A Classification Based Approach to Identifying and Mitigating Adversarial Behaviours in Deep Reinforcement Learning Agents

Seán Caulfield Curley
University of Galway
Ireland
s.caulfieldcurley1@nuigalway.ie

Karl Mason
University of Galway
Ireland
karl.mason@universityofgalway.ie

Patrick Mannion
University of Galway
Ireland
patrick.mannion@universityofgalway.ie

ABSTRACT

Opponent modelling is an area of significant interest in multi-agent systems (MAS). It has been shown that an agent can be trained with an adversarial policy which achieves high degrees of success against a state-of-the-art DRL victim despite taking unintuitive actions. This prompts the question: is this adversarial behaviour detectable through the observations of the victim alone? We find that widely used classification methods such as random forests are only able to achieve a maximum of $\approx 71\%$ test set accuracy when classifying an agent for a single timestep. However, when the classifier inputs are treated as time-series data, test set classification accuracy is increased significantly to $\approx 98\%$. This is true for both classification of episodes as a whole, and for "live" classification at each timestep in an episode. These classifications can then be used to "react" to incoming attacks and increase the overall win rate against Adversarial opponents by approximately 17%. Classification of the victim's own internal activations in response to the adversary is shown to achieve similarly impressive accuracy while also offering advantages like increased transferability to other domains.

KEYWORDS

Adversarial Reinforcement Learning, Deep Reinforcement Learning, Opponent Modelling

1 INTRODUCTION

In competitive environments, being able to reason about an opponent's past behaviour and using that reasoning to predict what they might do in the future is a crucial technique for success. However, if an opponent's actions seem nonsensical and yet they still win regularly, it is difficult to predict what they will do in the future because we don't even understand what they are doing now. This is the conundrum experienced by agents facing adversarial perturbations in opponent policies which are designed to confuse the victim agent to the point of defeat. Adversarial examples are commonly used in image classifiers to throw off models' predictions or even force them to predict some specific class [9, 13].

Naturally generated adversarial observations were first illustrated by Gleave et al. [6] in the 3D simulated physics environments created by Bansal et al. [2]. Gleave et al. [6] demonstrated that an adversarial agent could learn to reliably beat their victim despite taking apparently random actions. It was also shown that if the victim was made blind to the adversary's movements, the victim would become immune to the adversarial strategy. This suggested that the victim's own observations of the adversary were its downfall.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2023), Cruz, Hayes, Wang, Yates (eds.), May 9-10, 2023, Online, https://ala2023.github.io/. 2023.

This prompted the question: can agents learn to detect adversarial behaviour before it conquers them? The results we report in this paper demonstrate that widely-used supervised learning models can classify both normal and adversarial behaviour with very high degrees of accuracy. We also show that adversarial policies create highly "unusual" activations in the victim agent's neural network. These unusual activations can also be successfully used to classify opponent behaviour. When using a long short-term memory (LSTM) model for classification, the trained model can also be adapted for live classification at each time step in an episode. Despite the LSTM model being trained on entire episodes with a single output label, after training it can correctly classify timestep-by-timestep testing data with a high degree of accuracy after only $\approx 35\%$ of the episode's length. This LSTM allows the "victim" agent to greatly increase its robustness to adversarial attacks and hence its win rate.

The contributions of this work are as follows:

(1) We demonstrate the effectiveness of using a victim agent's raw observations to determine whether an opponent has specifically been trained to act in an adversarial manner. Game state encodings have been used many times in previous work to classify opponent behaviour, but to the authors' knowledge this is the first use of raw observations for opponent classification in a 3D simulated physics environment.
(2) Our work also shows that the activations induced in an agent during deployment may also be useful for reasoning about the behaviour of an opponent in the environment. We found that classifying opponents worked just as well with activations as input data, as it did with raw observations. Again, to the best of the authors' knowledge, this is the first time that the usefulness of activations for opponent classification has been demonstrated in a published paper. This approach could potentially be useful in multi-task or transfer learning settings in future work, as classifiers for opponent behaviour could be trained to classify using agent-specific features (neural network activations) rather than domain-specific features (observations), thereby improving transferability.
(3) Our final contribution is the implementation of the classifier within the "victim" agent to increase its robustness. The victim's win rate increases dramatically and visual inspection of evaluation episodes shows that victim effectively ignores the adversarial attacks.

Section 2 of this paper outlines the motivation for this work and gives an insight into previous related studies in adversarial attacks and opponent modelling. The techniques used to generate the datasets for classification are explained along with the model

parameters used in Section 3. The results from these classification approaches are given in the next section. Section 4 also highlights an area where our approach could potentially be improved and outlines how an agent that uses our classifiers to "react" performs. Section 5 concludes the paper and gives a brief outline of the plans to extend this work.

