Agency for Science, Technology and Research
SINGAPORE

Efficient Robotic Task Generalization Using Deep Model Fusion Reinforcement Learning

Wang Tianying, Hao Zhang, Wei Qi Toh, Hongyuan Zhu, Cheston Tan, Yan Wu, Yong Liu, Wei Jing

Research Engineer in Agency for Science, Technology and Research (A*STAR), Singapore

Table of contents

1.

Introduction

“

AI is the new electricity.

How to automate a robot?

Traditional Robot Programming

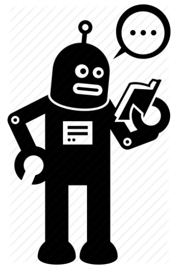
Manually programme a robot with instructions for a specific task.

- ▷ Manual and costly
- ▷ Limited to specific tasks
- ▷ Lack of adaptability

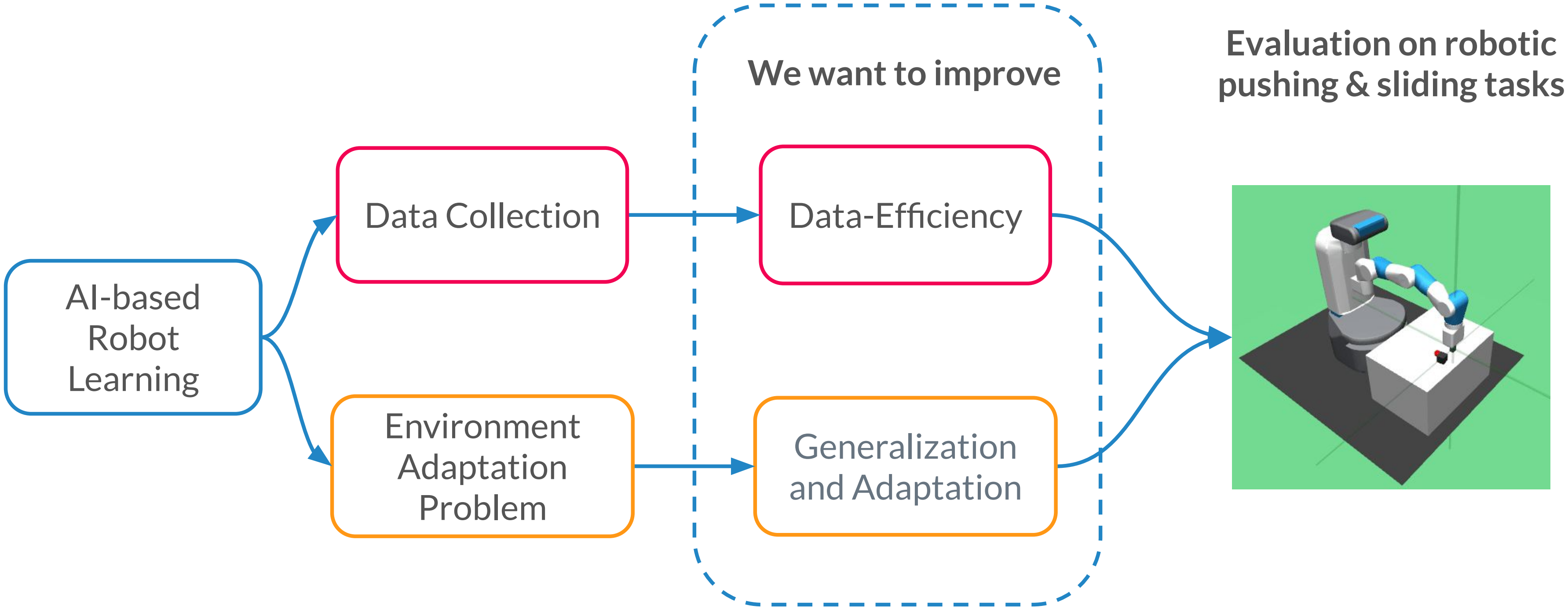
AI-based Robot Learning

Allows a robot to acquire novel skills or adapt to its environment through learning algorithms, e.g. Reinforcement Learning.

- ▷ Less manual programming
- ▷ General and robust to handle the same category of tasks



What is our motivation?

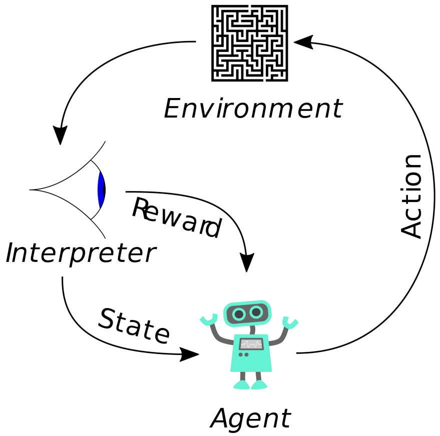


2.

