BARReL: Bottleneck Attention for Adversarial Robustness in Vision-Based Reinforcement Learning

Eugene Bykovets*
D-INFK, ETH Zürich
eugene.bykovets@inf.ethz.ch

Yannick Metz*
University of Konstanz
yannick.metz@uni-konstanz.de

Mennatallah El-Assady
ETH AI Center, Zürich
melassady@ethz.ch

Daniel A. Keim
University of Konstanz
keim@uni-konstanz.de

Joachim M. Buhmann
D-INFK, ETH Zürich
jbuhmann@inf.ethz.ch

Abstract

Robustness to adversarial perturbations has been explored in many areas of computer vision. This robustness is particularly relevant in vision-based reinforcement learning, as the actions of autonomous agents might be safety-critic or impactful in the real world. We investigate the susceptibility of vision-based reinforcement learning agents to gradient-based adversarial attacks and evaluate a potential defense. We observe that Bottleneck Attention Modules (BAM) included in CNN architectures can act as potential tools to increase robustness against adversarial attacks. We show how learned attention maps can be used to recover activations of a convolutional layer by restricting the spatial activations to salient regions. Across a number of RL environments, BAM-enhanced architectures show increased robustness during inference. Finally, we discuss potential future research directions.

1. Introduction

Visual-based reinforcement learning (RL) is a sub-field of reinforcement learning research that uses an image as input for the decision-making process. Prominent examples include video games like Atari [1, 8] or robotics applications. Learning from raw pixels is a promising approach because it is generally applicable and requires little feature engineering, but it can be challenging in practice. While RL has profited from advances in deep learning architectures for computer vision, it has inherited its susceptibility to adversarial attacks [6, 10]. In fact, trained RL agents are often very sensitive to the model to the quality of the visual input, i.e., they are easily corrupted due to environmental/equipment factors, like different lighting conditions, shadowing, camera quality/damage, or by adversarial intentions of the third parties, like adversarial attacks. The robustness of such systems is crucial for potential future real-world use, especially in safety-critical applications. In this work, we (1) propose a conceptually simple, general, yet effective inference-time method for defense against adversarial attacks. (2) In initial experiments, we demonstrate the effectiveness of the approach. Finally, we (3) discuss potential future research directions and extensions.

2. Related work

Adversarial attacks in Reinforcement Learning There are a number of previous works that tackle the problem of adversarial attacks on reinforcement learning agents. In RL, the goal of adversarial attacks is to deteriorate the policy's performance. Similar to the supervised settings, attacks can be classified into *white-* and *black-box* attacks [5]. Beyond attacks found in supervised settings, which are comparable to attacking the state/observation space of an agent [2, 10], there exist additional attacks such as poisoning the environment dynamics or reward function [5]. In this work, we focus on white-box state space attacks, i.e., attacks on the input observations in which the attacker has access to the underlying model.

Attention methods in Computer Vision Different regions of an image are not equally important for predictions. Identifying the most important part can be done via attention mechanisms. There are various approaches to model attention in computer vision [4]. In this work, we utilize *Bottleneck Attention Modules* (BAM) [9] which applies both spatial and channel-wise attention. Following common practice, we utilize *frame stacking* of a small sequence of proceeding game frames, which means that here channel-wise