2 BACKGROUND & RELATED WORK

Generally in adversarial example generation, perturbations are added directly to the inputs of a machine learning model which results in unintended or undesirable behaviour at the output. Szegedy et al. [13] were the first to show that making seemingly minor changes to images caused them to be misclassified by state-of-the-art image classification models. It was shown by Papernot et al. [9] that a specific target label could be predicted with fine-tuning of the perturbation. Huang et al. [7] applied these techniques to reinforcement learning by altering the network's input of the last 4 images from Atari games.

The foundation of this paper will be a follow on to the work of Gleave et al. [6] who demonstrated that an agent in a multi-agent environment can induce *natural* adversarial observations which significantly affect the performance of its opposing agent. Rather than directly modifying the "victim" agent's observations, the attacker was trained to take actions which induced abnormal activations in the victim, causing it to perform poorly. Lin et al. [8] also applied adversarial techniques to multi-agent systems, focusing on a single victim agent in a Starcraft 2 team. In both studies, the attackers trick the victim agent into taking sub-optimal actions by naturally creating adversarial observations. To date, the only improvement on Gleave's adversaries was made by giving the attacker access to the actions and observations of the victim [17]. This access allowed the attacker to target specific features of the victim's observations to induce maximum distance from the optimal policy. None of the above papers involved any modelling of the adversarial agent.

There have also been studies attacking agents in simulated real-world MAS using another agent. Pierpaoli et al. [10] showed that autonomous UAVs can be forced to fly in a path determined by a third party by using other UAVs to continuously threaten collision. Behzadan and Munir [3] outlined how autonomous vehicle agents can be trained to find an "optimal" policy leading to direct collision with another vehicle. Wachi [15] demonstrated that adversarial autonomous vehicles can be trained to induce failures (accidents or long delays) in victim autonomous vehicles without making any contact.

Opponent modelling is a popular area of research in which researchers aim to determine some properties of other agents in a multi-agent environment [1]. However, most studies of opponent classification rely on encodings of the game state to model opponent behaviour. For example, Weber and Mateas [16] predicted the opponent's strategy in Starcraft 2 using a feature vector of the timestamps of unit production. Spronck and Teuling [11] used 25 features including number of cities, number of units and population size to model the "preferences" of players in Civilization IV. A significant amount of opponent modelling research has focused on the domain of robot soccer. This is an environment with continuous states and actions similar to ours but studies have to-date inputted a representative feature vector rather than the raw game state [4, 12]. To the authors' knowledge, this is the first paper to classify opponent behaviour solely using the raw observations of the environment and of the other agent. It is also the first to use an agent's own activations as input to a classifier for opponent behaviour.

3 METHODOLOGY

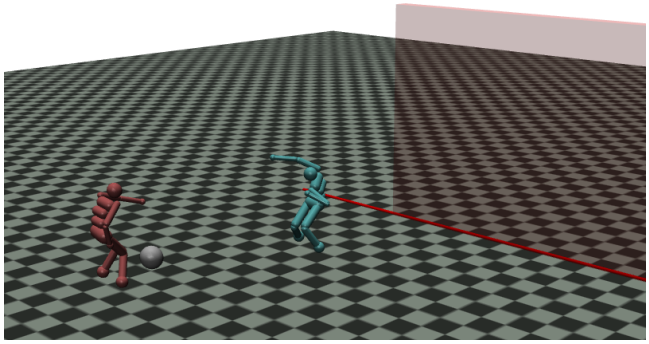

Figure 1: The *KickAndDefend* environment from Bansal et al. [2] which the adversarial agents were evaluated in.

All experiments in this paper were performed in the zero-sum multi-agent competition environments created by Bansal et al. [2]. We consider two types of classifier inputs: activations and observations. In the remainder of this paper, those terms will be used to represent the following:

(1) **Observations:** A vector with 384 entries, comprising the proprioceptive observations of the adversarial agent's joint angles/velocities/positions, the ball's positioning and other values such as actuator forces
(2) **Activations:** The outputs from each of the 128 nodes of the victim agent's trained neural network when an observation is fed into the input layer

Activations and observations were generated by simulating 500 episodes with already trained agents in the *KickAndDefend* environment (Figure 1) and outputting the "victim" agent's observation and the resulting activation at every timestep. Both Adversarial and regular (Zoo) goalkeeper agents were used. This was repeated for 3 different seeds per agent leading to $3victims \times 3attackers \times 500episodes = 4500episodes$ for both the Adversarial and Zoo cases. The pre-trained Zoo agent models are those provided by Bansal et al. [2], while the pre-trained adversarial agents are those provided by Gleave et al. [6]. In all cases, the goalkeeper is the opponent agent (which can be Adversarial or Zoo) and the victim agent is the "kicker".

3.1 Instantaneous Classification

Each timestep's activations for an episode were labelled according to the type of opponent that induced them (Adversarial or Zoo). The activations dataset was split into a training set 70% of the size of the original set and validation and test sets comprising 15% of the original set each.

