An Optimal Re-parametrization Scheme for Generalization in Reinforcement Learning

Dzeuban Fenyom Ivan
Department of Artificial Intelligence
Kyungpook National University,
Daegu, Republic of Korea
ivanfenyom@gmail.com

Oladayo S. Ajani
Department of Artificial Intelligence
Kyungpook National University,
Daegu, Republic of Korea
oladayosolomon@gmail.com

Rammohan Mallipeddi
Department of Artificial Intelligence
Kyungpook National University,
Daegu, Republic of Korea
mallipeddi.ram@gmail.com

***Abstract*—The use of Reinforcement learning (RL) in addressing a variety of engineering tasks has seen a lot of increase lately. However, because RL policies are trained under specific environmental conditions, their performance degrades when deployed in environments with different environmental conditions. As a result, several researchers have focused on developing schemes aimed at minimizing this performance degradation and consequently realizing RL agents that are capable of generalizing to different environmental conditions. In this paper, we propose an optimal re-parameterization scheme based on weighted averaging to facilitate generalization in RL. In the proposed method, two RL agents are trained on simulated environments with different environmental or model parameters, and then covariance matrix adaptation evolution strategies (CMA-ES) is used to determine the optimal averaging weights to combine the two agents for a test environment. We evaluate the performance of our method on a set of popular RL locomotion environments and show that it can significantly improve the generalization performance of RL policies.**

***Index Terms*—Sim-to-real Transfer, Reinforcement Learning, Model Averaging, Covariance Matrix Adaptation Evolution Strategies.**

I. Introduction

Reinforcement learning has seen an increase in popularity in recent years thanks to its ability to solve several decision and control tasks in domains such as classical control [1], [2], games [3]–[5], robotics [6] and transportation [7], [8] etc. Generally, RL algorithms are aimed at finding policies that maximize a measure of goodness or reward (formulated according to the associated task) by interacting with a dynamic environment. Specifically, the RL learning cycle involves an agent or policy taking an action a_t at time t based on the state of the environment s_t and the associated reward r_t. Consequently, the environment transitions to a new state s_{t+1} and the process is repeated until a termination criterion is reached [2].

Although satisfactory performance of RL agents or policies have been demonstrated on environments with the same parameters as those they were trained on, these RL policies tend to overfit to that particular environment leading to performance degradation when deployed in environments with slightly different parameters or conditions [9]. This limitation severely restricts the practical deployment of RL in real-world settings where generalization is very important due to the highly dynamic nature of the environment. A common technique proposed to improve generalization in RL is domain randomization [10], [11] where agents are trained on environments with random variations in dynamics, visuals, etc. By exposing the agent to varied conditions during training, the goal is to learn more robust policies that generalize to different environmental conditions. However, naively randomizing the training domain often fails to produce sufficient generalization gains. Furthermore, domain randomization strategies require extensive hyperparameter tuning and can lead to unstable training or make tasks excessively difficult.

979-8-3503-1458-8/23/$31.00 ©2023 IEEE

In this work, we propose an optimal re-parameterization scheme based on weighted averaging to facilitate generalization in deep RL. Using CARL [12], a framework containing a collection of well-known RL environments extended to contextual RL problems and specifically designed to evaluate generalization in RL, we train a couple of RL agents on environments with very different environmental parameters and then use covariance matrix adaptation evolution strategy (CMAES) to determine the optimal re-parameterization weights to combine these policies and create a new policy with better generalization abilities. To evaluate the performance of our method, experiments were performed on two popular RL locomotion tasks namely, Ant and Half cheetah [12]. For training and testing, we generate three variants for each of the environments with different environmental conditions, by varying parameters such as parameters such as friction, and Torso mass of the base environments. Consequently, two RL agents based on PPO [13] were trained on two of the environments, and consequently, a new agent is evolved through the proposed optimal reparameterization scheme. In the testing phase, all three models were evaluated on all three variants for each of the RL tasks. Our experiments showed that the policy based on the proposed scheme outperformed all the base models on all of the environments.

