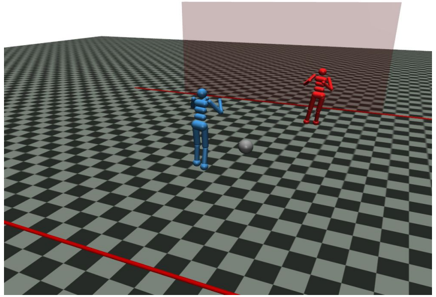
Adversarial Policies: Attacking Deep Reinforcement Learning

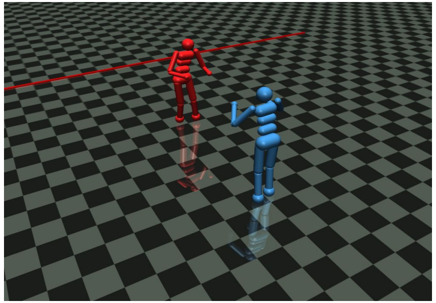
Presentation by Jarek Liesen and Lorenz Hufe

Video

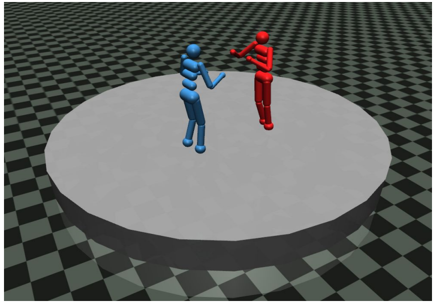
Our environments



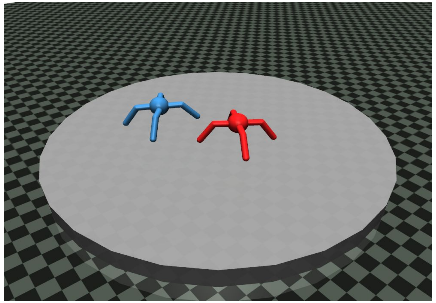
Kick and defend



You shall not pass



Sumo humans

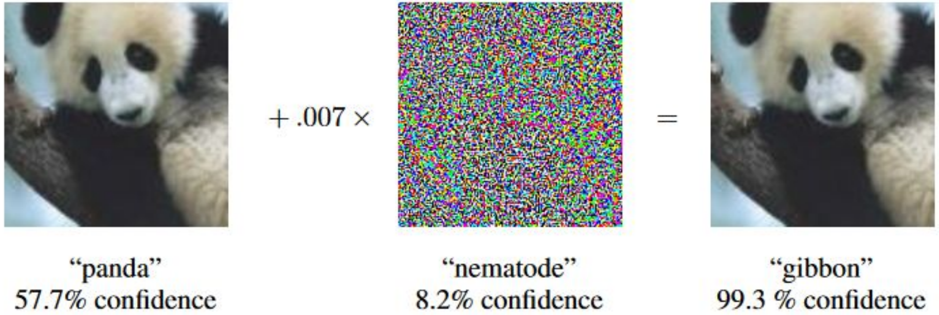


Sumo ants

- Blue and red: victim and opponent
- Observation: Each agent observes the configuration of both agents
- Action: The agents can move their joints
- Reward: Zero-sum games

Adversarial attacks

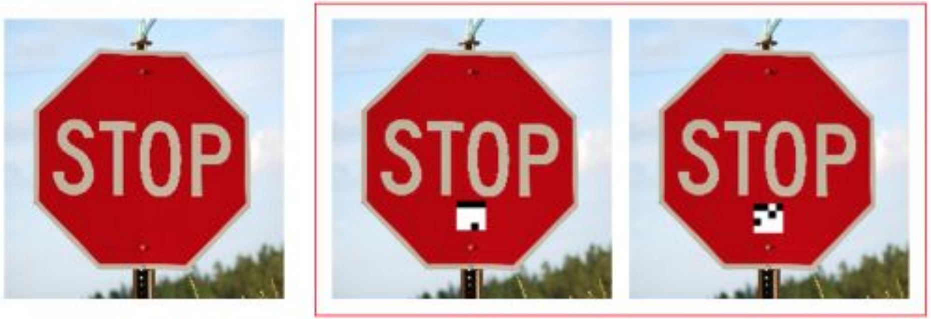
- Adversarial Attacks [2]: generate an input that maximizes the probability of an error
- How can such an attack be built?



