FLARE: Fingerprinting Deep Reinforcement Learning Agents using Universal Adversarial Masks

Buse G. A. Tekgul (Nokia Bell Labs & Aalto University)
N. Asokan (University of Waterloo & Aalto University)

07/12/2023

Model Theft vs. Ownership Demonstration in ML

AI/ML models: Business advantage, **Intellectual Property**

Adversary compromises model confidentiality:

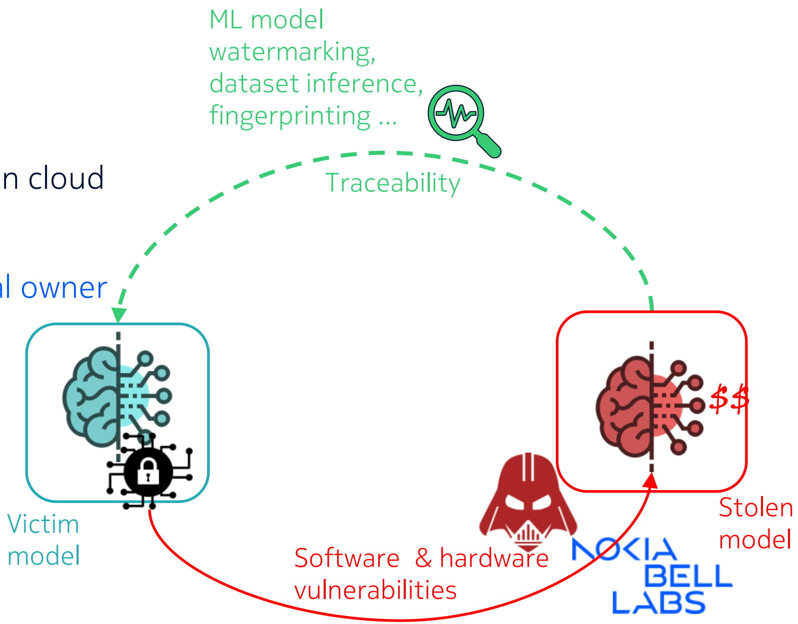
unauthorized redistribution or monetization of on device ML models

Proactive protection mechanisms:

- Computation with encrypted models, secure hardware, deployment in cloud

Reactive Defenses (Defense by deterrence):

- Ownership verification traces stolen copies of models back to original owner



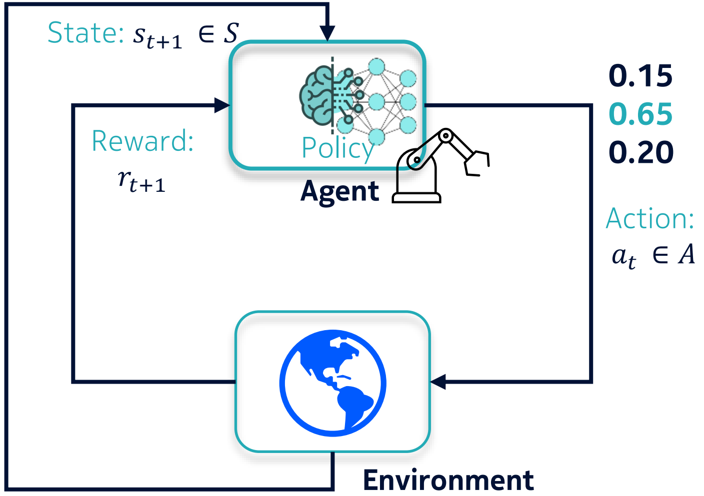
(Deep) Reinforcement Learning

Reinforcement Learning (RL): an agent continuously interacts with an environment to optimize its policy[1]

- Policy: Decision making strategy: $\pi(a_t|s_t): S \rightarrow A$
- Decided optimal action: $\hat{\pi}_i(s_t)$
- Optimal policy leads best average return from the task

Deep RL (DRL): Agent learns policies from high-dimensional inputs

- RL defines the objective:
  - maximizes future reward
- Deep Neural Networks (DNN) provides the mechanism:
  - approximates the policy



[1] Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction. The MIT Press.

Ownership Verification in DRL

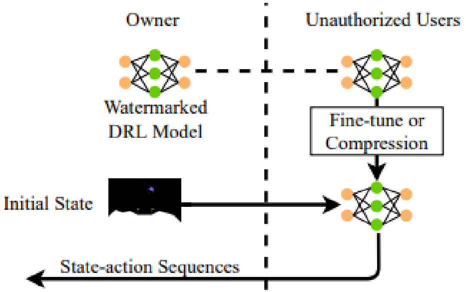
Current methods[1,2] adopt DNN model watermarking techniques

- Model watermarking embeds traceable information (watermark) to the model by either directly inserting it into model parameters or adding unique knowledge into a small subset of the training set

Model watermarking in DRL requires

- modifying both the training process and the reward function
- sending specific input states to start the verification process

[2]

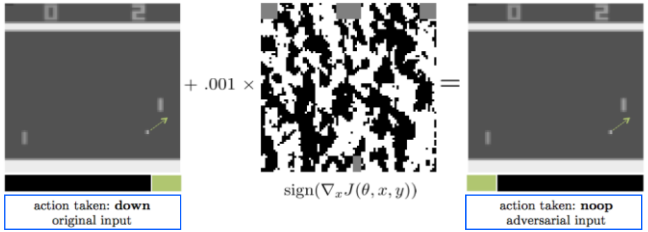


DNN fingerprinting methods[3,4] use individual or universal adversarial examples since they can characterize decision boundary of classifiers

