Multiple-Model Based Defense for Deep Reinforcement Learning Against Adversarial Attack

Patrick P. K. Chan[1(✉)], Yaxuan Wang[1], Natasha Kees[1], and Daniel S. Yeung[2]

[1] School of Computer Science and Engineering,
South China University of Technology, Guangzhou, China
patrickchan@scut.edu.cn
[2] IEEE SMC Society, Guangzhou, Hongkong

Abstract. Deep Reinforcement Learning models inherit not only generalization abilities but also vulnerabilities under adversarial attacks from Deep Neural Networks. The recent external model based defense method for Reinforcement Learning (RL) detects and corrects the action relying on only the observation prediction method. The observation prediction method may not perform well in complicated applications because of the knowledge of environment, which will downgrade the defense efficacy. This study proposes a multiple-model based defense method for RL which considers detection and correction tasks separately. Since the problem is broken down into two tasks, their complexity and difficulty is also lower, *i.e.,* a better performance is expected. We propose a Correlation Feature Map to extract the observation consistency in the temporal sequence which is destroyed by adversarial noise to separate clean and attacked states. Our correction only deal with the states classified as contaminated and maps them to proper actions. The performance of our proposed method is evaluated and compared to the state of the art method experimentally in various settings. The results confirm the superiority of our methods in terms of robustness and time.

Keywords: Adversarial learning · Reinforcement learning · Robustness

1 Introduction

Deep Neural Networks (DNNs) have been rapidly developed because of the excellent performance. However, many studies [5,16] show that DNNs are vulnerable in an adversarial environment in which an adversary may craft a sample in order to mislead a target system. Since Deep Reinforcement Learning (DRL) methods inherit the capabilities of DNNs, they have many landmark achievements

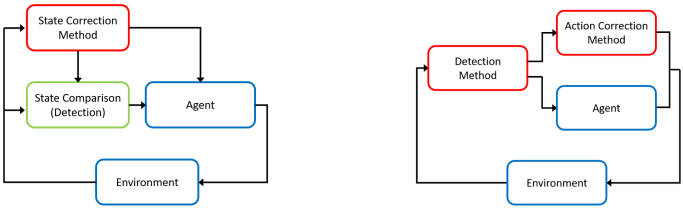
Supported by the Natural Science Foundation of Guangdong Province, China (No. 2018A030313203).

I. Farkaš et al. (Eds.): ICANN 2021, LNCS 12891, pp. 42–53, 2021.
https://doi.org/10.1007/978-3-030-86362-3_4

recently in various applications [11,12,15]. However, the vulnerability to adversarial samples of DNNs is also inherited by the DRL methods [3,7,9]. Recently, most of the discussions on adversarial attacks are in the framework of Supervised Learning which assumes samples are independent and identically distributed. However, this assumption does not apply to RL in which the consecutive states are closely related since they may share the same observations. As the setting of RL is different from Supervised Learning, the security of RL should be investigated separately. However, few studies focus on defense methods for RL. The influence of adversarial attacks is reduced by considering adversarial samples in training [8,14] using a robust loss function [4] or specially designed model structure [1,6], and using external models [10].

The defense methods using external models are focused in this study due to flexiblilty. To the best of our knowledge, there is only one defense method [10] in this type of defense. Both contamination detection and action correction of this method rely on an observation prediction model. We categorize this method as a single-model based defense method since both detection and correction only rely on one model, as shown in Fig. 1(a). The detection task is a simple comparison between the obtained state and corrected state, as shown in the green block. However, predicting an observation accurately in RL is difficult since comprehensive understanding of an environment is required. Unaccurate prediction will cause both incorrect detection and correction, *i.e.,* the mistakes will cumulate.

Our study devises a multiple-model based defense method for RL, shown in Fig. 1(b), which considers detection and correction tasks separately. Compared with a single-model based method, our multiple-model based method avoids the cumulation of mistakes in detection and correction tasks. In our detection model, a feature map named Correlation Feature Map (CFM) is proposed to extract the consistency of observations in the temporal sequence in a state in our detection method. The observation continuity is usually ignored when crafting an adversarial sample in recent attack methods as each state is modified independently. Therefore, the different patterns of observation continuity in clean and contaminated states are expected. Moreover, a correction method is devised in order to reduce the influence of the adversarial states. Since making a correct decision is the main objective, recovering an observation is difficult and unnecessary.



