Mitigating Non-Stationarity in Deep Reinforcement Learning with Clustering Orthogonal Weight Modification

Extended Abstract

Guoqing Ma †
Institute of Automation, Chinese Academy of Sciences
School of Future Technology, University of Chinese Academy of Sciences
Beijing, China
maguoqing2022@ia.ac.cn

Yuhan Zhang †
Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
zhangyuhan2022@ia.ac.cn

Yuming Dai
Institute of Automation, Chinese Academy of Sciences
School of Future Technology, University of Chinese Academy of Sciences
Beijing, China
daiyuming2025@ia.ac.cn

Guangfu Hao
Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
haoguangfu2021@ia.ac.cn

Yang Chen
Institute of Automation, Chinese Academy of Sciences
Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Chinese Academy of Sciences
Beijing, China
yang.chen@ia.ac.cn

Shan Yu *
Institute of Automation, Chinese Academy of Sciences
School of Future Technology, University of Chinese Academy of Sciences
Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Chinese Academy of Sciences
Beijing, China
shan.yu@nlpr.ia.ac.cn

ABSTRACT

RL agents often operate under the assumption of environmental stationarity, which poses a great challenge to learning efficiency since many environments are inherently non-stationary in state distribution. To address this issue, we introduce the Clustering Orthogonal Weight Modified (COWM) layer, which can be integrated into the policy network of any RL algorithm and mitigate non-stationarity effectively. By employing clustering techniques and a projection matrix, the COWM layer stabilize the learning process. Empirically, the COWM layer is integrated into various RL methods and outperforms state-of-the-art methods on the DMControl benchmark, highlighting its robustness and generality across various tasks and algorithms.

KEYWORDS

Reinforcement learning; Non-Stationarity; Orthogonal weight modification

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

ACM Reference Format:
Guoqing Ma †, Yuhan Zhang †, Yuming Dai, Guangfu Hao, Yang Chen, and Shan Yu *. 2025. Mitigating Non-Stationarity in Deep Reinforcement Learning with Clustering Orthogonal Weight Modification: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

In recent years, reinforcement learning (RL) has made significant progress across various domains, ranging from gaming to robotic control, often surpassing human performance [2, 4, 6, 7, 9]. Despite these advancements, a significant issue remains: the underlying assumption of a stationary environment [8]. In numerous RL tasks, environments are inherently non-stationary [1, 11], with critical environmental components undergoing time-dependent changes. In some extreme cases, the state transition function and reward function may both change over time [5]. This non-stationarity poses a challenge for RL agents in adapting effectively to dynamic environments.

We propose the COWM layer which mitigates non-stationarity and enhances the stability of the policy network in single-task by constraining its gradients. This approach minimizes interference with previously learned skills while learning new policies, thereby improving sample efficiency and convergence speed. The COWM layer exhibits high generalizability, making it suitable for all fully

†Equal Contribution
*Corresponding author

connected networks. It can be integrated into any RL algorithm that uses fully connected network as policy network. Experiments show that our method outperforms state-of-the-art vision-based and state-based RL approaches, significantly improving sample efficiency across various classical control tasks.

2 COWM LAYER IN POLICY NETWORK

2.1 Forward propagation and Projection matrix calculation

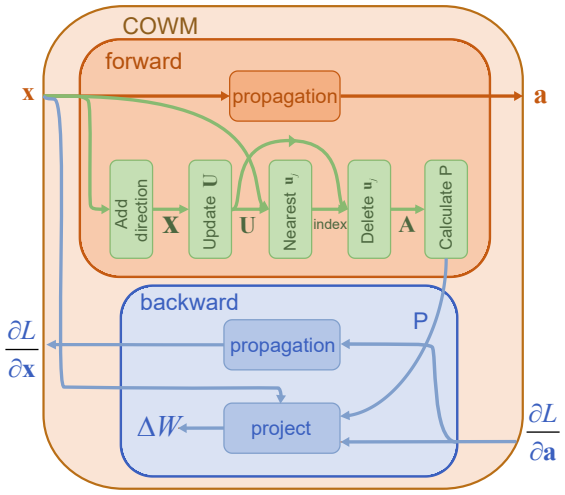

Figure 1: The architecture of COWM layer.

In terms of the computational process, Figure 1 illustrates the forward and backward propagation of the COWM layer. The forward propagation process remains unchanged. During each forward propagation process, a part of projection matrix P_l is computed (Eq. 1, 2).

$$\begin{aligned} U_{l-1} &= kmeans(X_{l-1}, c) \\ \mathbf{u}_{l-1}^{j} &= nearest(\mathbf{x}_{l-1}^{p}, U_{l-1}) \\ A_l &= \{U_{l-1}/\mathbf{u}_{l-1}^{j}\} \end{aligned} \tag{1}$$

$$P_l = A_l(A_l^T A_l)^{-1} \tag{2}$$

where $\mathbf{X}_{l-1} = \left[\bar{\mathbf{x}}_{l-1}^1, \bar{\mathbf{x}}_{l-1}^2, ..., \bar{\mathbf{x}}_{l-1}^F\right] \in R^{d \times F}$ is a matrix composed of the principal component of the input in layer l.