Related Work

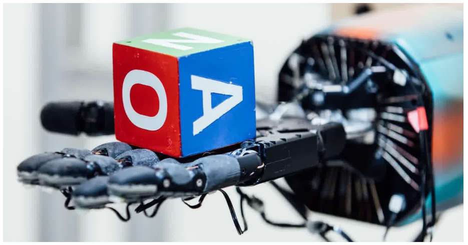
Reinforcement Learning

- Algorithm learns a **policy** of how to act in a given environment
- Every **action** has some impact in the **environment**, and the environment provides **reward** that guides the learning algorithm



Common Applications

- AlphaGo
- Atari Games (Value-based algorithms)
- Robotics (Policy-based algorithms)



How to improve data-efficiency and generalization?

- Reinforcement Learning (RL) usually requires **a large amount of data** to train the model.
- The performance of the previously trained model would drop with **different environment.**

Hindsight Experience Replay (HER)

Deep Deterministic Policy Gradient (DDPG)

Deep Model Fusion Reinforcement Learning (DMF-RL)

Robotic Reinforcement Learning

3.

Proposed Method

Problem Formulation

Basic reinforcement learning is modeled as a Markov decision process (MDP)

MDP could be represented as a tuple (S, A, T, r, λ), where

S is the state space

A is the action space

$T : S \times A \Rightarrow S$ is the state transition model

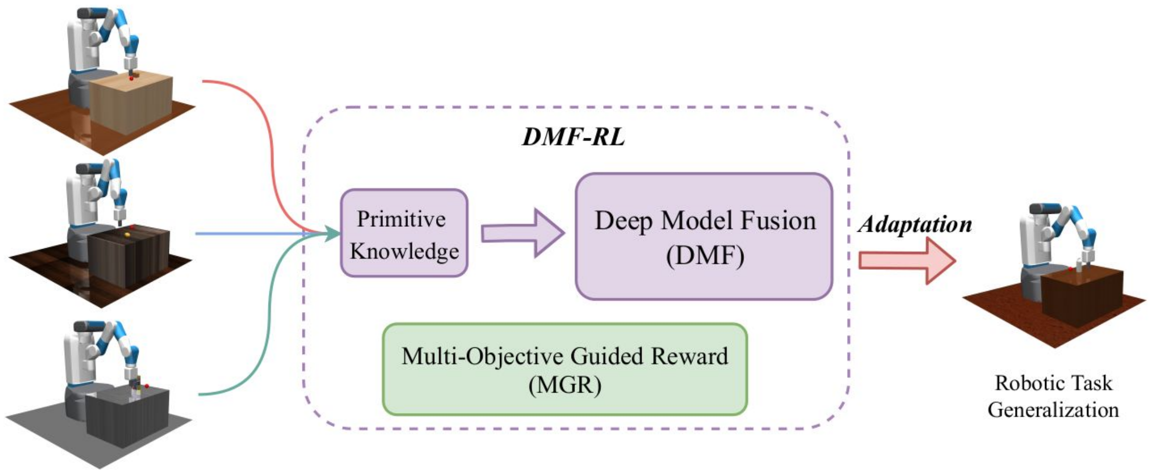
$r : S \times A \Rightarrow r \in R$ is the rewards by taking an action at a certain state

$\lambda \in [0, 1]$ is the discount factor

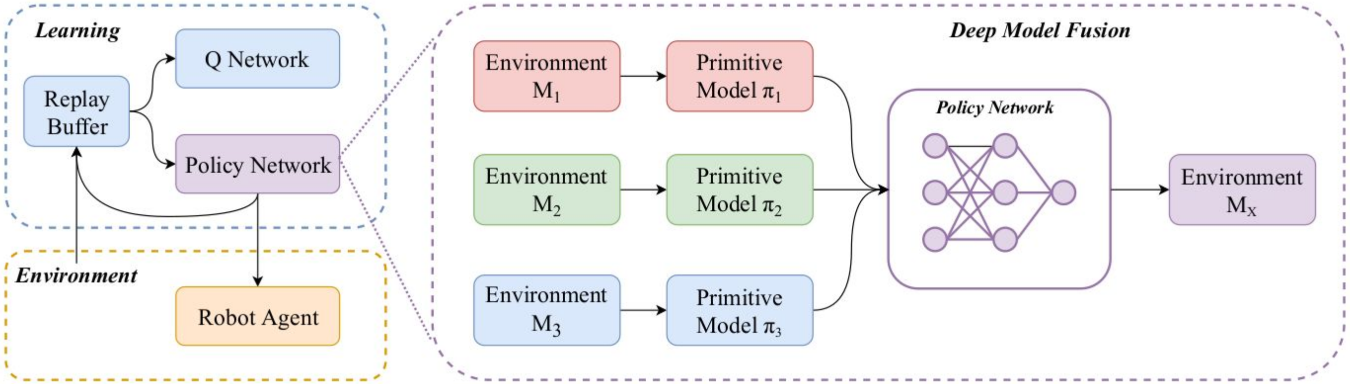
Deep Model Fusion Reinforcement Learning (DMF-RL)

