An Autonomous Non-monolithic Agent with Multi-mode Exploration based on Options Framework

Anonymous Authors

***Abstract*—Most exploration research on reinforcement learning (RL) has paid attention to 'the way of exploration', which is 'how to explore'. The other exploration research, 'when to explore', has not been the main focus of RL exploration research. The issue of 'when' of a monolithic exploration in the usual RL exploration behaviour binds an exploratory action to an exploitational action of an agent. Recently, a non-monolithic exploration research has emerged to examine the mode-switching exploration behaviour of humans and animals. The ultimate purpose of our research is to enable an agent to decide when to explore or exploit autonomously. We describe the initial research of an autonomous multi-mode exploration of non-monolithic behaviour in an options framework. The higher performance of our method is shown against the existing non-monolithic exploration method through comparative experimental results.**

***Index Terms*—non-monolithic exploration, autonomous multi-mode exploration, options framework**

I. Introduction

Exploration is the crucial part of RL algorithms because it gives an agent the choice to uncover unknown states. There have been many RL exploration research studies with various viewpoints, such as intrinsic reward [1], [2], [3], [4], [5], [6], [7], [8], skill discovery [9], [10], [11], Memory base [12], [13], [14], [15], and Q-value base [16]. Although exploration research has evolved, it has concentrated on 'how to explore', which is how an agent selects an exploratory action. However, the exploration research regarding 'when to explore' has not been researched in earnest.

There are two types of methodology regarding 'when to explore', which are monolithic exploration and non-monolithic exploration. The noise-based monolithic exploration, a representative monolithic exploration, is that a noise, which is usually sampled from a random distribution, is added to the original action of a behaviour policy before putting the final action to an environment. The original action of policy and the noise to be added act as an exploitation and exploration respectively. Hence, the behaviour policy using monolithic exploration is affiliated to a time-homogeneous behaviour policy. However, in a non-monolithic exploration, the original action of a behaviour policy is not added to a noise. They act for their own purpose at a separate step. Therefore, the behaviour policy using a non-monolithic exploration belongs to a heterogeneous mode-switching one (Fig. 1).

We have investigated the initial research [17] of non-monolithic exploration. As the tentative work, there are still several limitations. Firstly, there is only one exploration policy (we call it one-mode exploration). An agent can require more choice of entropy of exploration mode which denotes more exploration modes greater than one-mode exploration. Secondly, the period of exploration to be controlled should be not fixed but variable. Thirdly, the research takes advantage of a simple threshold hyper-parameter function, which is named 'homeostasis', for the variable scale of trigger signals for switching exploration or exploitation. However, there should be a natural switching mechanism by using the policy itself. It also claims other informed triggers, which are action-mismatch-based triggers and variance-based triggers.

In this paper, we propose an autonomous non-monolithic agent with multi-mode exploration based on an options framework to resolve the above-mentioned considerations. Specifically, we adopt a Hierarchical Reinforcement Learning (HRL) as an options framework chaining together a sequence of exploration modes and exploitation in order to achieve switching behaviours at intra-episodic time scales. Thus, we can achieve a multi-mode exploration with the different entropy. In order to enable autonomous switching between exploration policies and an exploitation policy where the switching is based on intrinsic signals, we adapt a guided exploration using a reward modification of each switching mode. A robust optimal policy is also researched to maintain the potential performance.

Meanwhile, for this research the following 5 questions should be answered. How can an options framework be adopted in order to take advantage of the context of a HRL for exploration modes and exploitation? How does an agent have the flexibility of the exploration period? How does an agent get more entropy choice of exploration mode? How can an agent determine the switching of non-monolithic multi-mode exploration by itself without any subsidiary function such as 'homeostasis'? How does an agent avoid the inherent disturbance of a policy but have a robust optimal policy?

It is worth mentioning that there are no similar works in the literature, so the reference methods are partially based on the method proposed in [17] even though their work is not based on an options framework. Lastly, HIRO is also compared with our model. In the end, our exploration method shows a better performance.

The contributions of our research are summarized as follows.

- *Development of an options framework model supporting an autonomous non-monolithic multi-mode exploration:* We introduce a novel HRL model architecture to support

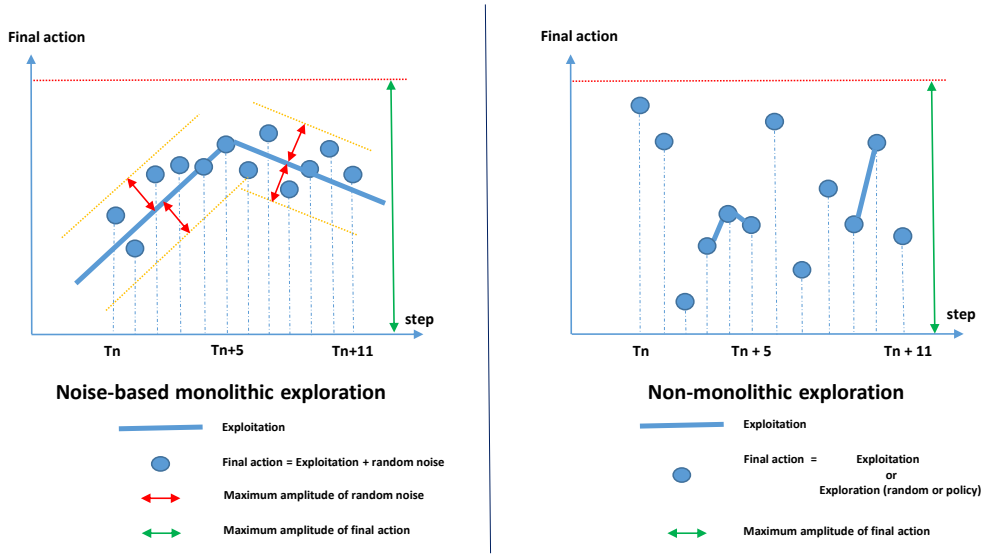


Fig. 1: An example of noise-based monolithic exploration (left) and non-monolithic exploration (right). The final action, which is a scalar in this example for the well understanding explanation, denotes the action of an agent represented with a solid circle at each step. The solid line denotes the exploitation, an original action of a behaviour policy. The solid circle in the noise-based monolithic exploration is a final action which combines the original action of a behaviour policy and a sampled bounded noise at each step. However, the solid circle in the non-monolithic exploration is defined according to the mode of each step, i.e. exploitation which is an an original action of a behaviour policy or exploration which is a random noise or a policy.

an autonomous non-monolithic multi-mode exploration for the first 3 research questions.

