# MaMiC: Macro and Micro Curriculum for Robotic Reinforcement Learning

Extended Abstract

Manan Tomar[1], Akhil Sathuluri[1], Balaraman Ravindran[1,2]
[1]Indian Institute of Technology Madras, Chennai, India
[2]Robert Bosch Center for Data Science and AI (RBCDSAI), Chennai, India
manan.tomar@gmail.com, akhilsathuluri@gmail.com, ravi@cse.iitm.ac.in

## ABSTRACT

Shaping in humans and animals has been shown to be a powerful tool for learning complex tasks as compared to learning in a randomized fashion. This makes the problem less complex and enables one to solve the easier sub task at hand first. Generating a curriculum for such guided learning involves subjecting the agent to easier goals first, and then gradually increasing their difficulty. This paper takes a similar direction and proposes a dual curriculum scheme for solving robotic manipulation tasks with sparse rewards, called MaMiC. It includes a macro curriculum scheme which divides the task into multiple sub-tasks followed by a micro curriculum scheme which enables the agent to learn between such discovered sub-tasks. We show how combining macro and micro curriculum strategies help in overcoming major exploratory constraints considered in robot manipulation tasks without having to engineer any complex rewards. The performance of such a dual curriculum scheme is analyzed on the Fetch environments.

**ACM Reference Format:**
Manan Tomar, Akhil Sathuluri, Balaraman Ravindran. 2019. MaMiC: Macro and Micro Curriculum for Robotic Reinforcement Learning. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, IFAAMAS, 3 pages.

## 1 INTRODUCTION

In order to solve complex robotic manipulation tasks it is important that we learn in an organized, meaningful manner rather than learning using data collected in a random fashion. Curriculum learning [1], [6] is a powerful concept that allows us to come up with such training strategies. Recently, curriculum learning has been used to solve complex robotic tasks (not necessarily manipulation) such as in [2], [5]. However, these approaches make the assumption that the agent can be reset to any desired state, and also make use of expert state action trajectories [5], which are expensive to generate. Unlike such techniques, our method is not restricted by the ability to reset. Moreover, we use state-only demonstration sequences for learning only in specific tasks, and do not use demonstrations at all for the other tasks, thus distinguishing our work from those in the imitation learning solution sphere.

One way of looking at the problem in hand is to extract sub-goals for a given task, learn sub-policies or skills that achieve these sub-goals, and then execute them in the right order. Such a top-down approach allows exploiting the structure of the problem, since the extracted sub-goals define the nature of the solution. Moreover, we also focus on the sequential nature of the problem, i.e. solving to achieve the first sub-goal, then the second sub-goal and so on. This is important as most robotic locomotion or manipulation problems can be recognized in this manner. In our method, the sub-goal extraction and sequencing is managed by the macro scheme, while learning each sub-policy is managed by the micro scheme. In order to achieve this, both of these methods exhibit and use concepts from curriculum learning.

## 2 MICRO CURRICULUM

A micro curriculum tries to alleviate the assumption of being able to start some trajectories from favorable states. We believe that starting at a particular state should be based on the environment's choice but not the agent's. We propose replacing all or some transition sample goals with the `micro goals` (refers to goals generated by the goal generator) which may be generated by any generative modeling technique. Using an off policy RL algorithm allows us to replace sampled transition goals from the buffer with `micro goals`. The goals are generated such that they are initially close to the achieved states at the end of each trajectory (i.e. the `achieved goal` distribution) and slowly shift to being closer to the actual or `desired goal` distribution of the task in hand. Since this procedure involves learning a mapping between goals and actions, eventually the agent is able to generalize well for the actual goal distribution. We relate this with curriculum learning because the agent initially learns for a goal distribution much simpler to learn i.e. the `achieved goal` distribution and then continues learning for increasingly difficult goals, leveraging the previously learned skills.

To train the goal generator, we make use of Generative Adversarial Networks or GANs [3] and modify the formulation used by [4]. We incorporate an additional parameter $\alpha \epsilon [0, 1]$ which governs the resemblance of the generated distribution to the `achieved goal` distribution and the actual or `desired goal` distribution. $\alpha = 0$ forces the generator to produce goals similar to the currently achieved states, while $\alpha = 1$ produces goals similar to the actual distribution. The exact objective function is given below.

$$min_D V(D) = \mathbf{E}_{g\sim p_{data}(g)}[(1-\alpha)\,(D(g_{achieved})-1)^2 + \alpha\,(D(g_{desired})-1)^2] \;+\; \mathbf{E}_{z\sim p_z(z)}[D(G(z))^2] \qquad (1)$$

*Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.* © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

$$min_G\ V(G)\ = \mathbf{E}_{z \sim p_z(z)}[(D(G(z))\ - 1)^2] \qquad (2)$$

,where $D$ denotes the discriminator network, $G$ the generator network, and $V$ the GAN value function. $p_z$ here is taken as a uniform distribution between 0 and 1 from which the noise vector $z$ is sampled. In all experiments that follow, we choose to update $\alpha$ if the success rate of the currently learned policy for goals generated by the GAN lies above a particular threshold consistently for a few epochs. This essentially tells us that the policy has now mastered achieving the currently generated goals with some degree of confidence and thus the GAN can now shift further towards producing goals resembling the desired distribution.