A number of classification algorithms were evaluated; namely random forest, k-nearest neighbours, Gaussian naive Bayes and logistic regression. Grid search cross validation was performed on each model to determine their most effective hyperparameters and to ensure a fair comparison between models.

3.2 Time-series Classification

Activations/observations were grouped by episode and each episode (or time-series) was given a single label corresponding to the opponent agent's behaviour. Episodes were padded to the maximum length of 500 timesteps and feature values were normalised to the range $[0, 1]$. Normalising was particularly important for observations as the ranges of different features varied widely e.g., angles between joints could fall in the range $[-120, 120]$ while the root coordinates were bounded by the range $[-1, 1]$.

Both activations and observations were classified using a network comprised of a masking layer, an LSTM layer of 100 units and a single dense output layer. The default Keras hyperparameter values were used throughout this proof-of-concept experiment as they displayed promising results immediately. The output was passed through a sigmoid activation layer to achieve binary classification. Both networks were trained for 10 epochs.

One point illustrated by Gleave et al. [6] is that the adversarial activations were dispersed evenly around the parameter space, in and amongst both Zoo and Random activations. Furthermore, visually analysing the episodes (Appendix A) makes it clear that the differences between an agent acting randomly and one acting adversarially are imperceptible to the human eye. Therefore, testing sets consisting of activations or observations of an agent acting randomly were generated to analyse the impact of random activations and observed behaviours on the performance of the classifier.

Finally, although training will involve classification of an entire episode, the goal of the final implementation will be to classify the opponent's behaviour live, during an episode. Thus, an experiment was performed where timesteps were inputted one by one with the classifier predicting its output at each step. Essentially, each sequence of timesteps until the last was then treated as its own sub-episode.

4 RESULTS

4.1 Instantaneous Classification

Table 1 outlines the 5-fold accuracies achieved by the each of the classification algorithms tested in this study. For each algorithm, the final model used was that which attained the highest test set accuracy after grid search cross validation of the hyperparameters. Evidently, no model was able to achieve a particularly high testing set accuracy. The random forest algorithm does show some promise with approximately 71% accuracy but a confusion matrix of its predictions (Figure 3) shows why it is still unsuitable for the task.

In Section 4.4, the method for defending against adversarial behaviour upon classification, "Masking", is outlined. This is the same method as was used by Gleave et al. [6] where the victim pretends the goalkeeper is fixed in the starting position and ignores its adversarial actions. The high rate of false Adversarial predictions mean that the agent would regularly ignore the actions of a "normal" goalkeeper. In fact, Gleave et al. [6] studied the effects of unnecessary masking and found that "masked victims do worse than a normal victim when playing normal opponents" [6]. False Zoo predictions are not as impactful in the sense that if the goalkeeper is acting adversarially and the victim does something other than masking, it will likely lose regardless.

Figure 4 provides insight into why the instantaneous classifiers are unable to correctly separate the data. There is significant overlap between both Adversarial and Zoo activations (when converted to 2 dimensions using t-SNE [14]) implying that attacks simulate "normal" behaviour with small tweaks to induce sub-optimal actions in the victim.

4.2 Time-Series Classification

Conversion of both the activations and observations into time-series data led to significant improvement in the classification accuracy.

4.2.1 Activations. After 10 epochs of training, the LSTM model achieved 98.0% accuracy on the test set. While this result is impressive, as mentioned in Section 3.2, classifying the entire episode is not the goal of this study. Instead, we want to achieve in-episode classification of the opponent so that the victim agent can react to and defend against attacks. For live classification, a list of all of the timesteps' activations up to that point is passed in and a prediction between 0 (Adversarial) and 1 (Zoo) is output. A batch of the last 35 predictions is then averaged and the result is classified. If the average of the predictions gets under 0.05, the model predicts that the opponent is acting adversarially. Conversely, if the average exceeds 0.8, Zoo behaviour is predicted. These thresholds and the 35 prediction size of the batch were chosen as they were found to achieve good accuracy while also forming predictions relatively quickly. This is illustrated in Table 2, where an upper threshold of 0.7 amasses the most correct predictions at all checkpoints throughout the episode except for the final one. However, an upper threshold of 0.8 achieves only slightly less correct classifications in the early checkpoints while making almost no incorrect predictions. As incorrect predictions are likely to be much more harmful than slightly late predictions, the higher threshold is used.