[*] Equal contribution

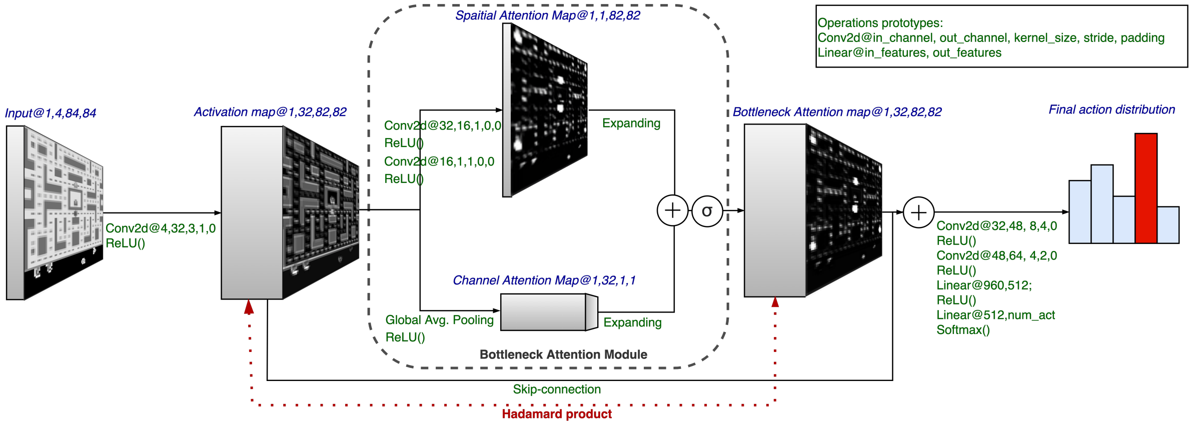

Figure 1. We specify all the tensors on top of picture of the tensor with *blue italic font* and operations with green font providing legend in top-right corner. The architecture of the **encircled Bottleneck Attention Module** can be found in Park et al. [9]. The dotted red line indicates the elements used in the "recovery" mechanism based on attribution maps.

attention enables temporal attention.

Defenses against adversarial attack Existing defense strategies include methods like adversarial training and game-theoretic approaches to defend against inference-time attacks or robust learning against training-time attacks [5]. In this work, we combine adversarial training with a novel defense strategy to defend against inference-time attacks.

3. Method

For our approach, we draw inspiration from selective attention [14]: only a part of visual input is necessary for decision making, i.e., a significant part of an input image contains non-relevant information. Gradient-based adversarial attacks modify both relevant and non-relevant parts of visual input. The entirety of the perturbation can corrupt the predicted output action distribution of an RL agent and consequently cause potentially erroneous behavior. Thus, our goal is to mitigate the effect of these attacks by removing non-relevant parts of the input, which can dramatically reduce the magnitude of the perturbation. We utilize *Bottleneck Attention Module* [9], which has been introduced as a simple and non-invasive extension to existing CNN architectures, to learn attention maps of the most relevant spatial and temporal features for vision-based RL. We use learned attention maps to *"recover"* the output of a convolutional layer perturbed by an adversarial attack. In effect, we use the attention maps to only retain relevant parts of the input.

3.1. Implementation Details

Training We use Proximal Policy Optimization [13], with the implementation and default training hyperparameters from the `StableBaseline3` library [11]. We train all models on three Atari environments, which are the part of a well-established benchmark used to compare RL methods [1], namely: `Breakout-v4`, `SpaceInvaders-v4`, `MsPacman-v4`. We apply the common pre-processing routing for observation; namely, transformation to greyscale, resizing, a frame-skipping of 4 frames, as well as stacking of the 4 last frames to encode temporal dynamics. As a baseline architecture (`Nature-CNN`), we use a 3-layer CNN, followed by two fully-connected layers and a softmax that outputs a categorical probability distribution over a set of possible actions similar to [8]. We extend this architecture by embedding a *BAM*-layer [9] between the first and second convolutional layer (`BAM-CNN`). These architectures contain 2.246.416 parameters (`Nature-CNN`) and 2.248.409 parameters (`BAM-CNN`), respectively. Figure 1 shows the used architecture. To have a performance baseline, we trained both Nature-CNN and BAM-CNN architectures on the default environments (denoted without any additional suffix in Tab. 2). We utilize adversarial training by applying Projected Gradient Descend (PGD) attacks [7] using `Foolbox` framework [12] with an $\epsilon = 0.1$ and get two additional training regimes: `Nature-CNN-Adv.` and `BAM-CNN-Adv.`. In RL, the quality of training is directly dependent on the learned policy, thus applying attacks too aggressively, e.g., at every step, might stop an agent from learning altogether. We found that attacking only every 10th frame had no significant impact on training performance see Tab. 2), but in turn only gave moderate robustness to adversarial attacks (Tab. 3 shows that adversarial attacks are successful with approx. 100% probability). However, our experiments showed that adversarial training seems to be necessary to learn appropriate attention maps. When trained on the default environment, the learned attention maps were generally not suited to support the recovery of activations.

