M. Bray (eds.) 2024. *Proceedings of the Cardiff University Engineering Research Conference 2023.* Cardiff: Cardiff University Press.
doi.org/10.18573/conf1

Cardiff University Engineering Research Conference 2023 was organised by the School of Engineering and held from 12 to 14 July 2023 at Cardiff University.

The work presented in these proceedings has been peer reviewed and approved by the conference organisers and associated scientific committee to ensure high academic standards have been met.

Cardiff University Press | Gwasg Prifysgol Caerdydd

First published 2024

Cardiff University Press
Cardiff University, PO Box 430
1st Floor, 30-36 Newport Road
Cardiff CF24 0DE

cardiffuniversitypress.org

Editorial design and layout by
Academic Visual Communication

ISBN: 978-1-9116-5349-3 (PDF)