Despite not being trained for live classification, the model performed well and attained 97.6% overall accuracy. Table 3 shows that even when the model does not predict its opponent correctly, it is rarely confidently incorrect. In fact, the 0.8% of incorrect Zoo predictions only amounts to a single prediction, meaning the model was only confidently incorrect for 1 of the 250 testing samples. Here, indecisions denote episodes where the averaged predictions never got outside of the range $[0.05, 0.8]$ within the maximum 500 timesteps. One possible cause of these indecisions is that some episodes are extremely short, ending in less than than 40 timesteps (3 of the 5 indecisions were in sub-40 timestep episodes). It would be unsurprising if an agent could not form a confident prediction in so few timesteps, especially considering the median length of the Adversarial and Zoo episodes are 272.5 timesteps and 171.5 timesteps, respectively (Table 4).

The time taken to reach correct predictions is another promising result from this method of live classification. Classifying an adversarial opponent's behaviour after only about 35% of an episode should mean that there is more than enough time to defend oneself appropriately and go on have a good chance to win the episode.

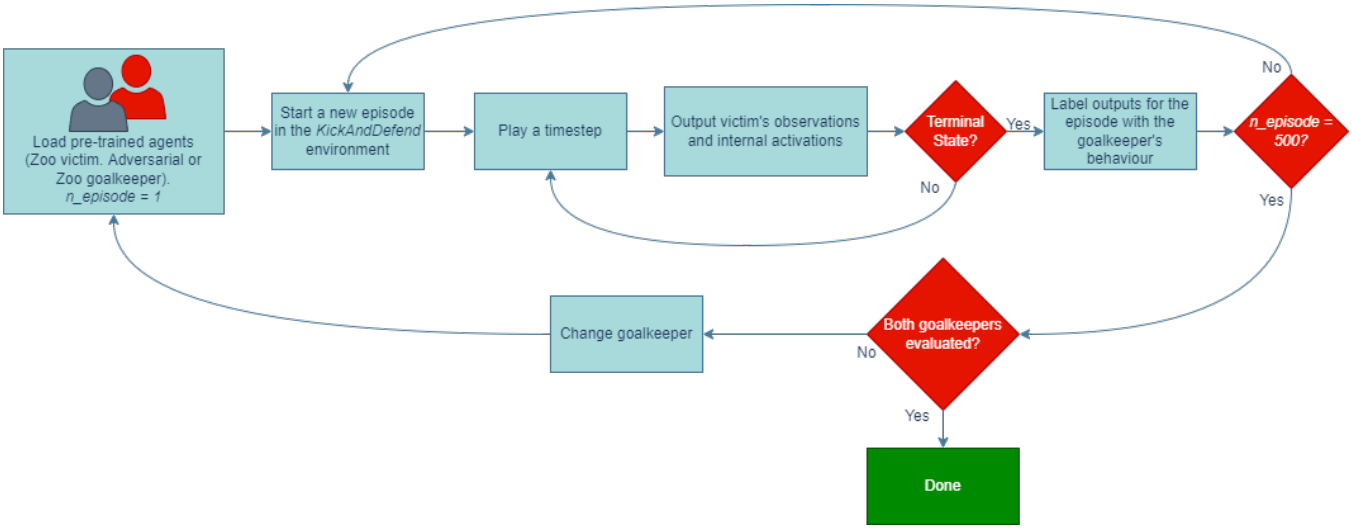

Figure 2: The evaluation process for creating the observation and activation datasets

Table 1: The 5-fold accuracies of each of the tested instantaneous classification methods.

Algorithm	Testing Set Accuracy	Training Set Accuracy
Random Forest	0.711 ± 0.002	0.999 ± 0
Logistic Regression	0.643 ± 0.003	0.643 ± 0.001
Gaussian Naive Bayes	0.568 ± 0.002	0.569 ± 0.001
k-Nearest Neighbours	0.646 ± 0.002	0.792 ± 0.001

Table 2: Number of correct and incorrect predictions at a number of checkpoints throughout 250 episodes. The Upper Threshold column describes the threshold which must be exceeded to predict Zoo behaviour while the lower threshold (Adversarial) is fixed at 0.05

	Upper Threshold	No. predictions at episode checkpoint				
		20%	40%	60%	80%	100%
Corrects	0.7	30	163	208	236	239
	0.8	16	109	166	203	244
	0.9	13	86	118	135	194
Incorrects	0.7	5	6	6	7	7
	0.8	0	0	0	1	1
	0.9	0	0	0	1	1

Table 3: Results of live classification of activations

True Label	Correct	Incorrect	Indecision	Avg. % of Episode to Correct Prediction	Avg. % of Episode to Incorrect Prediction
Adversarial	98.4%	0%	1.6%	35.9%	N/A
Zoo	96.7%	0.8%	2.4%	64.2%	76.4%

Furthermore, it suggests that there is no one "killer blow" which the goalkeeper performs to quickly end the episode. Instead, adversarial behaviour is detectable at an early stage long before it "successfully" disrupts the victim agent. This is confirmed in Section 4.4.