The main components of our systems are

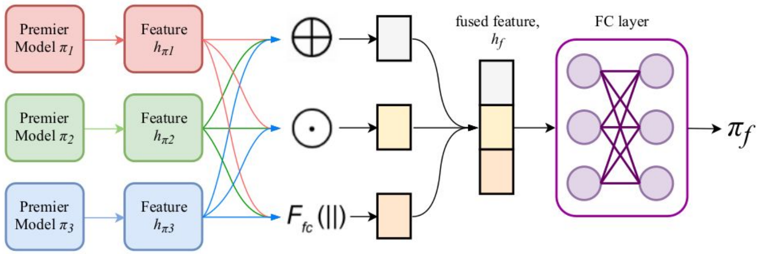
- a **Deep Model Fusion (DMF)** method to store and combine the previous trained knowledge, which speeds up the training process for task generalization; and
- a **Multi-objective, Guided Rewards (MGR)** system that converts the sparse rewards of typical RL problem to a multi-objective dense rewards system.



Deep Model Fusion (DMF)



Deep Model Fusion Reinforcement Learning Architecture



The Architecture of Policy Network with Deep Model Fusion

Multi-objective Guided Reward (MGR)

The general MGR is formulated as:

$$r = \alpha_1 G_f + \sum_{i=2}^{n} \alpha_i O_i + \alpha_{n+1} O_p$$

The MGR for robot pushing task is formulated as:

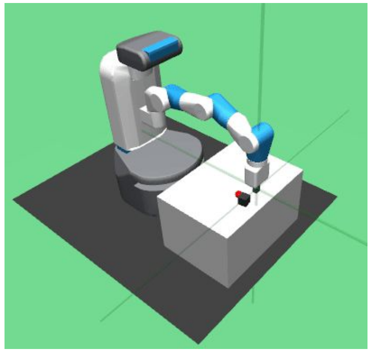
$$\begin{aligned} r(d_{og}, d_{oe}, d_{es}) = &\alpha_1(- \|d_{og} > \eta\|) + \alpha_2(-d_{oe}) + \alpha_3(-d_{og}) \\ &+ (\|d_{es} < \mu\|)(\log d_{es} - \log \mu) \end{aligned}$$