Inference Performance of the RL agents at the inference stage with respect to cumulative reward, averaged over 10 episodes, for all of the environments is presented in Tab. 2 that empirically confirms that both: (1) introducing of

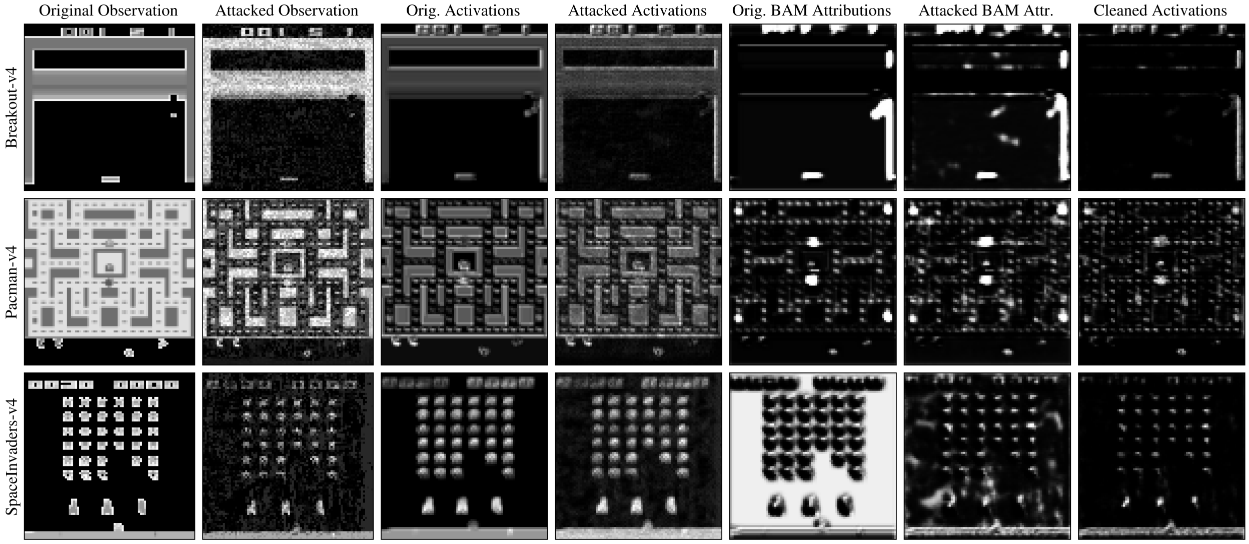

Figure 2. Qualitative sampled results of the proposed activation recovery process. All states were attacked with Linear PGD-method [7] with an ϵ of 0.1. For the observations, only the last channel is displayed (corresponding to the most frequent time step). All activations and attention are averaged over channels. While the attacks are very noticeable both in the activations and attention maps, the cleaned map removes a significant of the perturbations while retaining all necessary information for decision making.

BAM layer to architecture and (2) incorporation of adversarial samples during training do not lead to dropping of the performance.

3.2. Recovery algorithm

We want to describe the algorithm we use to ”recover” the intermediate activations with the learned attention maps. ψ_θ denotes the `BAM-CNN-Adv.` trained model (see Sec. 3.1), parameterized by θ with N layers with a BAM layer placed at index l. Adversarial examples $s_A = \mathcal{A}(\psi_\theta, s, \epsilon)$ for original states s are created by an attack algorithm $\mathcal{A}$, where ϵ controlling the severity of the perturbation. The goal of the proposed algorithm is to recover a policy $\tilde{\pi}(a|s_A)$ that is close to the original policy $\pi^*(a|s)$ induced by ψ_θ. The policies' similarity metrics are discussed in Sec. 4. We present the pseudo-code of the inference-time defense algorithm in Alg. 1. Our analysis shows that this particular reinforcement learning scenario produces highly polarized attention maps (i.e., very high contrast between high- and low-attention areas). Therefore, the learned BAM maps have a strong notion of what is important for decision-making. As a result, applying element-wise (Hadamard) multiplication allows us to ”prune” part of the input that is not significant for the decision-making but still affected by adversarial perturbations.

Algorithm 1 Adversarial Attack recovery

```
  Input: $\psi_\theta$
  Output: $\tilde{\pi}(a|s_A)$ that is close to $\pi^*(a|s)$
  $f^{(l-1)} \leftarrow \psi_\theta^{[1:l-1]}(s_A)$   ▷ For notation clarification see [1].
  $f^{BAM} \leftarrow \psi_\theta^{[1:l]}(s_A)$
  $f_{rec.}^{(l-1)} \leftarrow f^{(l-1)} \odot f^{BAM}$
  $\tilde{\pi}(a|s_A) = \psi_\theta^{[l:N]}(f_{rec.}^{(l-1)})$
```

4. Results

	Breakout-v4	MsPacman-v4	SpaceInvaders-v4
Existing Best	$1.5_{\pm 0.9}$	$504.0_{\pm 158.25}$	$166.5_{\pm 74.8}$
BAM-CNN-Adv.+Recovery	$6.3_{\pm 6.84}$	$687.0_{\pm 242.0}$	$170.0_{\pm 107.5}$
Difference	×4.2	×1.36	×1.02

Table 1. Comparison between Baseline and BamCNN+Recovery: We compare average rewards of best baseline method (see Tab. 2, e.g.`Nature-CNN-Adv.`) compared to our attention-map based defense method (`BAM-CNN-Adv.+Recovery`).