4.2.2 Observations. The same experiments as in Section 4.2.1 were also performed on the victim's raw observations. Again, the model learned the relationship well, this time achieving a test set accuracy of 98.4%. Similarly, the results of live classification presented in

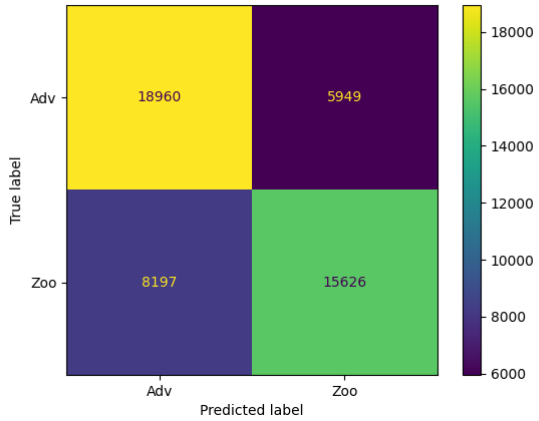

Figure 3: Random forest confusion matrix

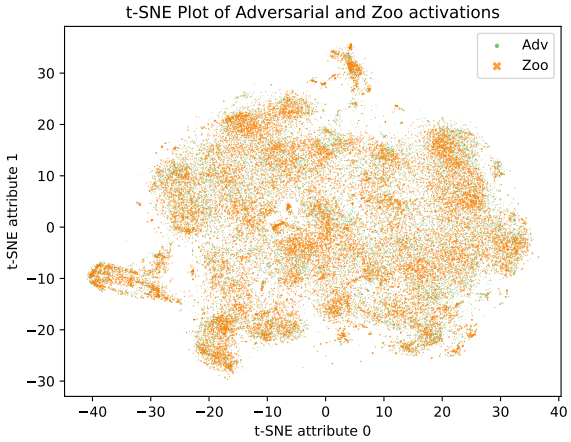

Figure 4: Adversarial and Zoo activations

Table 4: The distribution of episode lengths in the dataset. The 1-timestep long episodes comprise the kicker agent falling over immediately. Other than these, the shortest Adversarial and Zoo episodes are 5 and 7 timesteps, respectively.

Label	Shortest	Longest	Mean	Median
Adv	1	500	296.096	272.5
Zoo	1	500	183.728	171.5

Table 5 indicate that observation classification is as good or even better than activation classification. This is especially true with the absence of any confidently incorrect predictions. This could be due to the observations having more features (384 compared to the activations' 128). The Zoo prediction threshold was increased in this model from 0.8 to 0.975 as it was found through experimentation

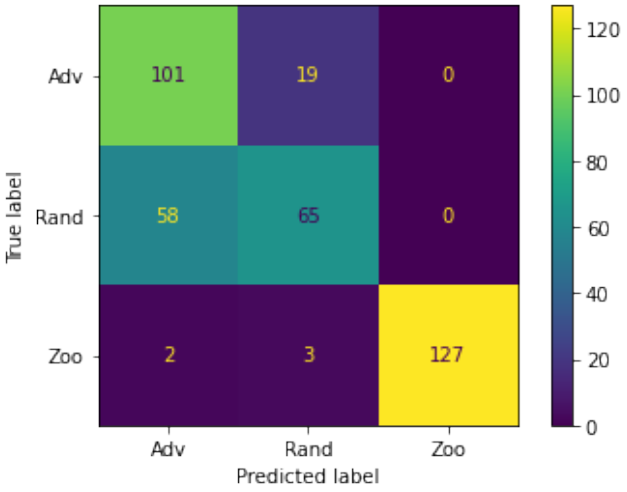

Figure 5: Multi-class Confusion Matrix for activations

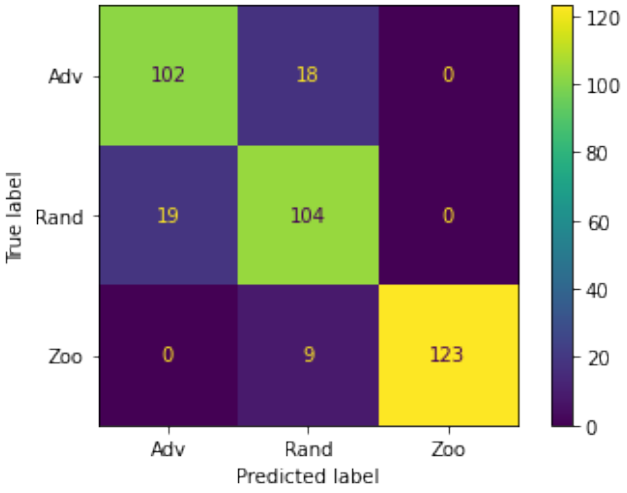

Figure 6: Multi-class Confusion Matrix for observations

to achieve a good balance between correct classification and timely predictions.