2.2 Orthogonal weight modification and Backpropagation

During backpropagation, the output layer of the neural network receives the gradient signal. On one hand, the gradient signal is propagated back to the input layer in the same manner as in standard BP algorithm (Eq. 3). On the other hand, the input is projected using the projection matrix before calculating the weight updates (Eq. 3).

$$\Delta W_l^{COWM} = -\eta \left(\frac{\partial L}{\partial \mathbf{a}_l} \mathbf{x}_{l-1} - \frac{\partial L}{\partial \mathbf{a}_l} P_l A_l^T \mathbf{x}_{l-1} \right) \tag{3}$$

Task	SAC	CURL	DrQ-v2	DreamerV3	COWM
Acrobot Swingup	5.1	5.1	128.4	210	**322**
Cartpole Balance	**963.1**	**979**	**991.5**	**996.4**	**999.7**
Cartpole Balance Sparse	**950.8**	**981**	**996.2**	**1000**	**1000**
Cartpole Swingup	692.1	762.7	**858.9**	819.1	831.4
Cartpole Swingup Sparse	154.6	236.2	706.9	**792.9**	770.1
Cheetah Run	27.2	474.3	691	728.7	**866.1**
Cup Catch	163.9	**965.5**	931.8	957.1	**983.4**
Finger Spin	312.2	**877.1**	**846.7**	818.5	829.3
Finger Turn Easy	176.7	338	448.4	787.7	**969.7**
Finger Turn Hard	70.5	215.6	220	810.8	**942.2**
Hopper Hop	3.1	152.5	189.9	369.6	**474.5**
Hopper Stand	5.2	786.8	893	900.6	**956.8**
Pendulum Swingup	560.1	376.4	839.7	806.3	**910.1**
Quadruped Run	50.5	141.5	407	352.3	**426.2**
Quadruped Walk	49.7	123.7	**660.3**	352.6	415.6
Reacher Easy	86.5	609.3	910.2	898.9	**985.1**
Walker Run	26.9	376.2	517.1	757.8	**764.4**
Walker Stand	159.3	463.5	974.1	976.7	**996.9**
Walker Walk	38.9	828.8	762.9	955.8	**982.8**
Median	86.5	463.5	762.9	810.8	**942.2**
Mean	236.7	510.2	682.8	752.2	**822.9**

Table 1: The performance of COWM and SOTA baselines on vision-based DMControl tasks under 1M environment steps across 3 random seeds.. The bold values are the highest among each row.

$$W_l(t+1) = W_l(t) + \Delta W_l^{COWM} \tag{4}$$

Finally, the COWM layer must also complete the backpropagation of gradients (Eq. 5). The COWM layer transmits the gradient information from the output layer back to the input side in the same manner as a linear layer. This gradient is used to train the deep neural network layer by layer.

$$\frac{\partial L}{\partial \mathbf{x}_{l-1}} = W_l \frac{\partial L}{\partial \mathbf{a}_l} \tag{5}$$

We adopt the actor-critic learning settings from DreamerV3 [3].

3 EXPERIMENTS

We evaluate the performance of COWM on the widely-used DMControl benchmark [10]. We evaluate the methods on 19 vision-based DMControl tasks. All experimental results and part of the training curves are presented in Table 1. The results show that COWM outperforms previous SOTA methods under 1M interactions on 15 tasks.

ACKNOWLEDGMENTS

This work was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences (CAS)(XDB1010302), CAS Project for Young Scientists in Basic Research, Grant No. YSBR-041 and the International Partnership Program of the Chinese Academy of Sciences (CAS) (173211KYSB20200021).

REFERENCES

[1] Samuel PM Choi, Dit-Yan Yeung, and Nevin L Zhang. 2001. Hidden-mode markov decision processes for nonstationary sequential decision making. *Sequence learning: paradigms, algorithms, and applications* (2001), 264–287.
[2] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz,

Julian Schrittwieser, Grzegorz Swirszcz, et al. 2022. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature* 610, 7930 (2022), 47–53.

[3] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2023. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104* (2023).

[4] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. 2023. Champion-level drone racing using deep reinforcement learning. *Nature* 620, 7976 (2023), 982–987.

[5] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. 2022. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research* 75 (2022), 1401–1476.

[6] Guoqing Ma, Zhifu Wang, Xianfeng Yuan, and Fengyu Zhou. 2022. Improving Model-Based Deep Reinforcement Learning with Learning Degree Networks and Its Application in Robot Control. *Journal of Robotics* 2022, 1 (2022), 7169594.

[7] Daniel J Mankowitz, Andrea Michi, Anton Zhernov, Marco Gelmi, Marco Selvi, Cosmin Paduraru, Edouard Leurent, Shariq Iqbal, Jean-Baptiste Lespiau, Alex Ahern, et al. 2023. Faster sorting algorithms discovered using deep reinforcement learning. *Nature* 618, 7964 (2023), 257–263.

[8] Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[9] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

[10] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690* (2018).

[11] Annie Xie, James Harrison, and Chelsea Finn. 2021. Deep reinforcement learning amidst continual structured non-stationarity. In *International Conference on Machine Learning*. PMLR, 11393–11403.