Tab. 1 and Tab. 2 summarize the reward performance of different models. Tab. 2 e.g. shows that adversarial training can improve performance on non-attacked (default) environments. We hypothesize, that the limited adversarial attacks can act as a type of regularization/data augmentation. However, as mentioned above, adversarial training generally does not lead to huge robustness gains, i.e. the reward for attacked environments is noticeably worse in any case. Tab. 1 summarizes the performance gains of our method against the strongest baseline method. Besides reward performance, we evaluate our method by measuring the following metrics: (a) The percentage of fully reversed attacks (`Reversed-TOP-1`), for which the action predicted based on the cleaned activations matches the original action based on the non-attacked state. (b) The percentage of par-

[1] $\psi_\theta^{[i:j]}(x)$ notation means slicing ψ_θ from layer with index i to layer with index j and apply to tensor x.

	Breakout-v4				MsPacman-v4				SpaceInvaders-v4			
	Nat.-CNN	BAM-CNN	Nat.-CNN-Adv.	BAM-CNN-Adv.	Nat.-CNN	BAM-CNN	Nat.-CNN-Adv.	BAM-CNN-Adv.	Nat.-CNN	BAM-CNN	Nat.-CNN-Adv.	BAM-CNN-Adv.
Default env.	$166.2_{\pm 105.3}$	$59.0_{\pm 15.5}$	$98.4_{\pm 82.2}$	$96.8_{\pm 38.1}$	$1736.0_{\pm 98.5}$	$2147.0_{\pm 59.7}$	$1802.0_{\pm 230.4}$	$1826.0_{\pm 47.4}$	$700.5_{\pm 217.9}$	$815.0_{\pm 217.9}$	$737.0_{\pm 350.2}$	$734.5_{\pm 276.8}$
Attacked env.	$0.4_{\pm 1.2}$	$0.5_{\pm 1.5}$	$0.2_{\pm 0.6}$	$1.5_{\pm 0.9}$	$504.0_{\pm 158.25}$	$70.0_{\pm 0.0}$	$297.0_{\pm 150.4}$	$253.0_{\pm 131.4}$	$29.5_{\pm 24.5}$	$33.0_{\pm 14.3}$	$166.5_{\pm 74.8}$	$42.0_{\pm 21.9}$

Table 2. Cumulative reward results of baselines: `Nature-CNN`, `BAM-CNN`, `Nature-CNN-Adv.`, `BAM-CNN-Adv.` for `Breakout-v4`, `SpaceInvaders-v4`, `MsPacman-v4` averaged over 10 episodes with standard deviations. For the adv. attacks, each frame is attacked with an $\epsilon = 0.05$.. We contrast these scores with the reward achieved by our proposed method (see Tab. 1).

	Breakout-v4				MsPacman-v4				SpaceInvaders-v4			
	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.5$
% Successful Attacks	66.22	93.42	95.76	84.54	74.0	100.0	100.0	100.0	94.0	100.0	100.0	100.0
Reversed-TOP-1	57.78	35.51	32.73	15.45	30.73	22.44	19.74	0.0	13.67	2.6	0.045	0.0
Reversed-TOP-2	89.53	76.59	66.36	66.10	50.62	44.15	34.627	5.6	40.15	17.77	7.2	0.24
Reversed-ANY	65.01	49.75	51.41	15.45	83.25	70.39	70.54	0.35	41.24	35.59	31.25	0.24

Table 3. Attack recovery results with different ϵ values for `Breakout-v4`, `SpaceInvaders-v4`, `MsPacman-v4`, averaged over 10 games each. The used architecture was trained on partly attacked data during training (with every 10th frame perturbed), which only resulted in marginal adversarial robustness. This adversarial training by itself is not able to effectively defend against adversarial attacks: As is visible in the first line, adversarial attacks manage to switch the selected action in most cases, even for small ϵ. Applying the presented recovery technique can decrease the impact of attacks up to a certain level.