(a) Single-Model based (b) Multiple-Model based

Fig. 1. Block diagrams of the Single-Model and Multiple-Model based Defense Method. (Color figure online)

Therefore, our method maps an attacked state to a correct decision directly. Simplicity is another advantage of our multiple-model based method compared to the single-model based one. As both detection and correction tasks are simpler than the observation prediction, a simpler model structure can be used. Not only a shorter training time but also a faster decision time is expected. Our proposed method is evaluated and compared experimentally with the state-of-the-art defense methods in three Atari games. Experimental results confirm the effectiveness of our proposed method in terms of robustness and speed, compared with the state-of-the-art defense methods.

The main contributions of this work are summarized as follows:

- A multiple-model based defense method combining detection and correction tasks against adversarial attacks in RL is proposed. It utilizes the accurate contamination detection to enhance the overall performance.
- The continuity of temporal observation sequence is captured by the proposed Correlation Feature Map in our detection model.
- A simple action correction method is proposed to reduce the impact of adversarial samples.
- Experimental results confirm the effectiveness of our method, compared with state-of-the-art methods.

2 Related Work

Adversarial Attack: Adversarial attack, sometimes called evasion attack [16], aims to reduce the performance of a model by manipulating samples in inference. Many studies prove that DNNs are sensitive to adversarial attacks [2,5,13,16]. In such attacks, it is assumed that samples in inference can be manipulated by a small change. Adversarial attacks against DRL mainly follow the idea of attacks in the Supervised Learning, *i.e.,* each sample is attacked independently. To be specific, an adversarial state $\tilde{s}$ is generated by an attack method A adding adversarial perturbation η to clean state s, *i.e.,* $\tilde{s} = A_\theta(s) = s + \eta$, where θ denotes the parameters of A. Fast Gradient Sign Method ($FGSM$) [5] is a common and famous *one-time attack* due to its low time complexity. The adversarial perturbation is calculated by the gradient of the model function:

$$\eta = \epsilon * sign(\nabla_s J(\theta, s, a)) \tag{1}$$

where $J(\theta, s, a)$ denotes the cross-entropy loss, and ϵ controls the upper limit of the perturbation. Carlini & Wagner L2 ($CWL2$) [2], which optimizes a near-optimal adversarial state in terms of minimal adversarial perturbation:

$$minimize \quad \|\eta\|_2^2 + c * max(max\{\pi(s+\eta)_i : i \neq t\} - \pi(s+\eta)_t, -\kappa) \tag{2}$$

where $\pi()_i$ is the i-th output of the target RL policies π, κ controls the attack confidence.

Adversarial Defense: Some defense methods have been proposed in order to reduce the influence of adversarial attacks to RL models and can be divided into

three categories: data modification, model modification and external models. Flexibility is a concern for both data and model modification methods, since retraining of the target model is required. The new models may perform worse than the original ones, especially when the attack is absent, *i.e.,* the general performance may be sacrificed for the robustness. As a result, our study focuses on the third type of defense methods which rely on external models. The only existing method using external models in RL Foresight [10] predicts the current observation x according to multiple historical observations and the last action by an action-conditioned observation prediction model. An observation considered contaminated based on whether or not it has a large difference from the predicted observation. The contaminated observation is then replaced by its predicted one for decision.

3 Analysis on Single-Model and Multiple-Model Based Defense

The performance analysis on single-model and multiple-model based defense is given in this section. We argue that a multiple-model based defense which contains the detection and correction models separately is able to achieve better performance. Let $p_{s,cor}$ be the accuracy of the only model used in a single-model based defense method, which quantifies the probability of corrected and original actions being the same. The defense accuracy of a single-model based method ACC_s is calculated as:

$$ACC_s = p_{s,cor} \tag{3}$$

On the other hand, let p_{det} and $p_{m,cor}$ be the detection and correction accuracy respectively for a multiple-model based defense method. Let α be the attack ratio defined as the probability of attack, *i.e.,* $\alpha \in [0, 1]$. The detection model first determines whether a state is contaminated, and then the correction model corrects the contaminated states. The overall defense accuracy of the multiple model ACC_m is defined as:

$$ACC_m = (1 - \alpha) * p_{det} + \alpha * p_{det} * p_{m,cor} \tag{4}$$

We assume $p_{cor} = p_{s,cor} = p_{m,cor}$ for discussion since their corresponding tasks both aim at correcting actions. In other words, the model used in a single-model based method can be used in the correction task of multiple-model based defense. By considering Eq. (3) and (4), the following statement can be obtained:

$$p_{cor} \leq \frac{(1 - \alpha) * p_{det}}{1 - \alpha * p_{det}} \quad if\ and\ only\ if \quad ACC_m \geq ACC_s \tag{5}$$