In order to test the models on random inputs, a normal agent played against a randomly acting goalkeeper for 500 episodes. The models were re-trained including the random inputs for 20 epochs and the results of classifying a held-out set are shown in Figures 5 and 6. Both the model trained on activations and the model trained on observations accurately separate Zoo ("normal") behaviour from random and adversarial behaviour. However, both models then struggle to distinguish between an opponent acting adversarially and an opponent acting randomly. The activations classifier is particularly poor, only achieving 68% accuracy on classification of the two abnormal behaviours. This is a potential avenue for improvement, however random inputs are not likely to be encountered very often and thus are not especially relevant. Firstly, the chances of

Table 5: Results of live classification of observations

True Label	Correct	Incorrect	Indecision	Avg. % of Episode to Correct Prediction	Avg. % of Episode to Incorrect Prediction
Adversarial	96.88%	0%	3.13%	30.37%	N/A
Zoo	98.36%	0%	1.64%	35.94%	N/A

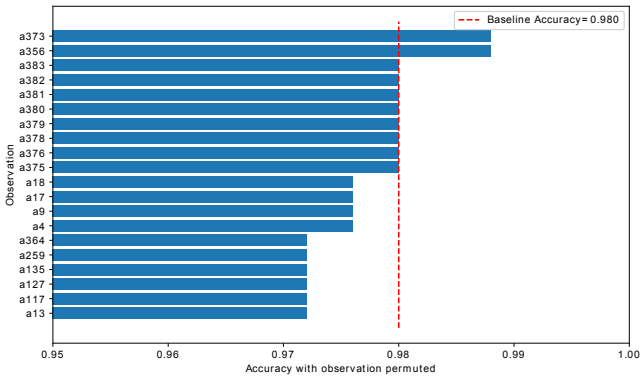

Figure 7: The most extreme changes in accuracy caused by permuting each observation (10 most and least affected). Note the y-label corresponds to the observation index from the list of 384 observations returned by the victim.

an opponent acting randomly in the first place is highly unlikely whether that is in a simulated or real-life environment. A trained agent will seek to maximise its reward which random behaviour is highly unlikely to achieve while a real person has no reason to act randomly. Most importantly though, false identification of adversarial behaviour when the opponent is acting randomly should not adversely affect our agent's performance. The approach of masking observations upon detection of adversarial behaviour means the victim will simply ignore the random opponent and likely win the episode regardless. Therefore, it is proposed to only train on and classify Adversarial and Zoo behaviour in future applications.

4.2.3 Feature Importance. Permutation feature importance [5] was used to calculate what effect each feature had on the overall accuracy . For each feature, in every episode the values for that feature were shuffled. The model then classified these new episodes and the overall accuracy was calculated. If shuffling a feature decreases the accuracy significantly, the feature is important because the model depends on it. Conversely if the accuracy increases from shuffling a feature, that means it is not at all important for classification. This results of this analysis are only presented for the observations case as the insights to be gained from a result like "Activation 42 of 128 is very impactful" are minimal.

Figure 7 outlines the 10 most and least important features. The overall changes to accuracy are relatively small; the biggest decrease in accuracy is $\approx$ 1% less than the baseline while the least important gives a $\approx$ 1% increase to the accuracy. However, there are still some interesting insights to be gained. For example, the fact that shuffling some observations increases the overall accuracy

Table 6: Prediction results of in-agent live classification of observations and activations

	True Label	Correct	Incorrect	Unsure
Observ-ations	Adversarial	90.42%	0%	9.57%
	Zoo	98.11%	0%	1.88%
Activ-ations	Adversarial	92.4%	4.8%	2.8%
	Zoo	95.3%	2.6%	1.9%

suggests that there are features which actively make classification more difficult. Therefore, feature selection may offer equal or even improved performance while reducing the model size. Some interesting observations which the model deems important include a13 (*right_knee* rotation), a9 (*abdomen_x* rotation) and a17 (*left_knee* rotation). One intriguing inclusion is a18 (*right_shoulder_1* rotation). The normal goalkeeper almost always uses its legs to tackle the victim or to save the ball but this result implies that abnormal shoulder movement is one of the biggest factors in disrupting the victim. This could indicate that we can manipulate limbs which normally would not have much impact to achieve our adversarial behaviour.

4.3 In-Agent Classification

To enable in-agent classification, firstly the saved models from Section 4 were loaded. The maximum and minimum values encountered during training for each of the observed features were also loaded so that they could be used for normalisation of the live values. Once the episodes began, the same procedure as in Section 3 was used to classify the victim's inputs (observations or activations);

(1) Normalise the most recent timestep's features according to the maximum and minimum values observed during training
(2) Form a prediction on all of the previously seen timesteps
(3) If the average of the last n predictions exceeds the upper threshold (Zoo) or goes under the lower threshold (Adversarial), output that average as the final classification

Each of the three victim agents given by Gleave et al. [6] were evaluated against each of the three Adversarial attackers provided in their paper. Each victim was also evaluated against each of the three Zoo agents provided by [2]. In all cases, an agent was evaluated against their respective opponent for 500 episodes. This lead to $3victims \times 3attackers \times 500episodes = 4500episodes$ against each type of opponent (Adversarial or Zoo). These evaluations were performed for both the observation and activation classifying case.