- *Development of a switching method for a non-monolithic exploration by using an inherent characteristic of a policy:* Our model use a guided exploration with a reward modification for the fourth research question.
- *Improved robustness of the policy:* A robust optimal policy can be ensured by taking advantage of an evaluation process for the last research question.

The rest of this research is explained as follows. Section II surveys the research of exploration and HRL related to our research. Section III explains our proposed model. Section IV describes the experiments for the performance measurement of our model compared with a non-monolithic model, [17], as a reference model and a monolithic exploration, HIRO. We discuss several acknowledged issues from the experiment in Section V. Finally, we present the conclusion of the current research and suggestions for future works in Section VI.

II. Related work

A. Options framework

The set of option, which is a generalized concept of action, over an MDP is comprised of a semi-Markov decision process (SMDP). Semi-MDPs are defined to deal with the different levels of an abstract action based on the variable period. HRL is a representative generalization of reinforcement learning where the environment is modelled as a semi-MDP [18].

Each action in non-monolithic exploration mode which adopts a multi-mode exploration has different effect during the different period. Thus the sequence of action is defined by taking advantage of an option framework for a multi-mode exploration.

B. Exploration

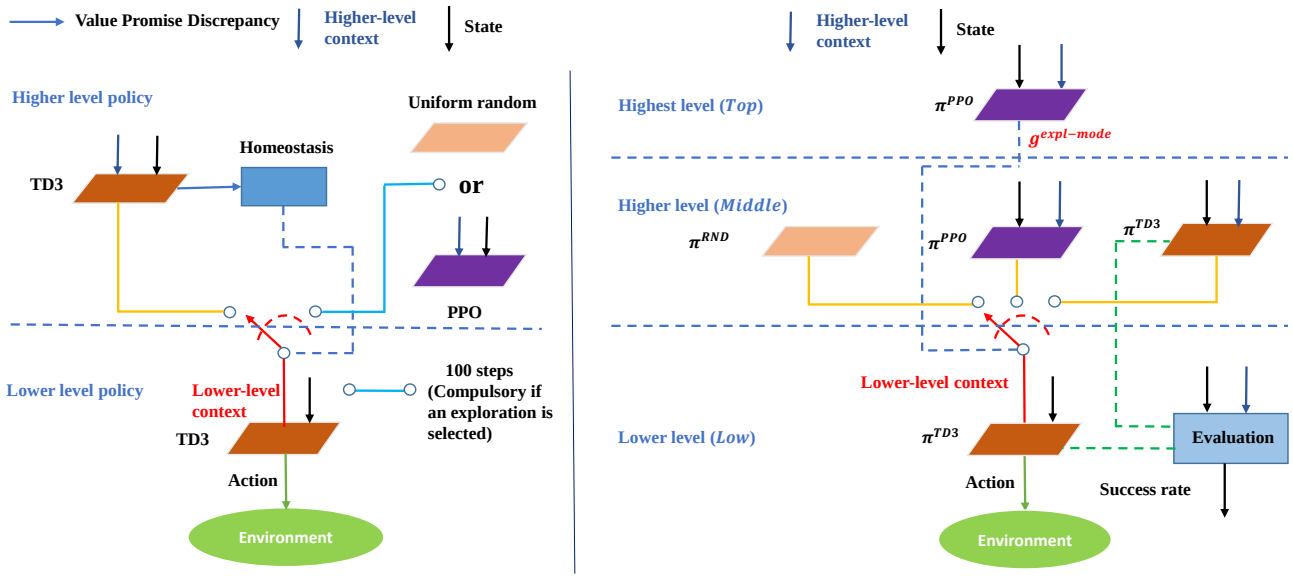
The various events for triggers have been considered whether they are acquired from uncertainty or not [19], [20], [21], [22], [23].

The experiment of [24] shows the efficacy of robot behaviour learning from self-exploration and a socially guided exploration supported by a human partner. [25] claims about the Bayesian framework which supports changing dynamics online and prevents conservativeness by using a variance bonus uncovering the level of transition of adversity. [26] claims Tactical Optimistic and Pessimistic (TOP) estimation for the value estimation strategy of optimistic and pessimistic online by using a quantile approximation [27]. Hence, the belief distribution is constructed by using a following quantile estimation.

$$q_{\tilde{Z}^{\pi}(s,a)} = q_{\bar{Z}(s,a)} + \beta q_{\sigma(s,a)} \tag{1}$$

where $q_{\bar{Z}(s,a)}$ and $q_{\sigma(s,a)}$ are the mean and standard deviation of quantile estimation respectively and Z(s,a) is a return random variable. The belief distribution is optimistic and pessimistic when $\beta \geq 0$ and $\beta < 0$ respectively.