Lines in Fig. 2 represents $ACC_m = ACC_s$ (*i.e.,* $p_{cor} = \frac{((1-\alpha)*p_{det})}{(1-\alpha*p_{det})}$) with $\alpha \in \{0, 0.01, 0.25, 0.5, 0.75, 0.99, 1\}$. Since the detection task is generally simpler than the correction task, it is reasonable to ignore the grey-triangle region ($p_{cor} \geq p_{det}$). By considering the triangle area under the diagonal, the area below and

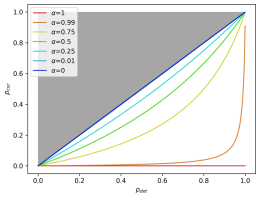


Fig. 2. Values of p_{cor} and p_{det} when $ACC_s = ACC_m$ with different α.

above the line $ACC_m = ACC_s$ are denoted by A_b and A_a respectively. A_b contains (p_{det}, p_{cor}) pairs which cause $ACC_m > ACC_s$, while A_a contains the ones which cause $ACC_m > ACC_s$. The size of A_b is much larger than A_a when *alpha* tends to 0. This means that even slight improvement of p_{det} from p_{cor} will cause $ACC_m > ACC_s$. A_b/A_a decreases with the increase of α. When α tends to 1, A_b tends to 0. This analysis indicates that multiple-model based defense is more suitable to deal with attacks with smaller attack ratio compared with the single one. Moreover, the multiple-model based method prefers the situation in which the detection accuracy is much larger than the correction accuracy.

By considering the the partial derivative of ACC_m with respect to p_{det} and p_{cor}, p_{det} plays a more important role than p_{cor} in the performance of a multiple-method based model when α is smaller:

$$if\ \Delta p \leq \frac{1}{\alpha} - 1,\quad then\ \frac{\partial ACC_m}{\partial p_{det}} \geq \frac{\partial ACC_m}{\partial p_{cor}} \tag{6}$$

where $\Delta p = p_{det} - p_{cor}$. When α is smaller, $1/\alpha - 1$ is larger. It implies the chance of satisfying that the hypothesis (*i.e.,* $1/\alpha - 1 \geq \Delta p$) is higher. As a result, when the attack rate is smaller, p_{det} plays a more important role in the multiple-method based model than p_{cor}, and vice versa. As attacked states are the minority in reality, the detection performance is more important in a defense method. Therefore, the main contribution of our proposed defense method focuses on the detection method, which is discussed in next section.

4 Proposed Approach

The defense methods using external models are investigated in our study since they can be applied to any RL model without retraining. As discussed in the previous session, a multiple-model based defense method is more efficient than a single-method based one. A multiple-model based defense, including the detection and action correction models are proposed for RL. Our proposed method detects a contaminated state according to the correlation between observations in a state since the continuity of consecutive observations is destroyed by the adversarial perturbation. The continuity of consecutive observations in a state is extracted by the proposed Correlation Feature Map (CFM). The decision on a clean state is then made by the original RL model, while a contaminated state is handled by our correction model.

4.1 Detection Model

A state containing a number of consecutive observations captures the responses of the environment in a sequence of time. However, most adversarial attack methods do not consider time continuity when crafting adversarial samples. As a result, the original distribution of observations in an adversarial state is disturbed, *i.e.*, the observations in adversarial states follow different distributions with ones in clean states.