The remainder of this paper is organized as follows. In section II, a background on CMAES, weighted averaging, and carl is provided. The proposed method is detailed in section III while section IV presents the results and discussions. Finally, the conclusions of this work are presented in Section V.

II. Background

A. CMA-ES

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a state-of-the-art stochastic optimization algorithm developed to solve non-linear, and non-convex optimization problems [14], [15]. CMA-ES belongs to the broader family of Evolution Strategies (ES), which draw inspiration from the principles of natural evolution for optimization [2], [16], [17]. CMA-ES is designed to optimize continuous, real-valued objective functions by iteratively exploring and adapting the search space. The algorithm operates in a manner similar to a natural evolutionary process, where candidate solutions (referred to as ”individuals” or ”parents”) are iteratively improved over generations. The key components of CMA-ES include:

- **Population Generation:** CMA-ES maintains a population of candidate solutions, represented as real valued vectors in the search space. These vectors are often referred to as ”individuals” or ”parents,” and they represent potential solutions to the optimization problem. Generally, CMA-ES employs a multivariate normal distribution to generate new candidate solutions. The distribution is characterized by a mean vector (representing the current best solution) and a positive-definite covariance matrix (representing the exploration directions in the search space).
- **Selection and Evaluation:** During each iteration, the individuals in the population are evaluated based on their fitness, which corresponds to the value of the objective function at their respective locations in the search space. The fitter individuals (usually known as elites) are selected to form the parent population for the next generation.
- **Adaptation of Covariance Matrix:** The crucial feature of CMA-ES is its ability to adapt the covariance matrix during the optimization process. The covariance matrix is updated based on the covariance of the selected parent population. This adaptation allows the algorithm to adjust the exploration of the search space dynamically, effectively dealing with ill-conditioned and anisotropic objective functions.
- **Mean Update:** The mean vector of the multivariate normal distribution is updated to be closer to the promising solutions found by the selected parents. This mechanism biases the search towards more promising regions of the search space.

These entire processes are repeated until a predefined termination criteria (usually the maximum number of function evaluations or function vale tolerance) is reached.

B. Contextual RL

Contextual Reinforcement Learning (cRL) addresses the issue of generalization in the RL setting by introducing distributions over multiple characteristics and properties of the environment [12]. Consider an agent learning a policy in an RL setting to pick a ball and put it in a cup. Generalization of this policy would involve assessing whether the agent can successfully pick up balls of different sizes, shapes, or even if it can put the object into other objects with similar physical properties to cups. The concept of cRL formalizes this by defining varying factors, such as the size of the cup’s handle or the height of the cup, as contexts that are sampled from a distribution. Each context then creates a separate Markov Decision Process (MDP), referred to as contextual MDPs (cMDPs), which are essentially variations of the same underlying MDP, differentiated by the changing contexts.

C. Model Averaging

Model averaging is a prominent ensemble technique within the realm of machine learning, where the combined expertise of multiple models, each of identical structures but trained under varying conditions, is harnessed to create a new model of superior performance [18]. The fundamental premise of this approach revolves around the notion of capitalizing on the diversifying effects induced by the disparate training conditions, ultimately yielding a composite model that surpasses the individual constituent models in its capacity for generalization [19].

The essence of model averaging lies in the fusion of the knowledge garnered from distinct models, all possessing analogous architectures but potentially divergent learned weights due to divergent training circumstances. By calculating the arithmetic mean of the weights of the participating models, the resulting model inherits a more encompassing representation of the underlying data distribution, effectively leveraging the complementary strengths of its constituent counterparts.

III. Proposed CMA-ES based model averaging scheme

The primary objective of the proposed approach is to enhance the generalization capabilities of Reinforcement Learning (RL) policies. To achieve this, we employ insights derived from policies trained in contextual MDPs and leverage the concept of model averaging to re-parameterize a new policy. By doing so, we aim to create a more robust and adaptable policy that can perform well in a wider range of situations.