White-box attack (weights known)	Black-box attack (weights unknown)
Derive gradient from the loss function w.r.t. the pixel and then do gradient ascend	Different methods (gradient estimation, local search, …)

Limitations of adversarial attacks

- The output of an adversarial attack is generally not observable in real world and therefore only of limited value



Stop Yield Speed Limit

(a) Normal (b) Attack

As what does the network classify this?

- We want natural observations -> Idea: adversarial policy in a multi agent environment

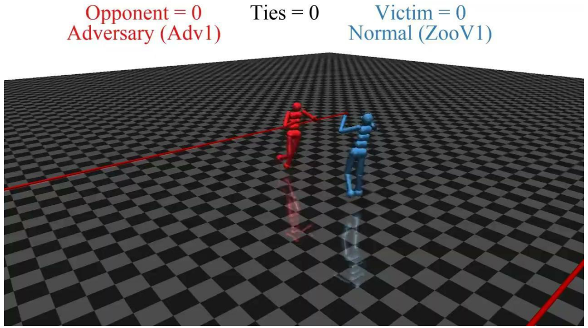
Adversarial policies

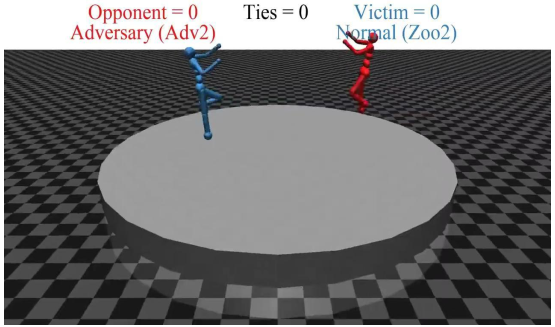
- Playing against one fixed policy turns out to be a pretty good adversarial attack!
- Fixing the other agent makes it single agent

$$\begin{gathered} R'_{\alpha}(s, a_{\alpha}, s') = R_{\alpha}(s, a_{\alpha}, a_{\nu}, s') \\ T_{\alpha}\left(s, a_{\alpha}\right) = T\left(s, a_{\alpha}, a_{\nu}\right) \end{gathered}$$ if a_v fixed

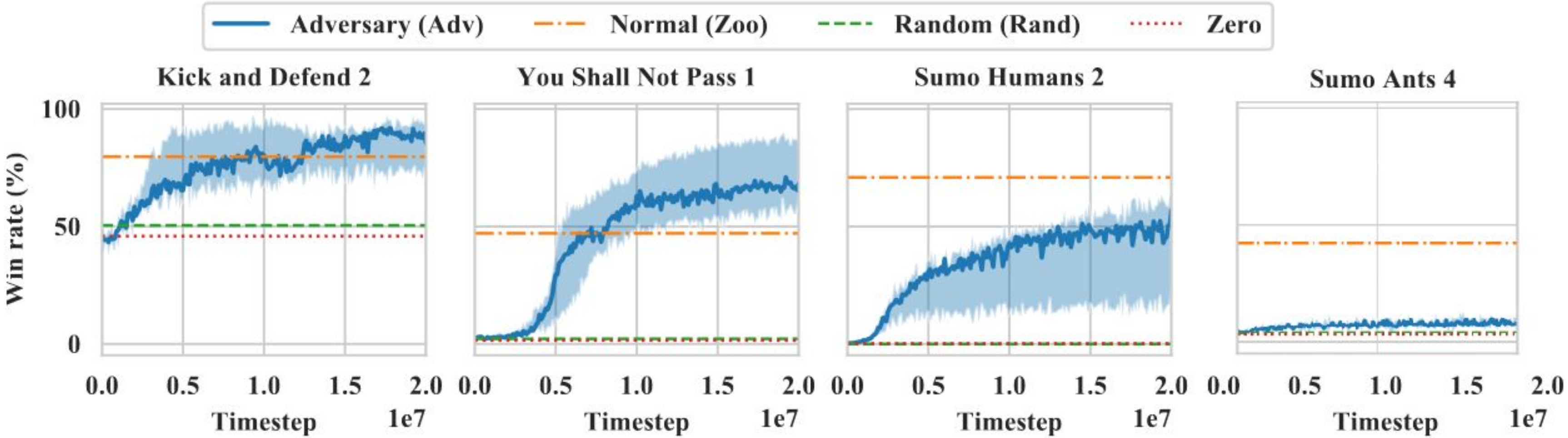
- Victim agents are pretrained agents from Bansal et al [3]
- Trained 20 Million timesteps versus fixed policy
- PPO Implementation by Stable Baselines [4]
- Sparse positive reward on win/negative on loss and ties