The live classification model performed very well using the above method, achieving accuracies of approximately 90% and 98% on

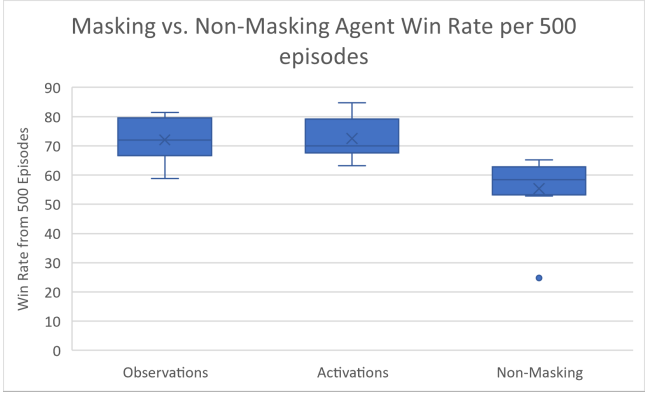

Figure 8: A boxplot of the win rates of both masking and non-masking agents. *Observations* corresponds to the victim reacting by masking based on its classification of its observations while *Activations* represents an agent reacting by masking based on its classification of its activations.

Adversarial and Zoo observations, respectively (Table 6). Furthermore, there were no incorrect predictions at all in the 9000 episodes used for evaluation. The trade-off between indecision and incorrect classification can be tweaked by changing the thresholds used for determining when the final prediction should be made. The chosen lower threshold of 0.15 and upper threshold of 0.95 were found to achieve a suitable compromise between timely predictions and high classification accuracy.

Similarly high correct prediction accuracies were observed for the activations case (Table 6). This time, a lower threshold of 0.05 and a upper threshold of 0.9 were used which lead to some incorrect guesses but also allowed for a higher correct classification rate of Adversarial inputs.

4.4 Reacting

While classification of behaviour is interesting on its own, it is more realistic that the user will want the agent to react in some way to a prediction of the opponent type. Therefore, masking (as per Gleave et al. [6]) was implemented. This involves storing the first position of the goalkeeper and using that initial value as a substitute for the actual observed position in order to negate adversarial attacks. To implement this, at the start of every episode, the goalkeeper's position was stored. Then, if Adversarial behaviour was detected, that initial position would be used as the observed position for all remaining timesteps in the episode.

Masking was tested by evaluating three different Zoo kickers against each of three Adversarial goalkeepers for 500 episodes each as in the previous section. This was done for an agent masking based on observations, an agent masking based on activations and a agent who does no masking (and "succumbs" to the adversarial attacks). Figure 8 illustrates the improvements in win rate as a result of using masking to react to attacks for each masking case for each of the 9 combinations of agents.

For the observation classifying case, the average win rate of the 9 victim agents grew from 55.3% to 72.1%. The highest win rate of an agent increased from 65.2% to 81.4%. Finally, the lowest win rate of an agent rose from 24.8% to 58.8%. Reacting to one's own activations increased the average win rate across the 9 victim agents to 72.5%. The highest win rate of the activation classifying agents was 84.8% while their lowest win rate stood at 63.2%. The new win rates are impressive and show that the classifier can achieve timely, correct predictions which can nullify the effects of Adversarial attacks. While masking is a simple fix, these results show that a more complicated defense may not be necessary to successfully defend against adversarial perturbations.

5 CONCLUSION & FUTURE WORK

This paper has demonstrated that it is possible to implement classifiers to accurately predict the behaviour of an opponent agent only using raw observations of the opponent or the victim's own internal activations. "Live" classification was also proven effective which enabled opponent modelling within simulation episodes. This lead to the average win rate of victim agents against adversaries to rise approximately 17%. Time-based models, such as the model implemented in this paper, could be used in wide range of opponent classification applications while an effective activation classifier should be transferable to many domains. One of the key considerations for the future is whether to use activations or observations for opponent classification. It is improbable that a real-world agent such as an autonomous car will have access to as fine-grained observations as the Mujoco agent does i.e., a car won't have a millimetre-level precision measurement of every angle, position, velocity, and inertia of every joint in a pedestrian's body. Therefore, classification of one's own activations appears to be more generally applicable. As activations are calculated in all agents that use neural networks to process observations, this approach could also work well in transfer learning or multi-task learning settings that require the same agent architecture to be used. On the other hand, observations are more explainable than activations. "The right elbow moving anticlockwise rapidly is the major cause of the victim's downfall" is far more intuitive than "Activation 99 in the 128 node network is the most impactful on our victim".