[17] claims the importance of a non-monolithic exploration against a monolithic exploration. Its representative non-monolithic exploration method utilizes 'homeostasis' based on



Implemented reference model **Our non-monolithic model**

Fig. 2: The architecture of our suggested model (right) compared with that of the reference paper using a homeostasis [17] (left)

the difference of value function between k steps which is referred to as the 'value promise discrepancy', i.e. $D_{promise}(t-k,t)$.

$$D_{promise}(t-k,t) := \left|V(s_t-k) - \sum_{i=0}^{k-1}\gamma^i R_{t-i} - \gamma^k V(s_t)\right| \quad (2)$$

where $V(s)$ is the agent's value estimate at state s, γ is a discount factor and R is the reward.

The above-mentioned researches regarding the guidance-exploration framework, robust-MDP research and the adaptive optimism inspired our research on the basis of [17].

C. Hierarchical RL

The Semi-Markov Decision Process(SMDP) takes advantage of options defining a domain knowledge and can reuse the solutions to sub-goals [18]. [28] claims that an Adaptive Skills, Adaptive Partitions framework supports learning near-optimal skills which are composed automatically and concurrently with skill partitions, in which an initial misspecified model is corrected. [29] proposes an algorithm to solve tasks with sparse reward in which the research suggests an algorithm to accelerate exploration with the construction of options minimizing the cover time. [30] also deals with a sparse reward environment. Thus, it formalizes the concept of fast and slow curiosity for the purpose of stimulating a long-time horizon exploration. The option-critic architecture has the intra policy, which follows the option chosen by the policy over options until the end of the condition of option termination [31]. [32] claims that each policy in HRL, which utilizes a flow-based deep generative model for retaining a full expressivity, is trained through a latent variable with a bottom-up layer-wise method. HIRO claims the method to synchronize adjacent levels of hierarchical reinforcement learning to efficiently train the higher level policy.

Our model makes use of HIRO for our exploration research because it is a traditional goal-conditioned HRL.

III. Our Model

From the rising issue of value promise discrepancy used in [17], our research pays close attention to an autonomous multi-mode non-monolithic exploration model where an agent makes an action as to when an exploration mode starts and exits by itself. In addition, the expected model takes advantage of its inherent characteristic for the action. For the purpose, our research adopts an options framework.

An options framework especially in a goal-conditioned HRL is the appropriate consideration to control the multi-mode exploration through a fully state-dependent hierarchical policy. For the first research question proposed in Section I, our model has three levels of HRL as shown in Fig. 2 together with the implemented model of [17]. Our model names each of the levels according to the height of the level: Top, $Middle$ and Low. The policies are in each level: π_T^{PPO} for Top, π_M^{TD3}, π_M^{PPO} and π_M^{RND} for $Middle$ and π_L^{TD3} for Low.

The hierarchical control process is easy to systematically construct a multi-mode exploration, $g^{\text{expl-mode}}$, as the option

TABLE I: Key Notations.

Symbol	Meaning
t	action step
$state$	current state
$next\_state$	next state
Top	The highest level
$Middle$	The higher level
Low	The lower level
$action$	The action of *Low* level (The action of π_L^{TD3})
$target\_pos$	The context of *Top* and *Middle*
$goal$	The current lower-level context of three sub-policies of *Middle* level (The current goal for π_L^{TD3})
$next\_goal$	The next lower-level context of three sub-policies of *Middle* level (The next goal for π_L^{TD3})
R	The reward received from an environment (The sign of R is negative in Ant domain of OpenAI Gym)
π_T^{PPO}	The policy(on-policy) of *Top* level
π_M^{TD3}	The policy(off-policy) of *Middle* level
π_M^{PPO}	The policy(on-policy) of *Middle* level
π_M^{RND}	The policy (The uniform random for random policy) of *Middle* level
π_L^{TD3}	The policy(off-policy) of *Low* level
$g^{\text{expl-mode}}$	The action of π_T^{PPO} of *Top* level
$\alpha_{g^{\text{expl-mode}}}$	The preset value of α according to $g^{\text{expl-mode}}$
$S\_O_{g^{\text{expl-mode}}}$	The reference value of $Success\_ratio$ according to $g^{\text{expl-mode}}$
$loss$	The loss of π_T^{PPO} of *Top* level
S_E	The $Success\_rate$ of evaluation function of π^{TD3} of *Middle* level
$Done\_m$	The count of *Done* during the horizon of *Top* level
R_m	The sum of R during the horizon of *Top* level
$Count\_m$	The count during the horizon of *Top* level
S_O_m	The ratio of success count regarding $g^{\text{expl-mode}}$ of π_T^{PPO} during the horizon of *Top* level
ρ	The preset value of target rate, i.e. the average number of switches of the reference model

against a function control such as homeostasis. The exploration mode policy, π_T^{PPO}, can choose one of three policies of *Middle* level as follows.

$$g^{\text{expl-mode}} \sim \pi_T^{PPO} \tag{3}$$

Therefore, the value of $g^{\text{expl-mode}}$ denotes one of two exploration modes, which are uniform random and PPO, or one exploitation which is TD3. It also provides several control benefits for exploration as there are the inherent characteristics of the options framework.

In order to accomplish the purpose of our research, our options framework model comprises four elements: the inherent switching mode decision of the policy itself, empowering more entropy degrees for exploration, a guided exploration mode, and the use of an evaluation process for robustness.

A. The inherent switching mode decision of a policy itself

Since the inherent training method of π_T^{PPO} is used in our model, one policy of *Middle* level can be chosen according to an option, $g^{\text{expl-mode}}$, of π_T^{PPO}. π_T^{PPO} is synthesizing the reward-maximization of policy on all modes into its own policy without a subsidiary aid. This leads to the fact that the period of both exploration and exploitation is controlled by the inherent characteristic of an agent. In the end, all characteristic of the non-monolithic exploration mode policy can be integrated to the reward-maximization of policy for the second research question in Section I. We can verify the choice of a switching mode on the count of each exploration mode as shown in Section IV.