Result Videos





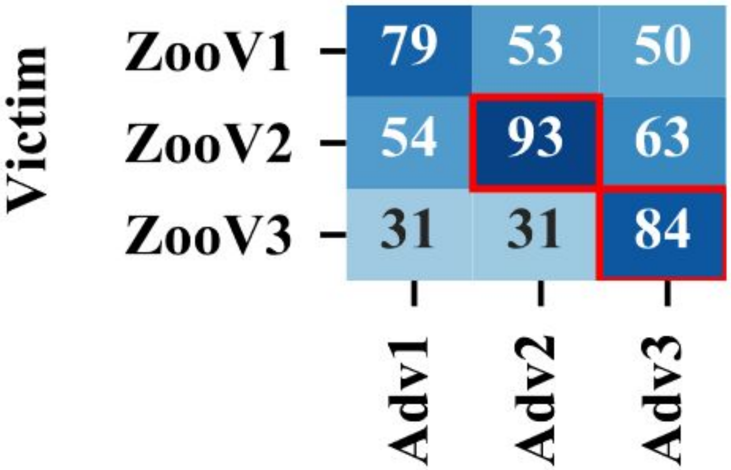
Winrate while training



Dimensionality matters:
Adversarial attack works best in high dimensional spaces
-> Bad results on low dimensional Sumo Ant task

Winrate against other agents

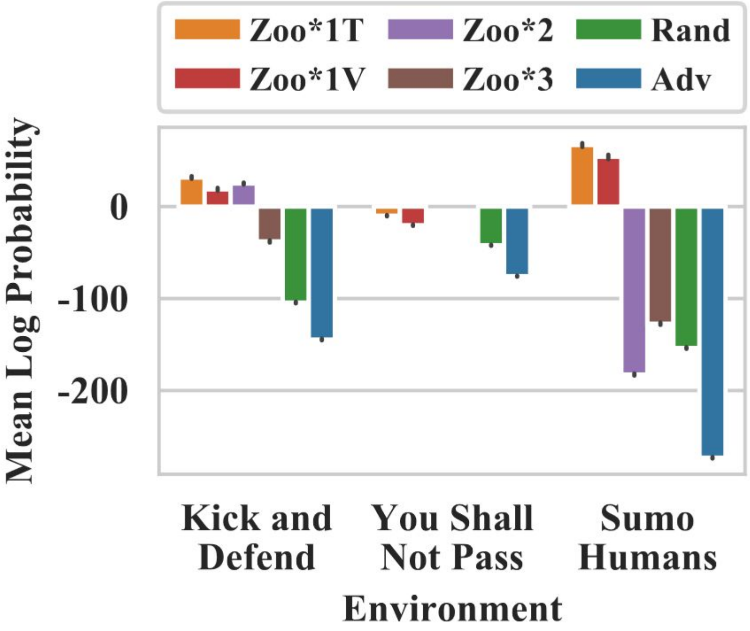
The adversarial policy performs well against it’s training partner, but does not learn a general method to win the game