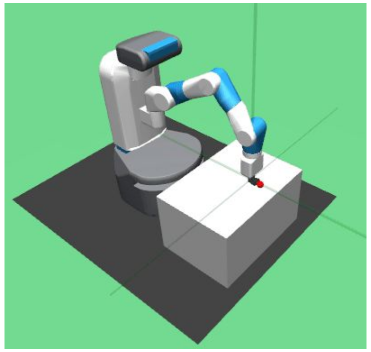
4.
Experiment & Discussion

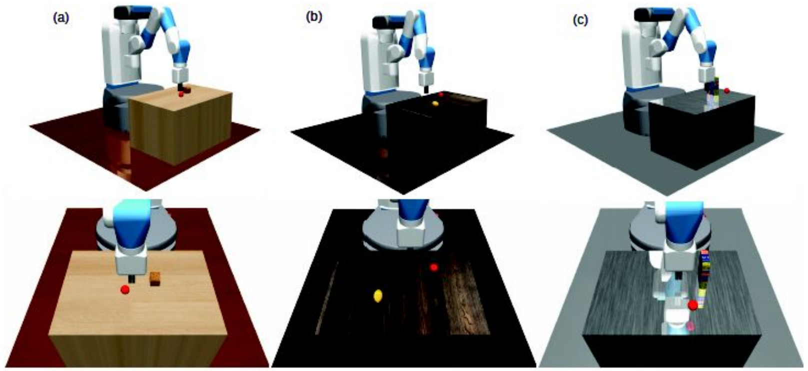
Experiment Set-up

OpenAI Gym Mujoco Simulation

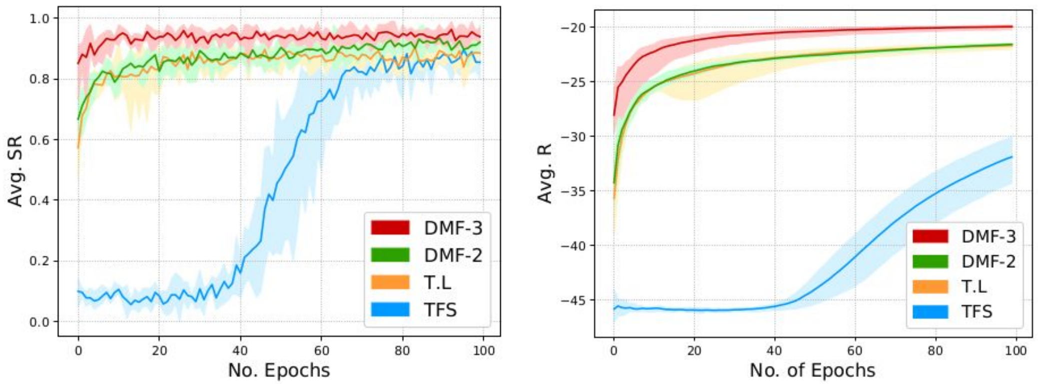
- Pushing and Sliding Tasks
- Various environment with different surface and different object shapes







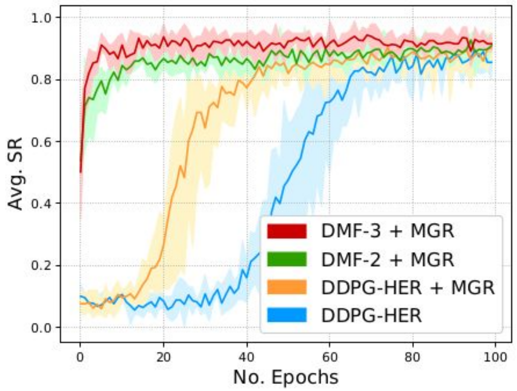
Results and Discussion



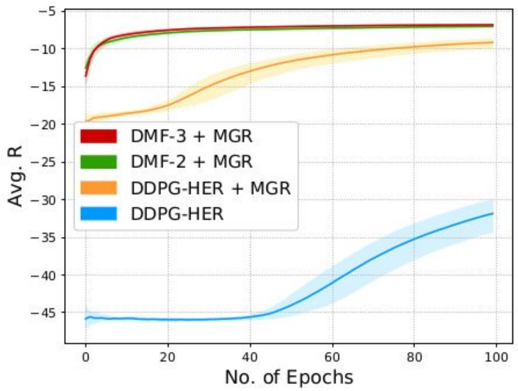
(a) Average success rates (b) Average returns

- The above figure shows the training results in the robot pushing application.
- Compared to training from scratch (TFS) and transfer learning (T.L.), the robot agent learned faster with our methods, DMF method that combined two models (DMF-2) and three models (DMF-3).
- The robot agent with primitive knowledge from our method demonstrated the good capability of task generalization and environment adaptation.

Results and Discussion



(a) Average success rates



(b) Average returns

- The MGR was also evaluated with different environment settings.
- The above figure shows the results comparison of MGR and DMF+MGR methods in robot pushing application.
- Compared to baseline algorithms (DDPG with HER), the agent learned faster with the proposed MGR system.

Results and Discussion

TABLE I: Success rate comparison of different methods

		DDPG-HER				**DDPG-HER + MGR**				**DMF-2 + MGR**				**DMF-3 + MGR**			
		50	100	150	200	50	100	150	200	50	100	150	200	50	100	150	200
Push	env-1	0.141	0.458	0.597	0.667	0.608	0.783	0.844	0.875	0.816	0.887	0.913	0.926	0.951	0.962	0.964	0.966
	env-2	0.175	0.518	0.648	0.718	0.445	0.657	0.733	0.773	0.828	0.859	0.875	0.885	0.868	0.892	0.9	0.904
	env-3	0.074	0.079	0.106	0.157	0.089	0.216	0.387	0.507	0.839	0.875	0.890	0.897	0.906	0.913	0.917	0.92
Sliding	env-1	0.206	0.342	0.411	0.451	0.342	0.513	0.597	0.645	0.424	0.576	0.647	0.679	0.662	0.726	0.755	0.774
	env-2	0.098	0.158	0.224	0.291	0.183	0.296	0.376	0.425	0.302	0.509	0.606	0.659	0.618	0.725	0.766	0.79
	env-3	0.089	0.13	0.147	0.159	0.323	0.512	0.588	0.632	0.472	0.612	0.672	0.708	0.677	0.738	0.764	0.766

- The above table shows the comparison of the results with different methods at different training stages.
- Our method demonstrated its effectiveness among different applications in various training stages.
- The success rate of the proposed method was consistently higher than other methods in different environments and different training stages.

5.

Conclusion

Conclusion

- **Deep Model Fusion (DMF)** method to store and combine the previous trained knowledge

- **Multi-objective, Guided Rewards (MGR)** system that solves the sparse rewards of typical RL problem.

- The experiments results showed that DMF-RL can speed up the training process for **task generalization**

“

We can build a much brighter future where humans are relieved of menial work using AI capabilities.