B. Empowering more entropy choice for exploration

Our model pursues multi-mode exploration for the exploration mode policy according to the degree of entropy of exploration mode as the degree of optimism. Our model has two exploration modes, which are a π_M^{RND} and a π_M^{PPO}, and one exploitation policy, π_M^{TD3}, in *Middle* level for the third research question in Section I. Thus, while an agent is being trained, we hypothesize that the degree of each entropy of three policies is as follows.

$$\pi_M^{RND} > \pi_M^{PPO} > \pi_M^{TD3} \tag{4}$$

Our model just consumes PPO for an exploration mode so that it will be discarded at the end of training. Our model takes care of only off-policy, TD3, as a final target policy. Meanwhile, PPO and TD3 are trained together whenever a data occurs due to one of three sub-policies of *Middle* level. If PPO is trained to some degree, our model expects that the performance of PPO is higher than the performance of uniform random regarding the result of exploration.

C. Guided exploration

There are two phases of a potential reward progress during the training of our agent. Thus, our model takes a guided exploration into consideration for the agent in order to keep the first phase. Since our model pursues an options framework in a goal-conditioned HRL, the exploration mode policy can follow a reward-maximizing policy so that the modification of reward R from an environment is conducted with a preset parameter $\alpha_{g^{\text{expl-mode}}}$ as follows.

$$R_{final} = R + \alpha_{g^{\text{expl-mode}}} * R \tag{5}$$

The value of $\alpha_{g^{\text{expl-mode}}}$ is differently or sometimes equally preset according to the type of $g^{\text{expl-mode}}$ as follows.

$$\alpha_{\text{uniform random}} > \alpha_{\text{ppo}} > \textit{or equal to } \alpha_{\text{td3}} \tag{6}$$

Finally, since the value of R_{final} is utilized in the training of the exploration mode policy, a reward-maximized option for the fourth research question in Section I is preferred by the exploration mode policy depending on the value of $\alpha_{g^{\text{expl-mode}}}$. As the value of $\alpha_{g^{\text{expl-mode}}}$ gets bigger, the occurrence probability of its exploration mode gets smaller.

D. Evaluation for robustness

For the second phase of the potential reward progress of our agent, our model adopts the online evaluation process to keep a robust optimal policy. The occurrence of success rate in the online evaluation process shows that the performance of an agent enters the second stage of reward progress in this research. From the second stage, our agent is required to have

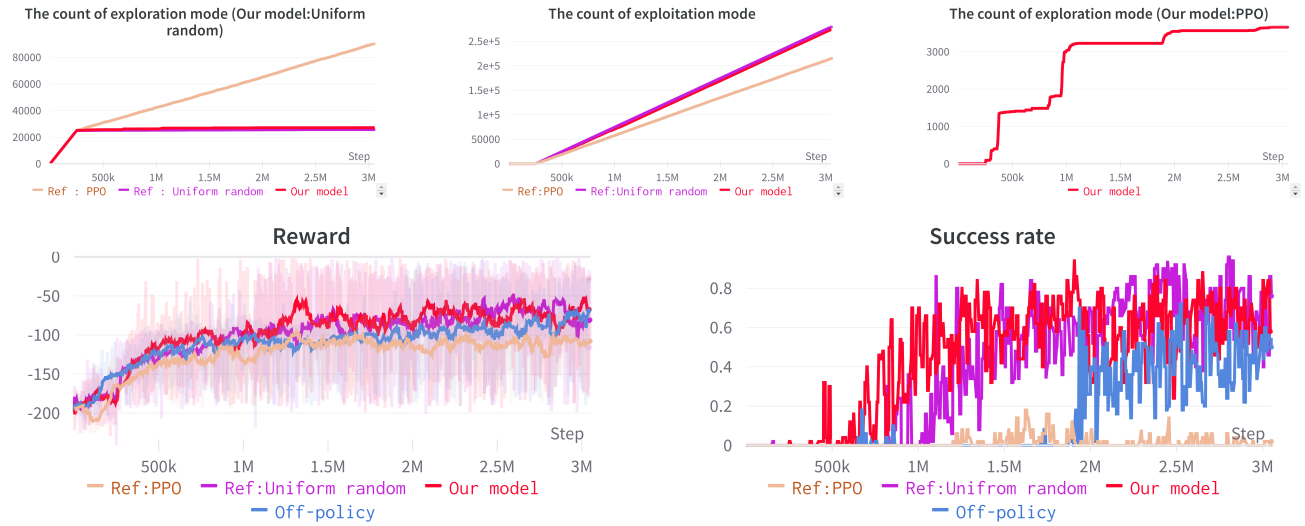


Fig. 3: The count of exploration modes and exploitation and the reward and success rate of higher level policy for our model, Ref:Uniform random, Ref:PPO and HIRO in Ant Push