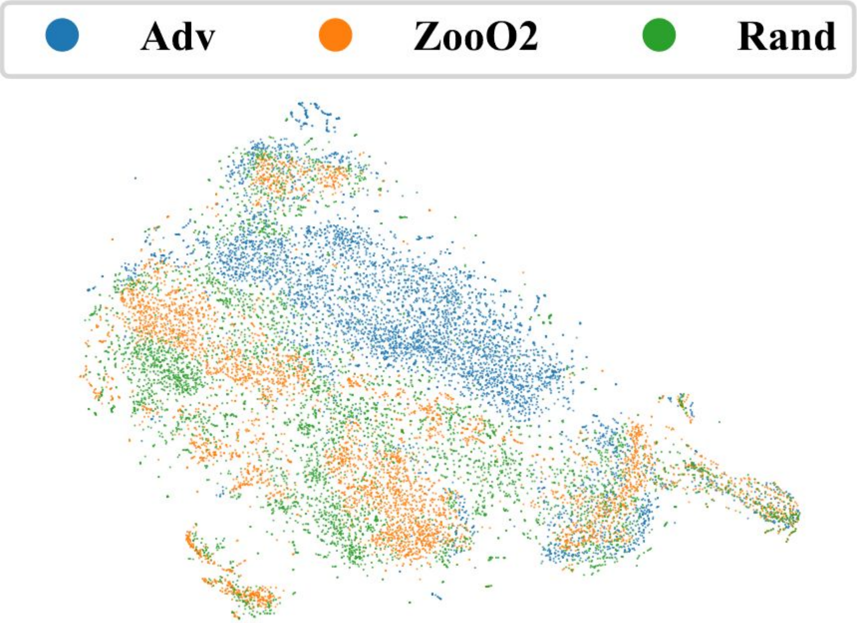
Victim	Adv1	Adv2	Adv3
ZooV1	79	53	50
ZooV2	54	93	63
ZooV3	31	31	84

Win Matrix for Kick and Defend

Understanding adversarial policies



Gaussian mixture model fit to activity of one victim, mean log probabilities for opponent policies



t-SNE embedding of the victim activations for different opponent policies, victim is ZooV2 for Kick and Defend

Defending against Adversarial Attacks

- Masking the configuration of the opponent significantly increases its performance (13% -> 99%)

- Fine-tuning against adversarial agents makes victims more robust (13% -> ~90% WR)
- Adversarial attack may be reapplied, but the resulting adversarial policy is conceptually different (1% WR)

Summary and Outlook

- Agents in multi-agent settings are susceptible to adversarial policies
- The adversarial policies induce abnormal activation patterns in the victims
- Masking and fine-tuning can help against adversarial agents
- It would be interesting to…
    - analyse the adversarial policies after repeated fine-tuning
    - train policies with adversarial attackers in the pool

Remarks

- Interesting and direct method for applying adversarial attacks in RL
- Some findings are not fully explained or left out
- Great analysis of the network activations
- Great additional material [Source code, Videos]

References

[1] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine & S. Russell, *Adversarial Policies: Attacking Deep Reinforcement Learning* in ICLR 2020
https://adversarialpolicies.github.io

[2] I. Goodfellow, J.Shlens & C. Szegedy, *Explaining and Harnessing Adversarial Example* in ICLR 2015

[3] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever & I. Mordatch, *Emergent Complexity via Multi-Agent Competition* in ICLR 2018

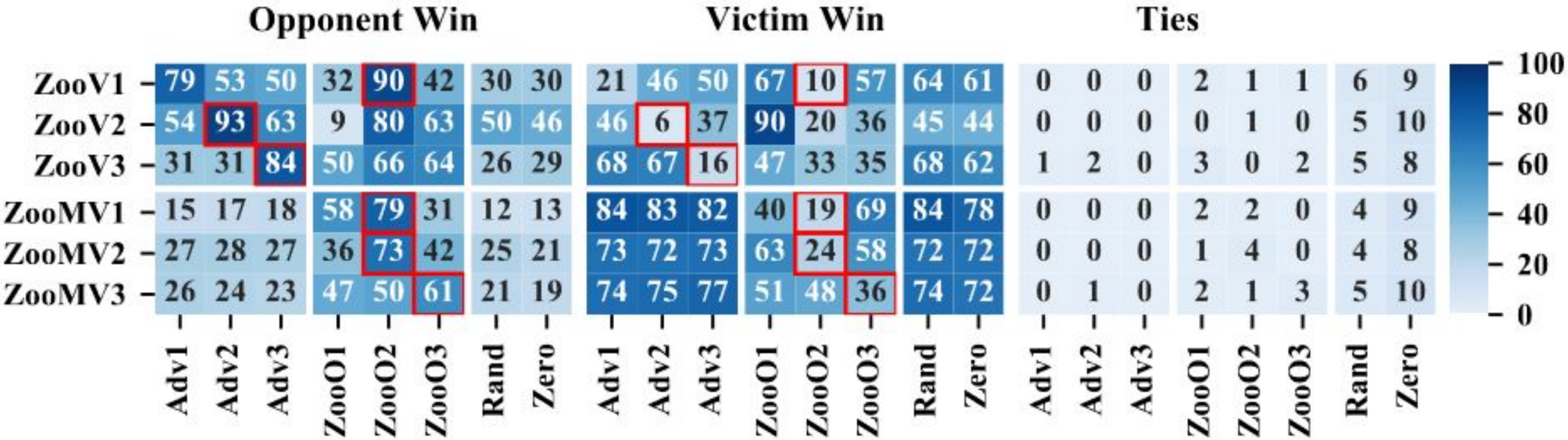
[4] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor & Y. Wu. *Stable Baselines*
https://github.com/hill-a/stable-baselines