Our detection method aims to capture the difference of observation distributions between adversarial and clean states. A state s_t at time t is defined as the stack of N consecutive observations in RL, *i.e.*, $s_t = \phi(x_{t-(N-1)}, ..., x_{t-1}, x_t)$, where $\phi()$ denotes a stacking function. The feature map of each observation of a state is first extracted independently based on the target DRL model. To extract the information from x_i, a state s_t^i is generated by setting all values in s_t to zero except the ones in x_i:

$$s_t^i = \phi(\boldsymbol{O}, ..., \boldsymbol{O}, x_i, \boldsymbol{O}..., \boldsymbol{O}) \tag{7}$$

where $\boldsymbol{O}$ is a zero matrix with the size of an observation. The feature map defined as the output of the m-th layer of the DRL model is then extracted for s_t^i. This setting preserves the information in x_i while ignoring the other observations. We propose the Correlation Feature Map (CFM) defined as the stack of the feature maps of all s_t^i in s_t, shown in Eq. (8). Figure 3 shows the CFM generation.

$$\text{CFM}(s_t) = \phi(f_m(s_t^{t-(N-1)}), ..., f_m(s_t^{t-1}), f_m(s_t^t))) \tag{8}$$

where $f_m()$ is defined as the feature map extracting function of the m-th layer of the target RL model. The last convolutional layer is chosen as f_m in this study since it preserves spatial information and contains more abstract features compared to other layers.

Finally, a detection model g_{det} is trained to distinguish clean states from adversarial ones based on their CFM difference. A dataset $\mathcal{D}$ containing clean states s and adversarial states $\tilde{s}$ is generated in advance. The clean states s are collected randomly from the interaction between the target RL model and environment. For each clean state s, corresponding adversarial state $\tilde{s}$ is crafted according to an attack method with particular parameters. A simple network

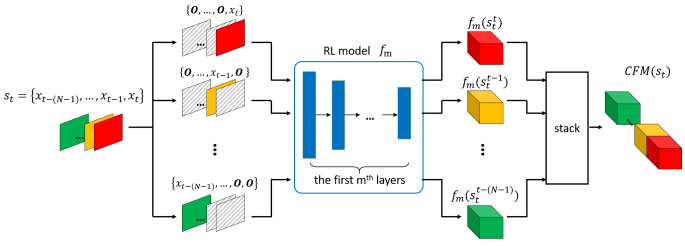


Fig. 3. Generation of Correlation Feature Map function (CFM)

structure of g_{det} should be used since the CFM are already extracted by the RL model. The structure consisting of two convolutional layers and four fully-connected layers is used in this study. All the convolutional layers have 512 channels with kernel size 3 * 3 and stride 1, while the sizes of fully-connected layers are 1024, 128, 16 and 1. The batch normalization is implemented after the first convolutional layer. The loss function is

$$L_{det} = -\log(h(\text{CFM}(\tilde{s}))) - \log(1 - h(\text{CFM}(s))) \tag{9}$$

where $h(a) = 1/\left[1 + exp(-g_{det}(a))\right]$. When detecting a new state s, its CFM is input to the detection model, *i.e.,* $g_{det}(\text{CFM}(s))$. If the output is larger than 0, s is classified as an attacked state; otherwise, it is clean.

4.2 Correction Model

This section devises an action correction model $\pi_{cor} : s \rightarrow a$ mapping an adversarial state $\tilde{s}$ to the decision made by the target model π on s, *i.e.,* $\pi_{cor}(\tilde{s}) = \pi(s)$. The model only handles the states classified as contaminated, which makes the correction task simpler since the characteristics of clean and contaminated states are different. The corrected action is then used to interact with the environment. Our correction model is trained by minimizing the following regression loss using s and $\tilde{s}$ in $\mathcal{D}$:

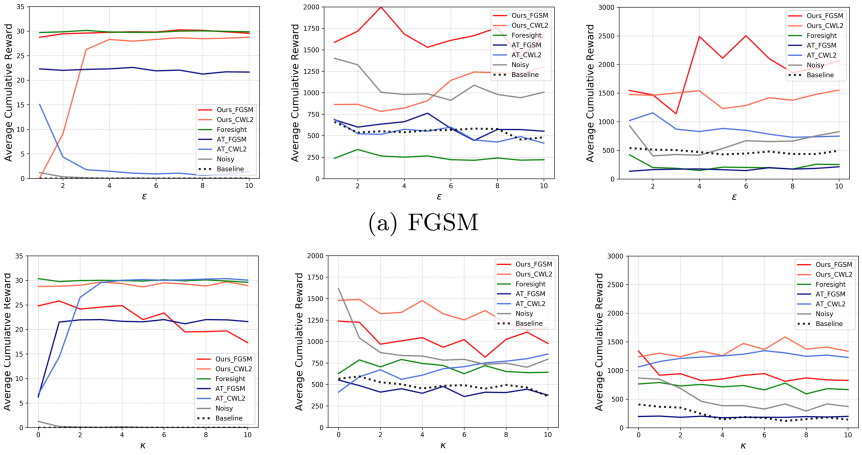
$$L_{cor} = \|\pi_{\text{cor}}(\tilde{s}) - \pi(s)\|_2 \tag{10}$$

The learning of π_{cor} is guided by the output of π on s. The structure of π_{cor} is set as the same as π since their tasks are similar. In order to take advantage of the feature extraction ability of π, its parameters are applied to initialize π_{cor}.