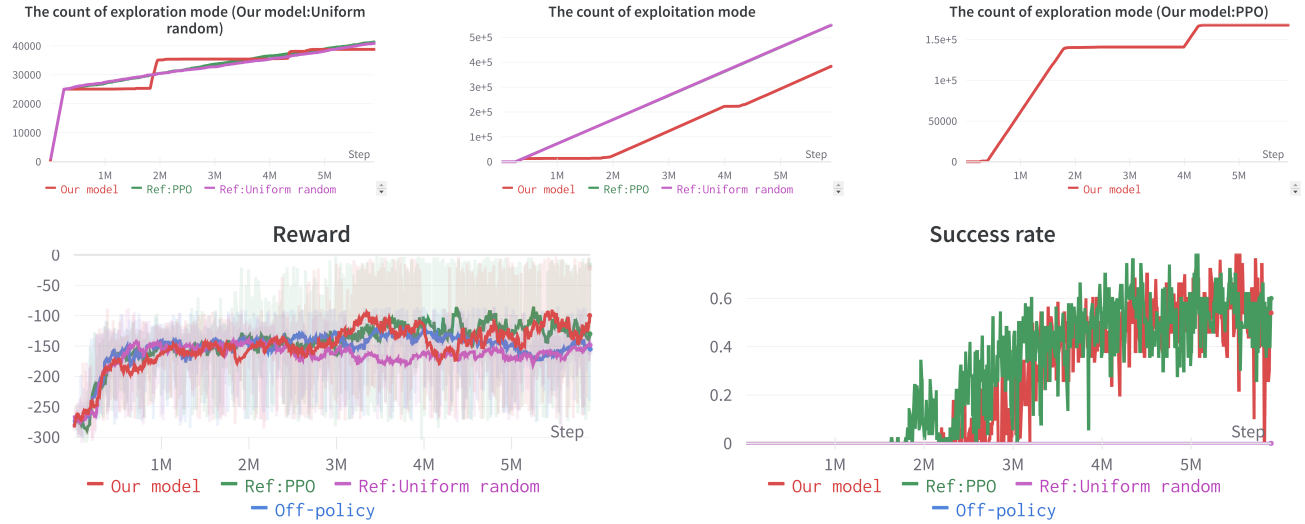


Fig. 4: The count of exploration modes and exploitation and the reward and success rate of higher level policy for our model, Ref:Uniform random, Ref:PPO and HIRO in Ant Fall

robust optimal policy by using online evaluation process. The online process evaluating the off-policy, π_M^{TD3}, operates every preset step. Then, it outputs the success rate, S_E, according to the type of $g^{\text{expl-mode}}$. Thus, $loss_{final}$ of the exploration mode policy, π_T^{PPO}, for the fifth research question in Section I is calculated in this research as follows.

$$loss_{final} = loss + S\_E * loss \tag{7}$$

In our model, as the online value of S_E increases, the loss of the exploration mode policy in the mode of uniform random and online policy becomes bigger than its online original loss.

IV. Experiments

The control of multi-mode exploration of our model as an autonomous non-monolithic agent is shown by the count of exploration modes and exploitation, since their counts are critical for the analysis of our model. Each count describes the current situation of reward-maximization of policy on all modes. Through the analysis, we aim to answer the following crucial question. *Can our model show better performance than that of the representative model of the reference paper, [17], and a noise-based monolithic exploration policy?* We evaluate our model and them in two tasks, Ant Push and Ant Fall, of Ant domain of OpenAI Gym. The reference models for the comparison are two models of [17], XU-

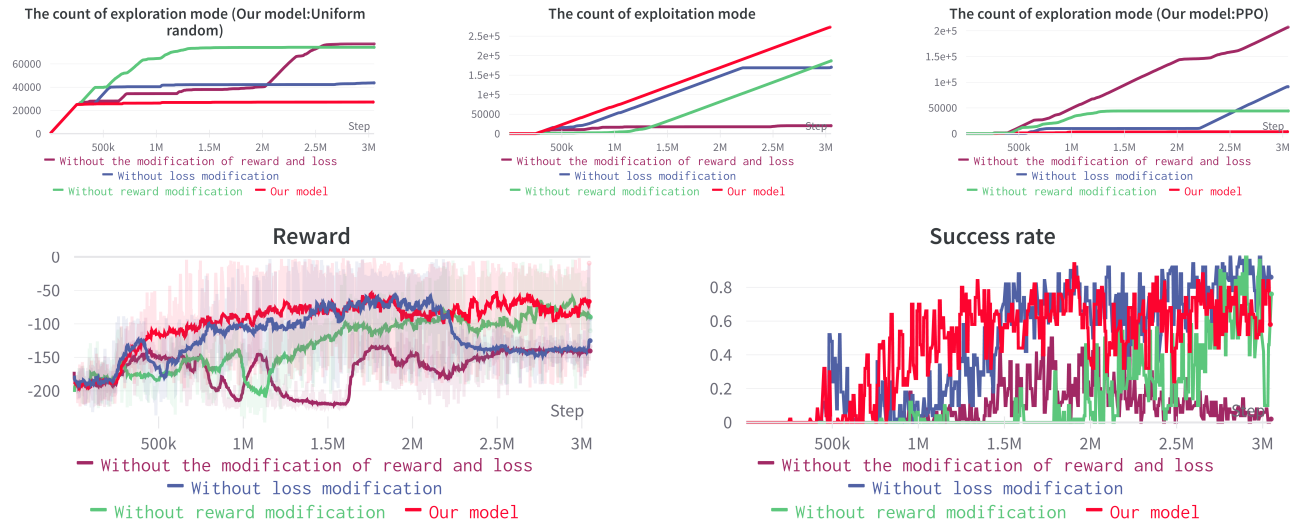


Fig. 5: Three types of ablation study against our normal model in Ant Push

intra(100, informed, p^*, X) and XI-intra(100, informed, p^*, X), which are called 'Ref:Uniform random' and 'Ref:PPO' respectively in our reference model[1]. PPO is utilized for the intrinsic explore mode of our reference model. A noise-based monolithic exploration policy is HIRO, which is composed of TD3 at each level. Our model and two reference models are also implemented based on HIRO. In order to evaluate the best performance among three models, we have the four analysis items as follows:

1) How many counts are assigned to each policy through whole training steps?
2) How does the transition of our model between exploration mode and exploitation occur compared with the forced exploration transition of reference model?
3) How much is the difference between uniform random and on-policy as the exploration policy of our model based on a guided exploration strategy?
4) How much does the evaluation process influence the performance of the second reward phase?

The results of Ant Push and Ant Fall are represented in Fig. 3 and Fig. 4 respectively. Moreover, the Appendix provides more implementation details including Algorithm[1]. If this paper is accepted, we will publish the implemented source code.