The process involved in this approach can be delineated into two main stages, each serving a crucial purpose in achieving our overarching goal. The first stage is the training stage, where we subject RL policies to train in environments with different challenging and diverse parameters (contextual MDPs). The challenging environments parameters are those that affect generalization performance of vanilla RL algorithms [20]. By exposing the policies to these extreme scenarios, they are forced to adapt and learn intricate strategies to succeed in each setting. This diversity in training environments fosters the acquisition of specialized knowledge and skills by the individual policies.

The second stage is the reparametrization stage, where we employ weighted averaging to evolve a new policy based on the policy parameters RL agents from stage 1. Specifically, this weighted averaging scheme helps to consolidate the knowledge accumulated by the various policies during the training phase. However the performance of the resulting model from this stage depends on efficient selection of the averaging weights. Therefore to find those weights, we formulate the reparamterization task as an optimization problem where the decision variables are the weights and the objective function is the test contextual MDP. In order to solve the optimization problem, we employ CMA-ES. This process results in the creation of a collective model that combines the strengths and expertise of each individual policy, thereby yielding a more comprehensive and versatile representation of the RL landscape.

$$\pi_{proposed} = w_i\pi_i + w_{i+1}\pi_{i+1}...w_{i+N-1}\pi_{i+N-1} \qquad (1)$$

where π_i refers to the i_{th} policy, w_i refers to the weigth assigned to the i_{th} policy and N refers to the number of training contextual MDPs. In the context of this work, we set $N = 2$.

IV. Experimental Study

This section details the experiments conducted in this study to evaluate the performance of the proposed optimal reparameterization scheme. First, we highlight the RL environments (MDPs) employed and their characteristics. Second, we describe the experimental setup and evaluation metrics. Finally the results from the experiments are presented and discussed.

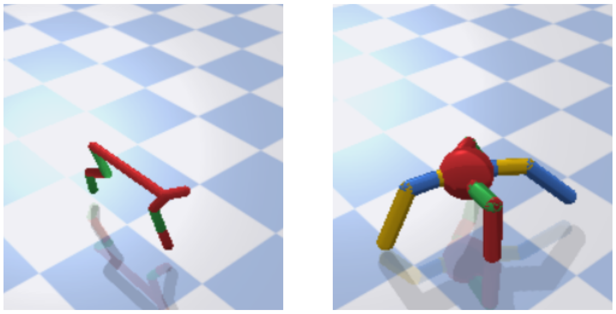
A. *RL locomotion environments*

Two RL locomotion environments were studied, Half Cheetah and Ant. To create the contextual MDPs, the friction and the torso masses were varied for both environments as well as the actuator strength in the case of Half Cheetah.

- **Half Cheetah**: Half cheetah is a locomotion environment where a two-dimensional entity comprising of 9 segments (a fixed head and torso, front and back paws, front and back thighs, front and back feet) connected by 8 joints aims to apply torque on its joints to move in a forward direction at the maximum speed. A positive reward is given for forward movement distance while a negative reward penalized backward motion. A graphical illustration of the Half Cheetah locomotion task is shown in Fig. 1a.
- **Ant**: Ant is a three-dimensional robot entity comprised of a central torso with four two-segment legs attached to it whose goal is to move in a forward direction by applying torque on its joints. A positive reward is given for forward movement distance while a negative reward penalized backward motion. A graphical illustration of the Ant locomotion task is shown in Fig. 1b.

B. *Experimental setup and evaluation metrics*

The two aforementioned locomotion environments were formalized based on OpenAI's gym framework [21] using the Mujoco physics engine [22]. The contextualization of the MDPs were performed according to the CARL [12] python's framework.