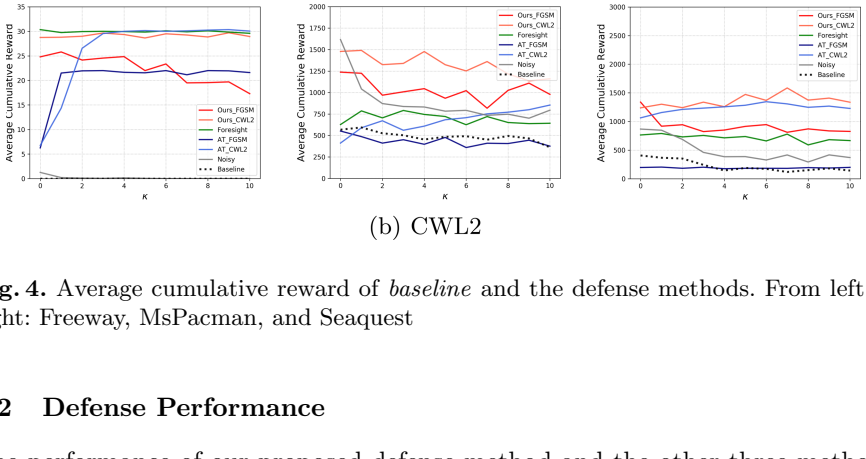
5 Experiment

5.1 Experimental Setting

Three Atari games (Freeway, MsPacman and Seaquest) are considered. DQN is used as the target agent, and all the settings follow [12]. The program codes will be available when the paper is accepted. Two popular adversarial attack methods, *FGSM* and *CWL2*, are used. Different attack parameters are considered, *i.e.,* $\epsilon = \{1, 2, ..., 10\}$ for *FGSM* and $\kappa = \{0, 1, ..., 10\}$ for *CWL2*. *Baseline* which contains only DQN but no defense method is considered for comparison. *Ours_A* represents our model trained with D contaminated by attack method A, *i.e., Ours_FGSM* and *Ours_CWL2*. In D, 10,000 clean states are collected, and corresponding adversarial states generated with random attack parameters from the range mentioned above. The attack ratio α is 0.5. Our detection and correction models are trained for 50 and 200 epochs respectively. *Foresight* [10], *AT* [8], and *Noisy* [1] are used for comparison. All the settings follow the original papers.



(a) FGSM



(b) CWL2

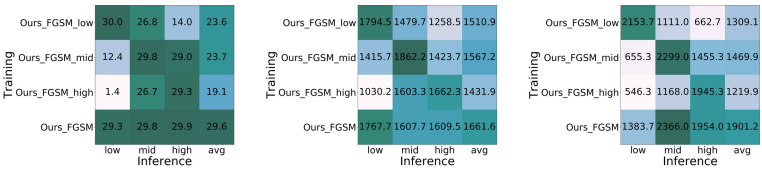
Fig. 4. Average cumulative reward of *baseline* and the defense methods. From left to right: Freeway, MsPacman, and Seaquest

5.2 Defense Performance

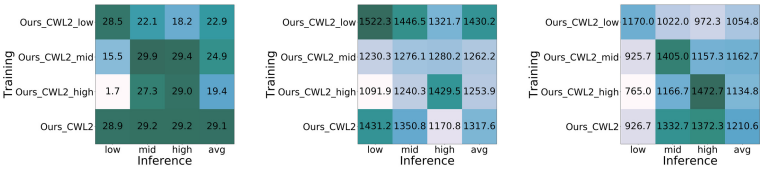
The performance of our proposed defense method and the other three methods are discussed in this section. Figure 4 shows the average cumulative reward of defense methods under *FGSM* and *CWL2* attacks with different parameters. When the attack methods in training and inference are the same, our methods perform significantly better than other all methods, except *Foresight* in Freeway. Our methods perform slightly worse than *Foresight* in Freeway. *Foresight* has excellent performance in Freeway but not in MsPacman and Seaquest, because the environment of Freeway is the simplest. Even when the attack methods used in training and inference are different, the performance of our methods is still satisfying. These results confirm the efficacy of our defense method against adversarial attacks. Although the knowledge on the attack strategy may be inaccurate, its performance is still consistently reasonable.