A. Comparison with the reference paper and pure off-policy

1) Ant Push: Our model outperforms all other models through almost all training steps. The exploitation of our model and two reference models occurs during the most of the training steps. The exploration mode of our model and two reference models takes place less than the exploitation of them. HIRO shows the best performance in the early period but quickly loses the potential through whole training steps as the other models takes advantage of the diverse exploration modes. The performance of 'Ref:Uniform random' is better than that of 'Ref:PPO'.

The exploration mode of Uniform random of our model and 'Ref:Uniform random' does not take place for a long time, but for a short time and gradually. Meanwhile, more exploration of 'Ref:PPO' occurs than that of 'Ref:Uniform random' according to a preset target rate ρ where the incessant exploration occurs after the starting mode.

After the starting mode[A], the PPO exploration mode of our model has about 3600 steps, which is more than the Uniform random exploration mode of our model, which is about 2100 steps. Most of the PPO exploration mode of our model occurs before 1M steps. The comparison of the total steps of the two exploration modes and exploitation of our model is as follows.

$$\pi_M^{TD3} >> \pi_M^{PPO} > \pi_M^{RND} \tag{8}$$

The guided exploration strategy produces the exploration mode of our model based on the modification of reward, Equation (5), and the ratio of success count S_O_m.

The second phase in Ant PUSH task starts from the steps when the reward occurs above -100 since the success rate, S_E, of the evaluation process occurs from about 500K and passes 0.6 at 1M steps. The situation of collapsed reward does not take place for a long time because of the loss modification, Equation (7), relying on the evaluation process.

2) Ant Fall: Our model shows a competitive performance against all other models after 3M steps. The preset of reward modification of Ant Fall is different from that of Ant Push , which means that $\alpha_{\text{on-policy}}$ is equal to $\alpha_{\text{off-policy}}$. The exploitation of our model occurs over 1M steps less than that of the two reference models, which is different from the situation of Ant Push. The performance of HIRO is stationary and decrease in the latter part. Unlike Ant Push, the performance of 'Ref:PPO' is better than that of 'Ref:Uniform random'.

The exploration mode of 'Ref:Uniform random' and our model takes place for longer than that of 'Ref:Uniform ran-

[1]Please read the section 3.1 of [17] for the experimental detail

dom' and our model in Ant Push. Meanwhile, the exploration mode of 'Ref:PPO' and that of 'Ref:Uniform random' are almost the same since a preset target rate ρ for 'Ref:Uniform random' and 'Ref:Uniform random' is the same.

Although $\alpha_{\text{on-policy}}$ is equal to $\alpha_{\text{off-policy}}$, since the ratio of success count regarding each action of the second level is also modified, after the starting mode, the total step comparison of two exploration mode and exploitation of our model is through whole training steps as follows.

$$\pi_M^{TD3} > \pi_M^{PPO} >> \pi_M^{RND} \tag{9}$$

Unlike the Ant PUSH task, the second phase in the Ant Fall task suffers a drop of reward between 4M and 4.5 M steps. The success rate of the evaluation process stays between 0.5 and 0.6 during the period. The recovery of reward quickly takes place due to the success rate compared with IV-B2.

B. Ablation study

We investigate our model without the reward modification, the loss modification and both modifications. Fig. 5 shows the results of the experiment compared with our normal model in the Ant Push task. Therefore, the part related to our normal model in Ant Push is removed for the purpose of experimenting each case.

1) Without the reward modification: While the exploitation has less steps than our normal model, the exploration of Uniform random and PPO has more steps than our normal model. The performance of reward and success rate slowly increase.

2) Without the loss modification: Again, the two exploration modes have more steps than our normal model and the exploitation has less steps than our normal model. It shows a drop of reward between 2.2M steps and 3M steps due to the increase in PPO exploration. Although the success rate is better than that of our normal model during the period, its performance is worse than that of our normal model.

3) Without both the reward modification and the loss modification: Too many explorations and less exploitation cause the worst performance.

V. Discussion

A. The effect of on-policy for exploration

When the on-policy operates in the beginning of exploration, the performance of on-policy, π_M^{PPO}, is not competitive. However, after it is trained by itself or other policies to some extent, the performance of on-policy shows better performance than the random policy. Meanwhile, In practice π_T^{PPO} is likely not to suffer a local minima due to the three policies of *Middle* level.

B. The effect of reward modification

In Ant Fall task, the performance of our model in the early steps up to about 2M steps lags behind all other models. The reason is that the on-policy operates for long time up to then since $\alpha_{\text{on-policy}}$ is equal to $\alpha_{\text{off-policy}}$. The reward modification for the guided exploration takes advantage of the fixed value of $\alpha_{g^{\text{expl-mode}}}$, which is not an adaptive strategy.

C. The effect of loss modification

The occurrence of on-policy and random policy in the Ant Fall task between 4M and 4.5M steps gives rise to a drop in performance of the agent. In particular, the modeling of uncertainty reflecting the success rate, S_E, can be considered. The higher S_E is, the lower the uncertainty is. Thus, S_E is related to the uncertainty.

VI. Conclusion

In order to overcome the issues of a non-monolithic exploration of [17], this paper introduces an autonomous non-monolithic agent with multi-mode exploration based on options framework. We reveal the potential of our model to follow a behaviour thought of humans and animals. Our model takes advantage of the difference in the degree of entropy of each exploration policy with a guidance-exploration framework. A robust optimal policy can be expected due to the evaluation process. The research on a guided exploration of the adaptive strategy for the multi-mode exploration of an autonomous non-monolithic agent is required. The further research on the modeling of S_E in the agent is also required for the robust optimal policy.

References

[1] G. Ostrovski, M. G. Bellemare, A. Oord, and R. Munos, "Count-based exploration with neural density models," in *International conference on machine learning*. PMLR, 2017, Conference Proceedings, pp. 2721–2730.