tially reversed attack (`Reversed-TOP-2`) when the recovery leads to the predicted action matching either the first or second choice of the original action. (c) The percentage of partially reversed attack (`Reversed-ANY`) when the recovered action is different from the attacked action (or already the original action assigned maximum probability). We also report the (d) percentage of successful PGD attacks (`% Successful Attacks`), i.e., how often the attack is able to switch the maximum probability action. The results for different ϵ across different environments can be found in Tab. 3. Figure 2 shows qualitative visual results of the "recovery" technique. The adversarial attack targets the entire frame, which is clearly visible in the activations. The recovered activation shows how a significant part of the applied perturbations is removed from the input image. As a result, the output action distribution after the "recovery" procedure is closer to the original action distribution (see Tab. 3). For example, in `Breakout-v4` for moderate $\epsilon = \{0.01, 0.05\}$, the original action is restored with 57.78% and 35.51% probability, respectively (out of 4 total actions). For `MsPacman-v4`, for approx. 30% of states the original action is restored (out of 9 possible actions). This recovery is noticeable in the final performance of the trained agents attacked with adversarial observations. We see improvements that scale with the success rate of the defense.

5. Discussion & Future work

A great strength of the presented method is it's simplicity and low application cost. The additional BAM-layer and clean-up process during inference only adds marginal complexity in terms of model parameters and inference time. BAM itself is suitable for any type of CNN architecture, but we might explore other types of attention (e.g., self-attention) for other architectures like vision transformers [3]. Another potential benefit is the use of selective attention to increase robustness against other types of distractors like background images. While the initial results showcase the potential of the method, the performance of agents is still worse in the attacked environments. It is, therefore, necessary to explore how to improve the method further. Another shortcoming is the fact, that we applied the presented defense in a grey-box setting, in which the attacker is not aware of the inference procedure. To further investigate robustness, we should explore adaptive attack procedures [15] with the attacker having full knowledge of the attack. Furthermore, a comparison to other defense methods [5] is warranted.

Besides the aforementioned improved evaluation, we plan to extend our method in follow-up work: (a) Investigating its utility for core computer vision problems and different backbones that do not share characteristics of reinforcement learning and the investigated environments. (b) Investigate different "recovery" procedures on feature maps, e.g., using different operations besides element-wise multiplication. (c) Use multiple BAM attention layers in a network and apply the clean-up- procedure multiple times, which might increase robustness, (d) Investigate the possibility of application of fewer amount of attacks during training. (e) Explore different types of adversarial attack methods. (f) Extend the current method to vector-based RL and continuous action spaces.

6. Conclusion

In this paper, we presented a simple adversarial defense method for deep RL based on *Bottleneck Attention*. This type of selective attention could be a general tool to restrict the effect on non-targeted adversarial perturbations. We presented initial promising results that demonstrate the potential of the method to recover activations perturbed by adversarial attacks. In future work, we plan to apply a more thorough evaluation and study of potential improvements.

References

[1] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013. 1, 2

[2] Jinyin Chen, Xueke Wang, Yan Zhang, Haibin Zheng, and Shouling Ji. Attention mechanism based adversarial attack against deep reinforcement learning. In *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, pages 19–43, Cham, 2021. Springer International Publishing. 1

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv e-prints*, page arXiv:2010.11929, Oct. 2020. 4

[4] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, and et al. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, pages 1–38, 2022. 1

[5] Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 3(2):90–109, 2021. 1, 2, 4

[6] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017. 1

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 2, 3

[8] Volodymyr Mnih, Koray Kavukcuoglu, and et al. Silver, David. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015. 1, 2

[9] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. BAM: bottleneck attention module. *CoRR*, abs/1807.06514, 2018. 1, 2

[10] You Qiaoben, Xinning Zhou, Chengyang Ying, and Jun Zhu. Strategically-timed state-observation attacks on deep reinforcement learning agents. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. 1

[11] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. 2

[12] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*, 2017. 2

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[14] Yujin Tang, Duong Nguyen, and David Ha. Neuroevolution of self-interpretable agents. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, page 414–424, New York, NY, USA, 2020. Association for Computing Machinery. 2

[15] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *CoRR*, abs/2002.08347, 2020. 4