One avenue for further research is to investigate how best to train adversarial agents whose behaviours are more difficult to distinguish from normally expected behaviours. This could be accomplished by incorporating a penalty into the reward function of the adversarial agent each time its behaviour can be distinguished from that of a non-adversarial agent, by a classifier such as ours. This should lead to a level of deception (like "feinting" in sports) which is a more rational type of attack than the current method of contorting on the ground. A more advanced adversary could also be used to train a higher level (non-)victim which could be "fed back" again, leading to an "arms race" of improved defenders and attackers.

Adversarial behaviour has significant potential for interference with learning agents deployed in real-world applications. For example, a pedestrian agent trained to disrupt an autonomous vehicle's pedestrian path prediction model could have disastrous consequences. Therefore, it is crucial that defences against adversarial

behaviour are developed in parallel with attacks. Opponent classification approaches such as the one presented in this paper could allow victim agents to detect and react appropriately to incoming attacks. Another interesting direction following from this research is to develop methods to visualise which observations are most important to the victim agent. Live identification of the joints/limbs which the victim is paying attention to could offer great insight into why the adversarial actions are so effective.

A RELATED CODEBASES

(1) Multi-agent Competition: https://github.com/openai/multiagent-competition [2]
(2) Adversarial policies videos: https://adversarialpolicies.github.io/#videos [6]

ACKNOWLEDGMENTS

This work is supported but the National University of Galway, Ireland College of Science and Engineering Postgraduate Scholarship.

REFERENCES

[1] Stefano V. Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66–95. https://doi.org/10.1016/j.artint.2018.01.002
[2] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. 2018. Emergent Complexity via Multi-Agent Competition. *In Proc. ICLR-18* (2018).
[3] Vahid Behzadan and Arslan Munir. 2020. Adversarial Reinforcement Learning Framework for Benchmarking Collision Avoidance Mechanisms in Autonomous Vehicles. *IEEE Intelligent Transportation Systems Magazine* PP (01 2020), 1–1. https://doi.org/10.1109/MITS.2019.2898964
[4] Grazia Bombini, Nicola Di Mauro, Stefano Ferilli, and Floriana Esposito. 2010. Classifying Agent Behaviour through Relational Sequential Patterns. In *Proceedings of the 4th KES International Conference on Agent and Multi-Agent Systems: Technologies and Applications, Part I* (Gdynia, Poland) *(KES-AMSTA'10)*. Springer-Verlag, Berlin, Heidelberg, 273–282.
[5] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. 2019. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.* 20, 177 (2019), 1–81.
[6] A Gleave, M Dennis, N Kant, C Wild, S Levine, and S Russsell. 2020. Adversarial Policies: Attacking Deep Reinforcement Learning. *In Proc. ICLR-20* (2020).
[7] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial Attacks on Neural Network Policies. *arXiv preprint arXiv:1702.02284* (2017). https://doi.org/10.48550/ARXIV.1702.02284
[8] Jieyu Lin, Kristina Dzeparoska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. 2020. On the Robustness of Cooperative Multi-Agent Reinforcement Learning. In *2020 IEEE Security and Privacy Workshops (SPW)*. 62–68. https://doi.org/10.1109/SPW50608.2020.00027
[9] Nicolas Papernot, Patrick Mcdaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The Limitations of Deep Learning in Adversarial Settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)* (2016), 372–387.
[10] Pietro Pierpaoli, Magnus Egerstedt, and Amir Rahmani. 2015. Altering UAV flight path by threatening collision. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. 4A4–1–4A4–10. https://doi.org/10.1109/DASC.2015.7311414
[11] Pieter Spronck and Freek den Teuling. 2010. Player Modeling in Civilization IV. In *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Stanford, California, USA) *(AIIDE'10)*. AAAI Press, 180–185.
[12] Timo Steffens. 2003. Feature-based declarative opponent-modelling. In *Robot Soccer World Cup*. Springer, 125–136.
[13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6199
[14] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9 (11 2008), 2579–2605.
[15] Akifumi Wachi. 2019. Failure-Scenario Maker for Rule-Based Agent using Multi-agent Adversarial Reinforcement Learning and its Application to Autonomous Driving. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 6006–6012. https://doi.org/10.24963/ijcai.2019/832
[16] Ben G. Weber and Michael Mateas. 2009. A data mining approach to strategy prediction. In *2009 IEEE Symposium on Computational Intelligence and Games*. 140–147. https://doi.org/10.1109/CIG.2009.5286483
[17] Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. 2021. Adversarial Policy Training against Deep Reinforcement Learning. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 1883–1900. https://www.usenix.org/conference/usenixsecurity21/presentation/wu-xian