5.3 Transferability on Attack Parameters

This section discusses the transferability of our defense method across attack parameters used in the training and inference phases. Three levels of attack parameters are defined for each attack method according to the intensity: low, middle and high, which are $\epsilon \in [1,3], [4,6], [8,10]$ for *FGSM*, and $\kappa \in [0,2]$, $[4,6]$, $[8,10]$ for *CWL2*. Figure 5 shows the performance of our model training with one attack level under different levels. Better defense performance yields a higher average cumulative reward, which is illustrated by a darker color. The cells in the diagonal, representing that same parameter settings are used in both training and inference, achieve the largest rewards. Our method has the best



(a) FGSM



(b) CWL2

Fig. 5. Average cumulative reward of our method with different level of attack parameter. From left to right: Freeway, MsPacman, and Seaquest

performance when the full knowledge on the attack is obtained. On the other hand, more different attack settings used in training and inference reduces the efficacy of our method more significantly. For example, the reward achieved by the model using HIGH is lower than the ones using MID and LOW when the attack is using LOW in inference. When the real attack parameters are unknown, using weaker attack parameters is preferable. Regarding to the models trained with MID, its performance under HIGH is better than the one under LOW.

5.4 Performance on Detection Task

The detection performance of our method is discussed and compared with *Foresight* in this section. For each attack method, a test set consists of 10,000 clean states and 10,000 adversarial states. The experimental results are shown in Table 1. The largest accuracy in a column is bolded. Our method achieves excellent performance in detection compared with *Foresight*. Accuracy of our detection model is larger than *Foresight* in all cases no matter which attack method is used in training, except Freeway under *FGSM*. Accuracy of our method using the same attack method in training and inference is higher than 95% in all settings, while it is higher than 83% when the attack methods are different. The performance of our method drops when different attack methods are used in training and inference. Accuracy of *Ours_CWL2* is higher than *Ours_FGSM* under both attack methods. One possible explanation is adversarial samples generated by *CWL2* are more various than *FGSM*. On the other hand, the observation prediction in *Foresight* is inaccurate when the applications are complicated, *e.g.,* MsPacman and Seaquest.

Table 1. Detection accuracy (%).

(a) FGSM

	Freeway	MsPacman	Seaquest
Foresight	**99.99**	53.27	59.82
Ours_FGSM	99.77	**99.33**	**99.11**
Ours_CWL2	87.95	92.01	92.22

(b) CWL2

	Freeway	MsPacman	Seaquest
Foresight	95.18	66.56	72.63
Ours_FGSM	95.55	85.28	83.80
Ours_CWL2	**95.63**	**98.04**	**96.23**

Table 2. Correction accuracy on adversarial samples (%).

(a) FGSM

	Freeway	MsPacman	Seaquest
Foresight	**91.16**	**45.28**	29.61
Ours_FGSM	90.61	43.47	**30.79**
Ours_CWL2	67.21	38.11	15.57

(b) CWL2

	Freeway	MsPacman	Seaquest
Foresight	**89.74**	41.79	22.27
Ours_FGSM	44.27	37.45	14.31
Ours_CWL2	88.46	**47.46**	**22.92**

5.5 Performance on Correction Task

The performance on adversarial samples of the correction task is shown in Table 2. Our method and *Foresight* are evaluated by the probability of actions in clean states and corrected actions being the same. Since our correction model only deals with attacked states, only accuracy on adversarial samples is considered. Also, the results show that *Foresight* has similar accuracy on clean and adversarial samples. Our correction model performs similarly with *Foresight* when the same attack methods are considered in training and inference, *i.e.,* the accuracy difference is less than 5% in most cases. Although the accuracy of our correction is lower than *Foresight* when using different attack methods in training and inference, the overall performance of our defense method is still higher due to our better detection performance. This confirms the superiority of the multiple-model based defense compared with the single one, which is consistent with the analysis in Sect. 3. Also, similar to the results found in last discussion, the correction performance of *Foresight* highly relies on the complexity of the environment.