[2] O. Zhelo, J. Zhang, L. Tai, M. Liu, and W. Burgard, "Curiosity-driven exploration for mapless navigation with deep reinforcement learning," *arXiv preprint arXiv:1804.00456*, 2018.

[3] J. Fu, J. Co-Reyes, and S. Levine, "Ex2: Exploration with exemplar models for deep reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

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

[5] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. Xi Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel, "#Exploration: A study of count-based exploration for deep reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[6] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," *arXiv preprint arXiv:1808.04355*, 2018.

[7] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos, "Unifying count-based exploration and intrinsic motivation," *Advances in neural information processing systems*, vol. 29, 2016.

[8] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "Vime: Variational information maximizing exploration," *Advances in neural information processing systems*, vol. 29, 2016.

[9] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *arXiv preprint arXiv:1802.06070*, 2018.

[10] K. Gregor, D. J. Rezende, and D. Wierstra, "Variational intrinsic control," *arXiv preprint arXiv:1611.07507*, 2016.

[11] J. Achiam, H. Edwards, D. Amodei, and P. Abbeel, "Variational option discovery algorithms," *arXiv preprint arXiv:1807.10299*, 2018.

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

[13] N. Savinov, A. Raichuk, R. Marinier, D. Vincent, M. Pollefeys, T. Lillicrap, and S. Gelly, "Episodic curiosity through reachability," *arXiv preprint arXiv:1810.02274*, 2018.

[14] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune, "Go-explore: a new approach for hard-exploration problems," *arXiv preprint arXiv:1901.10995*, 2019.

[15] Y. Guo, J. Choi, M. Moczulski, S. Feng, S. Bengio, M. Norouzi, and H. Lee, "Memory based trajectory-conditioned policies for learning from sparse rewards," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4333–4345, 2020.
[16] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped dqn," *Advances in neural information processing systems*, vol. 29, 2016.
[17] M. Pislar, D. Szepesvari, G. Ostrovski, D. Borsa, and T. Schaul, "When should agents explore?" *arXiv preprint arXiv:2108.11811*, 2021.
[18] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.
[19] M. A. Wiering and H. Van Hasselt, "Ensemble algorithms in reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 930–936, 2008.
[20] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," *Advances in neural information processing systems*, vol. 31, 2018.
[21] S. Flennerhag, J. X. Wang, P. Sprechmann, F. Visin, A. Galashov, S. Kapturowski, D. L. Borsa, N. Heess, A. Barreto, and R. Pascanu, "Temporal difference uncertainties as a signal for exploration," *arXiv preprint arXiv:2010.02255*, 2020.
[22] J. Downar, A. P. Crawley, D. J. Mikulis, and K. D. Davis, "A cortical network sensitive to stimulus salience in a neutral behavioral context across multiple sensory modalities," *Journal of neurophysiology*, vol. 87, no. 1, pp. 615–620, 2002.
[23] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Empowerment: A universal agent-centric measure of control," in *2005 ieee congress on evolutionary computation*, vol. 1. IEEE, 2005, Conference Proceedings, pp. 128–135.
[24] A. L. Thomaz and C. Breazeal, "Experiments in socially guided exploration: Lessons learned in building robots that learn with and without human teachers," *Connection Science*, vol. 20, no. 2-3, pp. 91–110, 2008.
[25] E. Derman, D. Mankowitz, T. Mann, and S. Mannor, "A bayesian approach to robust reinforcement learning," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, Conference Proceedings, pp. 648–658.
[26] T. Moskovitz, J. Parker-Holder, A. Pacchiano, M. Arbel, and M. Jordan, "Tactical optimism and pessimism for deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 849–12 863, 2021.
[27] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018, Conference Proceedings.
[28] D. J. Mankowitz, T. A. Mann, and S. Mannor, "Adaptive skills adaptive partitions (asap)," *Advances in neural information processing systems*, vol. 29, 2016.
[29] Y. Jinnai, J. W. Park, D. Abel, and G. Konidaris, "Discovering options for exploration by minimizing cover time," in *International Conference on Machine Learning*. PMLR, 2019, Conference Proceedings, pp. 3130–3139.
[30] N. Bougie and R. Ichise, "Fast and slow curiosity for high-level exploration in reinforcement learning," *Applied Intelligence*, vol. 51, no. 2, pp. 1086–1107, 2021.
[31] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.
[32] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine, "Latent space policies for hierarchical reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1851–1860.
[33] O. Nachum, S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," *arXiv preprint arXiv:1805.08296*, 2018.

APPENDIX

Our research takes advantage of HIRO[2] for the implementation of an options framework. The reference code of HIRO is open source code[3]. In addition, PPO open source code is also adopted[4]. Finally, those two open programs are combined to implement our model. In the Algorithm[1], we represent only the main part of our algorithm which is implemented on the reference code.

In this implementation, we try to take advantage of the original parameter setting of reference codes, HIRO[3] and PPO[4]. The size of neural network of actor and critic for the exploration policy, π_T^{PPO}, and on-policy, π_M^{PPO}, are the same as the that of π_M^{TD3} and π_L^{TD3}. The size of neural network of π_M^{TD3} and π_L^{TD3} is the same as that of neural network which HIRO utilizes. All training parameters of π_M^{TD3} and π_L^{TD3} are used with the parameters defined on 'train_param_hiro.csv' file. The training parameters of the exploration policy, π_T^{PPO}, and on-policy, π_M^{PPO} are adopted with the same ones from PPO[4] reference code except for K_epochs which is changed from 80 to 10. The value of horizon of the exploration policy, π_T^{PPO} is 50 steps. Its training step is 400 steps. The value of horizon of π_M^{PPO} is the same as that of π_M^{TD3}. The training step of π_M^{PPO} is 30 steps. The K_epochs of π_M^{PPO} is also 10. The size of action and state are the same as those of Ant domain.