(a) Half Cheetah

(b) Ant

Fig. 1: Graphical illustration of the locomotion environments

The variations in the context for the MDPs were inspired by [20] where they found environmental parameters ranges which are challenging and impossible for agents trained on default environmental parameters to solve. The variations applied to realize the contextual MDPs were as follow:

- **Friction**: For both Half cheetah and Ant, the training contexts' friction was selected randomly in the range $[0.2, 0.5] \cup [1.1, 1.4]$ which is shown to be a challenging range of friction value for half cheetah [20] and the testing context's was selected at random in the range $[0.5, 1.1]$ which is a range which can be solved by an agent trained on the default parameters.
- **Torso Mass**: Similarly, the torso masses were also selected randomly within challenging range($[5, 7.5] \cup [12.5, 15]$) and the testing context was set in a less challenging range($[7.5, 12.5]$) for both Half cheetah and Ant.
- **Actuator Strength**: The actuator strength was only varied for the half cheetah environment. For the training context, the values were selected at random within the ranges $[100, 200] \cup [400, 500]$ while for the testing context it was selected at random within the range $[200, 400]$

In the first stage where the contextual MDPs are trained using the vanilla RL framework, we employ the implementation of the PPO algorithm [13] provided by stable-baselines3 an open source reliable implementations of Reinforcement Learning algorithms [23]. In order to ensure that the policies learned from the contextual MDPs during the training stage are well trained, we set the total timesteps for training the PPO algorithm to 3×10^6. In the second stage, where CMA-ES is used to evolve the weights to combine the policy parameters from the training stage, we set the population size of CMA-ES to 10 and the total number of generations was set to 100. In terms of the objective function, mean episode rewards was used where each episode comprises of maximum of 1000 timesteps. Consequently, the total number of timesteps during the entire process is limited to 1×10^6.

All the experiments conducted in this work are carried out using Python installed on a 64-bit ubuntu 22.04 PC, with a

2.50GHz intel-i5 CPU and 16GB RAM.

C. Experimental results and discussions

1) Half Cheetah: Table I presents the results of the experiments conducted based on the Half Cheetah locomotion environment. Specifically, the mean rewards for the policies trained based on the contextual MDPs as well as the resulting policy from the proposed reparametrization scheme are presented. In order the demonstrate the generalization capabilities of the agents, the table presents test results across all the three contextual MDPs. From the results, it can be observed that the proposed reparameterization scheme results in an agent that outperforms the other agents when tested on all the contextual MDPs of the Half Cheetah locomotion environment. Furthermore, the results demonstrate that the proposed method results in a robust agent as the performance of the agent across all the contextual MDPs is highly comparable if not equal.

TABLE I: Evaluation results on Halfcheetah Environment

Agent	Training Env1	Training Env2	Testing Env
agent1	-958.1859131	-980.7937622	-934.630249
agent2	-1602.746216	-1609.807617	-1613.52002
avgeraged model	5646.800781	5602.361816	5718.705078

2) Ant: Table II presents the results of the experiments conducted based on the Ant locomotion environment in terms of the mean episodic rewards for the policies trained based on the vanilla RL scheme (PPO) as well as that of the resulting policy from the proposed reparametrization scheme. In order the demonstrate the generalization capabilities of the agents, the table presents test results across all the three contextual MDPs. From the results, it can be observed that the proposed reparameterization scheme results in an agent that outperforms the other agents when tested on all the contextual MDPs of the ant environment. Furthermore, the results demonstrate that the proposed method is able to generalize well across all the contextual MDPs as its performance is highly comparable for all the contexts.