[5] L. van der Maaten, G. Hinton, *Visualizing Data using t-SNE* in Journal of Machine Learning Research

[6] S. Bhambri, S. Muku, A. Tulasi & A. Buduru *A Survey of Black-Box Adversarial Attacks on Computer Vision Models*

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu *Towards Deep Learning Models Resistant to Adversarial Attacks*

Appendix A:



Opponent Win

	Adv1	Adv2	Adv3	ZooO1	ZooO2	ZooO3	Rand	Zero
ZooV1	79	53	50	32	90	42	30	30
ZooV2	54	93	63	9	80	63	50	46
ZooV3	31	31	84	50	66	64	26	29
ZooMV1	15	17	18	58	79	31	12	13
ZooMV2	27	28	27	36	73	42	25	21
ZooMV3	26	24	23	47	50	61	21	19

Victim Win

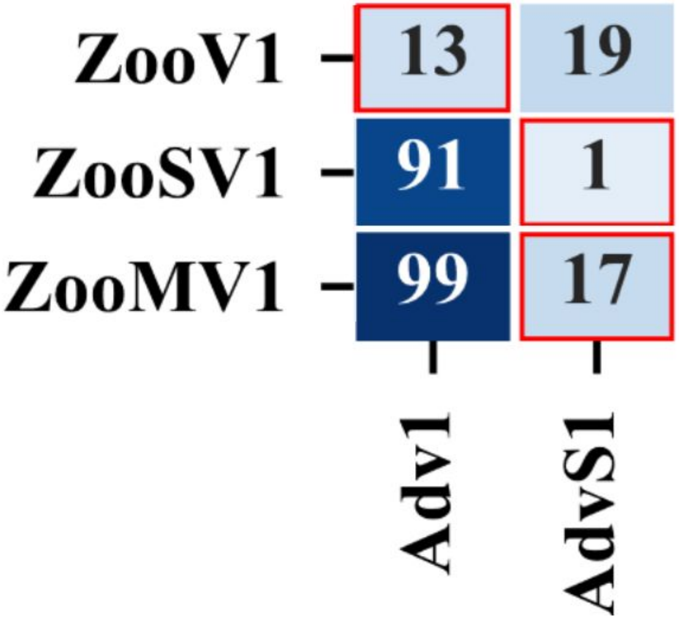
	Adv1	Adv2	Adv3	ZooO1	ZooO2	ZooO3	Rand	Zero
ZooV1	21	46	50	67	10	57	64	61
ZooV2	46	6	37	90	20	36	45	44
ZooV3	68	67	16	47	33	35	68	62
ZooMV1	84	83	82	40	19	69	84	78
ZooMV2	73	72	73	63	24	58	72	72
ZooMV3	74	75	77	51	48	36	74	72

Ties

	Adv1	Adv2	Adv3	ZooO1	ZooO2	ZooO3	Rand	Zero
ZooV1	0	0	0	2	1	1	6	9
ZooV2	0	0	0	0	1	0	5	10
ZooV3	1	2	0	3	0	2	5	8
ZooMV1	0	0	0	2	2	0	4	9
ZooMV2	0	0	0	1	4	0	4	8
ZooMV3	0	1	0	2	1	3	5	10

(a) Kick and Defend

Appendix B



	Adv1	AdvS1
ZooV1	13	19
ZooSV1	91	1
ZooMV1	99	17