Ant Push and Ant Fall in our implementation are tested until 3.15M and 6M steps respectively. During the first 100K steps, *Middle* and *Low* levels are trained according to the intention of the original code. Then, during the next 250K steps, only random policy runs and all levels including *Top* level are trained. After only random policy runs, all modes which are random policy, on-policy and off-policy work. All logs start after the first 100K steps.

The target rates, ρ, regarding the reference models of [17] are as follows. In Ant Push, the ρ of Ref:PPO and Ref:Uniform random are 0.01 and 0.0001 respectively. In Ant Fall, the ρ of Ref:PPO and Ref:Uniform random are the same, at 0.001.

The following parameters are the parameters of our model. $S\_O_{g^{\text{expl-mode}}}$ of both Ant Push and Ant Fall in the starting mode is 0.9. Then, After starting mode, $S\_O_{g^{\text{expl-mode}}}$ is changed as follows. In Ant Push, $S\_O_{g^{\text{expl-mode}}}$ of all three sub-policies, π^{TD3}, π^{PPO} *and* π^{RND}, of *Middle* level is the same as 0.6. In Ant Fall, $S\_O_{g^{\text{expl-mode}}}$ of π^{TD3} and π^{PPO} is 0.6. $S\_O_{g^{\text{expl-mode}}}$ of π^{RND} is still 0.9. Meanwhile, In Ant Push, $\alpha_{g^{\text{expl-mode}}}$ of π^{TD3}, π^{PPO} *and* π^{RND} are 0, 0.4 and 0.7 respectively. In Ant Fall, $\alpha_{g^{\text{expl-mode}}}$ of π^{TD3}, π^{PPO} *and* π^{RND} are -0.2, -0.2 and 0.7 respectively.

Furthermore, the sign of success rate, S_E, is positive for π^{PPO} and π^{RND} and negative for π^{TD3} in both tasks. The purpose of S_E is to increase or reduce the loss of π_T^{PPO} regarding the original loss value.

The functions used in Algorithm[1] are as follows. $Clamp\_Max(...)$ is the function of clamping the action of *Low* level into the max_value range. $Train_T(..., S\_E, g^{\text{expl-mode}}, ...)$ is the train function of *Top* level. $Train\_\pi_M^{TD3}(...)$ is the train function of π_M^{TD3} of *Middle* level. $Train\_\pi_M^{PPO}(...)$ is the train function of π_M^{PPO} of *Middle* level. The evaluation function of π^{TD3} of *Middle* level is $Evaluate\_\pi_M^{TD3}(...)$. $Judge\_success(...)$ is the judge function regarding the success of π^{TD3} of *Middle* level. The

[2] [33] names the devised model HIRO.
[3] https://github.com/ziangqin-stu/rl_hiro
[4] https://github.com/nikhilbarhate99/PPO-PyTorch

higher-level context, $target\_pos$, is used in common on the policies of *Top* and *Middle*.

The model used in the ablation study is the same as the parameters of the model used in Ant Push. The condition of each case is applied to implement it.

Algorithm 1 Multi-exploration mode based on options framework

Initialize:
  Set the value of $\alpha_{g^{expl\text{-}mode}}$ *according to* $g^{expl\text{-}mode}$ *of* π_T^{PPO}
  Set the value of $S\_O_{g^{expl\text{-}mode}}$ *according to* $g^{expl\text{-}mode}$ *of* π_T^{PPO}
procedure *Evaluate*_$\pi_M^{TD3}(..., \pi_M^{TD3}, \pi_L^{TD3}, ...)$
  According to π_M^{TD3} *and* π_L^{TD3},
    compute S_E
end procedure
procedure $Train_T(..., S\_E, g^{\text{expl-mode}}, ...)$
  if $g^{\text{expl-mode}}$ *is Random uniform or PPO* **then**
    $loss_{final} = loss + S\_E * loss$
  else
    $loss_{final} = loss - S\_E * loss$
  end if
end procedure
for $t = 0, \ldots, T-1$ **do**
  $action \leftarrow Clamp\_Max(\pi_L^{TD3}(state, goal) + Noise)$
  Execute action, observe R and next_state
  $Done \leftarrow Judge\_success(state, target\_pos)$
  Increase Done_m by 1 if Done is True
  Increase R_m by R
  Increase Count_m by 1

  if *the horizon of Top level* **then**
    $S\_O\_m \leftarrow Done\_m \,/\, Count\_m$;
    $R_{final} \leftarrow R\_m + \alpha_{g^{\text{expl-mode}}} * R\_m$
    if $S\_O\_m >= S\_O_{g^{\text{expl-mode}}}$ **then**
      $Done\_m \leftarrow True$
    end if
    if *the training time of Top level* **then**
      $Train_T(..., S\_E, g^{\text{expl-mode}}, ...)$
    end if
    if *the starting mode* **then**
      $g^{\text{expl-mode}} \leftarrow Random\ policy$
    else
      $g^{\text{expl-mode}} \sim \pi_T^{PPO}$
    end if
    $R\_m, Count\_m, Done\_m \leftarrow 0$
  end if

  if *the horizon of Middle level* **then**
    According to $g^{expl\text{-}mode}$,
      $next\_goal \sim \pi_M^{RND}, \pi_M^{PPO}$ *or* π_M^{TD3}
    if *the training step of* π_M^{PPO} **then**
      $Train\_\pi_M^{PPO}(...)$
    end if
  end if

  $state \leftarrow next\_state$
  $goal \leftarrow next\_goal$

  if *the training step of* π_M^{TD3} **then**
    $Train\_\pi_M^{TD3}(...)$
  end if

  if *the evaluation step of Middle level* **then**
    $Evaluate\_\pi_M^{TD3}(..., \pi_M^{TD3}, \pi_M^{TD3}, ...)$
  end if

end for