At each iteration of the micro algorithm, the goal generator produces a `micro goal` which is used to condition the behavior policy and collect samples by executing it. For each episode, the end of trajectory state, called as the `achieved goal` is collected and stored in memory. While training, a mini batch of data is sampled and some or all of the goal samples are relabelled with new ones using the goal sampling strategy (described below). The `achieved goals` and the `desired goals` are used to update the goal generator periodically. The `desired goals` essentially either are the goals corresponding to the task in hand or any of the `sub-goals` provided by the sub-goal extraction method. Therefore, this allows the micro scheme to be run independently as well as in combination with the macro method. For replacing goals by sampling new ones, we consider different strategies such as having a mixture of `HER goals` (these refer to achieved states in a trajectory while following the currently learned policy, randomly sampled as is proposed by HER) and `micro goals` (referred to as micro-g), and having a mixture of `HER goals` and `desired goals` (referred to as micro-sg).

## 2.1 Micro - Tasks Considered

We consider harder variants of the pushing and sliding tasks for testing the micro scheme. These tasks are "made hard" by ensuring that the object and the target do not lie in similar distributions initially and are far apart from each other for all episode samples. We also consider the pick and place task, which requires an object to be picked and placed at a target site. The target is never sampled on the table and always in the air. We also do not start any episode with the block already in the robot's gripper, thus making sure that favorable starts are not considered. We are able to learn optimal policies for all three tasks, while HER fails completely.

# 3 MACRO CURRICULUM

A macro curriculum scheme allows extracting sub-goals by leveraging demonstrated states or observations and sequentially learning the sub-policies for each sub-goal. In the experiments we consider, this implies that learning to achieve the second sub-goal is facilitated by leveraging previous learning of achieving the first sub-goal (learning to push uses already gathered information about learning to reach). We argue that this setting is general enough because each sub-policy itself learns a hard task (the task of reaching) instead of simple "macro" actions (moving the manipulator continuously in a particular direction). This allows representing the final task policy as comprising each sub-policy. Specifically, we consider long horizon tasks and assume that few demonstration state trajectories $\tau = s_0, s_1, ...s_t$ are available for the given tasks. In general, detecting changes in state representation has been shown to be a good method for extracting sub-goals. This is since system dynamics change suddenly around such sub-goals. In our case, the dense reward $r_{dense}\ = ||\ g_{achieved} - g_{desired}\ ||^2$ computed per time step for a demonstration is used as the signal for sub-goal extraction. We compute the gradient ratio for such a signal and choose the sub-goal as the state for which consistent spikes are observed. The intuition for finding a good sub-goal in a typical manipulation task is to observe that there is a sudden change in the dynamics when the system starts interacting with the object. Learning between two such `sub-goals` can be performed by following a micro curriculum scheme detailed above.

## 3.1 Macro - Tasks Considered

We introduce a new task setting called Receptor-PickandPlace which comprises an object placed on a table, a receptor site on the table, and a target located in the air. The agent is required to pick and place the object at a target, which gets activated only if the object first passes through the receptor site. Therefore, the agent is not rewarded even if the object is successfully placed at the target, if it does not pass from the receptor site initially. Such a task becomes extremely difficult to solve because of a sequencing behavior involved and a sparse reward available. We show how combining the macro and micro schemes can solve this task, by 1) leveraging demonstration states to extract a sub-goal near the receptor site and 2) using a powerful micro scheme to realize the sequencing of tasks involved, i.e. first moving the block to the receptor and then to the target.

For the Receptor-PickandPlace task, recognizing the receptor as a sub-goal is crucial to learning. There is a significant peak in the dense reward gradient ratio around the receptor location, proving that the sub-goal extraction in the macro scheme is able to leverage demonstrations efficiently. This when combined with a micro scheme is able to learn the sequence of going to the receptor first with the block, thus activating the target, followed by placing it over the target. Median success rates for all tasks are shown in the table below.

| Task | Micro-sg | Micro-g | HER | MaMiC |
|---|---|---|---|---|
| Push-hard | 100% | 92% | 1% | - |
| Slide-hard | 42% | 31% | 1% | - |
| PickAndPlace | 98% | 95% | 0% | - |
| Receptor-PickPlace | 2% | 1% | 0% | 98% |

# 4 FUTURE WORK

The next challenge is to show how such a technique performs on even more longer horizon tasks, perhaps involving multiple objects as well. Working with image based observations can allow for learning richer representations useful in sub-goal extraction. Moreover, collecting state or observation demonstration trajectories is relatively simpler and more intuitive with images. Considering better heuristics for how $\alpha$ is updated to produce goals closer to the `DesiredGoal` distribution is an important point to improve upon. Another avenue for future work is to incorporate different schemes of sub-goal extraction which exploit domain specific properties.

## REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 41–48.

[2] Carlos Florensa, David Held, Markus Wulfmeier, and Pieter Abbeel. 2017. Reverse curriculum generation for reinforcement learning. *arXiv preprint arXiv:1707.05300* (2017).

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[4] David Held, Xinyang Geng, Carlos Florensa, and Pieter Abbeel. 2017. Automatic goal generation for reinforcement learning agents. *arXiv preprint arXiv:1705.06366* (2017).

[5] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2017. Overcoming exploration in reinforcement learning with demonstrations. *arXiv preprint arXiv:1709.10089* (2017).

[6] Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. 2017. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407* (2017).