5.6 Influence of Attack Ratios

This section investigates the influence of the attack ratio α on the performance of our method and *Foresight*. Figure 6 show the average cumulative reward under *FGSM* with $\epsilon = 5$ when $\alpha = \{0.1, 0.2, ..., 0.9\}$ respectively. It shows in general, a larger α causes a smaller average cumulative reward. The reward decreases faster when attack methods in training and inference are different for our method. From the results shown in Table 1, the detection accuracy of our method when using different attack methods in training and inference is close to the one using the same attack methods. However, the correction accuracy of these two situations are much different, shown in Table 2. On the other hand, the performance of

Fig. 6. Average cumulative reward with different levels of attack ratio α.

Foresight is more stable than our methods since its performance on both clean and contaminated states is similar. Moreover, the advantages of our method using different attack methods reduce compared to *Foresight* with the increase of α. This confirms the discussion in Sect. 3.

5.7 Time Complexity

The average training times in MsPacman are 7.8, 17.9, 18.5 and 53.0 h for our method, *Foresight*, *AT* and *Noisy* respectively. It show that our method require the shortest training time, *i.e.,* average 7.8 h although our model contains the RL agent, detection and correction models. This may be because our detection and correction models have relatively simple structures. *Noisy* needs the longest training time since the revised RL model is more complicated than the original one. The training time of *AT* and *Foresight* are more than double of our method because of retraining. On the other hand, the average inference times are 10.2, 16.2, 0.7 and 0.7 ms respectively. The inference time of *AT* and *Noisy* is the shortest among all methods since they do not rely on an external model. Our time is shorter than *Foresight* because of the simpler network structure.

6 Conclusion

The robustness of single-model based methods, *e.g.,* Foresight, relies on the accuracy of the state correction process, which is usually a difficult task since comprehensive understanding of the environment is required. Therefore, single-model based methods may not be suitable for complicated applications. A multiple-model based defense method containing the detection and correction tasks is proposed in this study. Since the complexity of these tasks is lower, a simpler model can be used and also better performance is expected. Our method distinguishes attacked states from clean states by using the proposed Correlation Feature Map (CFM) which captures the observation consistence in the temporal sequence of a state. The decision on an attacked state is made by our action correction method mapping an attacked state to its original action. All the experimental results illustrate that our proposed defense method achieves excellent performance and outperforms the other methods in terms of robustness and running time. This study provides a foundation for developing secure RL systems.

References

1. Behzadan, V., Munir, A.: Mitigation of policy manipulation attacks on deep q-networks with parameter-space noise. In: Computer Safety, Reliability, and Security - SAFECOMP 2018 Workshops, vol. 11094, pp. 406–417 (2018)
2. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks (2017)
3. Chan, P.P., Wang, Y., Yeung, D.S.: Adversarial attack against deep reinforcement learning with static reward impact map. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, pp. 334–343 (2020)
4. Gallego, V., Naveiro, R., Insua, D.R.: Reinforcement learning under threats. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp. 9939–9940 (2019)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Proceedings of the 3rd International Conference on Learning Representations (2015)
6. Havens, A.J., Jiang, Z., Sarkar, S.: Online robust policy learning in the presence of unknown adversaries. In: Proceedings of the 2018 Annual Conference on Neural Information Processing Systems, pp. 9938–9948 (2018)
7. Huang, S., Papernot, N., Goodfellow, I., Duan, Y., Abbeel, P.: Adversarial attacks on neural network policies. arXiv preprint arXiv:1702.02284 (2017)
8. Kos, J., Song, D.: Delving into adversarial attacks on deep policies. In: Proceedings of the 5th International Conference on Learning Representations (2017)
9. Lin, Y.C., Hong, Z.W., Liao, Y.H., Shih, M.L., Liu, M.Y., Sun, M.: Tactics of adversarial attack on deep reinforcement learning agents. arXiv preprint arXiv:1703.06748 (2017)
10. Lin, Y.C., Liu, M.Y., Sun, M., Huang, J.B.: Detecting adversarial attacks on neural network policies with visual foresight. arXiv preprint arXiv:1710.00814 (2017)
11. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
12. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
13. Moosavi-Dezfooli, S., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
14. Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., Chowdhary, G.: Robust deep reinforcement learning with adversarial attacks. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2040–2042 (2018)
15. Silver, D., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)
16. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)