TABLE II: Evaluation results on Ant Environment

Agent	Training Env1	Trainig Env2	Testing Env
agent1	468.9058228	498.521637	439.2935486
agent2	718.0932007	718.0866699	718.1958008
avgeraged model	1002.122498	1002.128296	1002.117432

V. Conclusion and Future Works

This paper proposed a novel optimal reparameterization scheme based on weighted model averaging to improve the generalization performance of reinforcement learning policies. The experimental results on the Ant and Half Cheetah benchmark environments demonstrated that the proposed technique can significantly enhance generalization capabilities compared to models trained on a single context. Specifically, the evolved ”averaged” policy consistently achieved higher rewards across both training and test MDPs compared to the individual policies it comprised. The success of this method highlights the potential of exposing agents to diverse training contexts and recombining their learned representations to create more robust policies. The weighting optimization through CMA-ES proves to be an effective way to combine the knowledge acquired by individual policies for generalization. Overall, this work makes a valuable contribution by introducing a simple yet effective model averaging approach to tackle the important challenge of generalization in reinforcement learning. Future works include combining an ensemble of more than two agents to see if it leads to an improvement in the results and evaluating the proposed solution on real-world robotics tasks to see if the generalization abilities can extend to real-world problems.

Acknowledgement

This research was supported by the Core Research Institute Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2021R1A6A1A03043144) and the National Research Foundation of Korea (NRF) under project BK21.

References

[1] L. Buşoniu, T. De Bruin, D. Tolić, J. Kober, and I. Palunko, “Reinforcement learning for control: Performance, stability, and deep approximators,” *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018.

[2] O. S. Ajani and R. Mallipeddi, “Adaptive evolution strategy with ensemble of mutations for reinforcement learning,” *Knowledge-Based Systems*, vol. 245, p. 108624, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122002817

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, “Mastering the game of go without human knowledge,” *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[5] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, “Grandmaster level in starcraft ii using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[6] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[7] S. Phon-Amnuaisuk, S.-J. Tan, Y.-H. Yap, F. C.-M. Choong, P. Vejjanugraha, K.-C. Khor, and K.-H. Ng, “Multi-agent traffic light controls with sumo,” in *2023 8th International Conference on Business and Industrial Research (ICBIR)*. IEEE, 2023, pp. 824–830.

[8] E. Aboyeji, O. S. Ajani, and R. Mallipeddi, “Effect of number of lanes on traffic characteristics of reinforcement learning based autonomous driving,” *IEEE Access*, pp. 1–1, 2023.

[9] A. Rajeswaran, K. Lowrey, E. Todorov, and S. Kakade, “Towards generalization and simplicity in continuous control,” in *NIPS*, 2017.

[10] X. Chen, J. Hu, C. Jin, L. Li, and L. Wang, “Understanding domain randomization for sim-to-real transfer,” *arXiv preprint arXiv:2110.03239*, 2021.

[11] F. Muratore, C. Eilers, M. Gienger, and J. Peters, “Data-efficient domain randomization with bayesian optimization,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 911–918, 2021.

[12] C. Benjamins, T. Eimer, F. Schubert, A. Biedenkapp, B. Rosenhahn, F. Hutter, and M. Lindauer, “Carl: A benchmark for contextual and adaptive reinforcement learning,” *arXiv preprint arXiv:2110.02102*, 2021.

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

[14] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es),” *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[15] N. Hansen and A. Auger, “Principled design of continuous stochastic search: From theory to practice,” in *Theory and Principled Methods for the Design of Metaheuristics*, 2014, pp. 145–180.

[16] H.-P. P. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*. USA: John Wiley & Sons, Inc., 1993.

[17] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies – a comprehensive introduction,” *Natural Computing*, vol. 1, pp. 3–52, 2004.

[18] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith *et al.*, “Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 965–23 998.

[19] V. Gupta, S. A. Serrano, and D. DeCoste, “Stochastic weight averaging in parallel: Large-batch training that generalizes well,” *arXiv preprint arXiv:2001.02312*, 2020.

[20] C. Packer, K. Gao, J. Kos, P. Krähenbühl, V. Koltun, and D. Song, “Assessing generalization in deep reinforcement learning,” *arXiv preprint arXiv:1810.12282*, 2018.

[21] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” *arXiv preprint arXiv:1606.01540*, 2016.

[22] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[23] 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. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html