[1] Behzadan and Hsu (2019). Sequential Triggers for Watermarking of Deep Reinforcement Learning Policies
[2] Chen et al . (AAMAS 2021). Temporal Watermarks for Deep Reinforcement Learning Models
[3] Lukas et al. (ICLR 2019). Deep Neural Network Fingerprinting by Conferrable Adversarial Examples
[4] Peng et al. (CVF 2022). Fingerprinting Deep Neural Networks Globally via Universal Adversarial Perturbations

Adversarial Examples

[1]



Adversarial perturbation is added into clean input

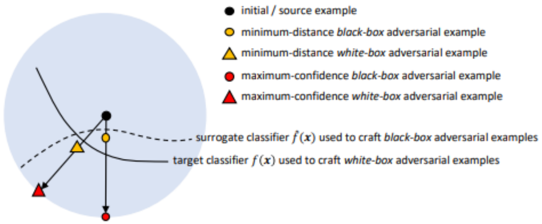
DNN[1]: Classifier is victim, incorrect label

DRL[2,3]: Policy is victim, sub-optimal action

Universal adversarial masks: Single, minimum amount of perturbation r that fools victim model on almost all data points[4]

- Constrained via ϵ, i.e., $||r||_p \leq \epsilon$
- Effectiveness of r measured via fooling rate, δ_r on a test set

[5]



Adversarial examples can transfer between different DNN models trained for the same task[5]

[1] Szegedy et al. (2013). Intriguing Properties of Neural Networks
[2] Huang et al. (2018). Adversarial Attacks on Neural Network Policies
[3] Gleave et al. (ICLR 2020) Adversarial Policies: Attacking Deep Reinforcement Learning
[4] Moosavi-Dezfooli et al. (CVPR 2017) Universal Adversarial Perturbations
[5] Demontis et al . (USENIX 2019) Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks

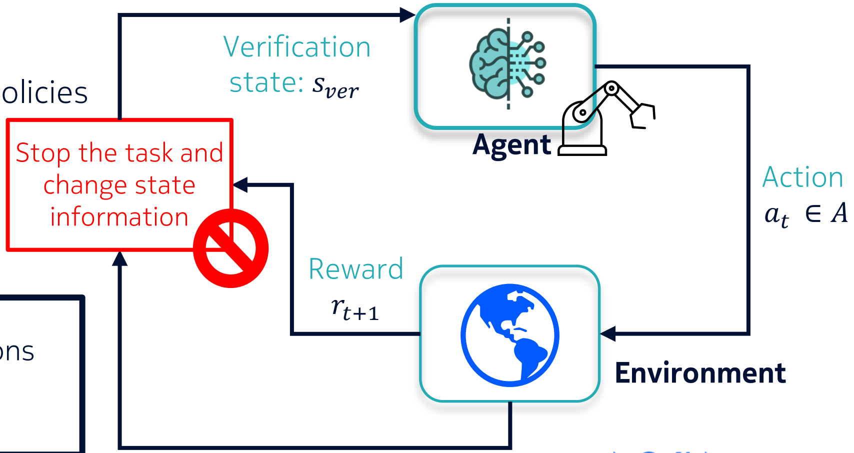
FLARE: Fingerprinting DRL Policies using Universal Adversarial Masks

In DRL,

- There is no 1-to-1 mapping between input state and the optimal action
- Inserting individual adversarial states into a dynamic environment requires stopping the task

FLARE : The first fingerprinting method for DRL policies

- computes non-transferable, universal adversarial masks that can transfer from victim to stolen policies but not independent policies
- verifies the true ownership of stolen model by measuring the similarity of the changed behavior



Does not depend on any specific start state, optimal/sub-optimal actions

No need to stop the task, just a noise addition for a while

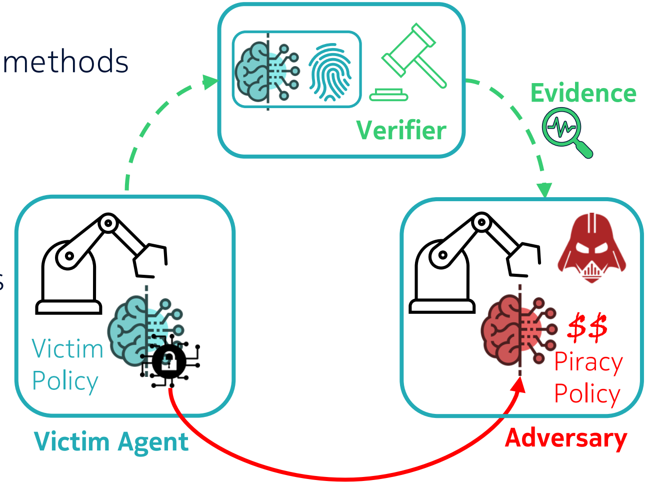
FLARE: Adversary Model

Adversary (A):

- Aims to obtain/redistribute an illegal copy of the victim agent's policy
- Has access to a similar environment, computational capabilities
- Attempts to evade/degrade effectiveness of possible ownership verification methods
- Well-informed adversaries

Verifier (Judge, Trusted third party, J):

- Has white-box access to victim policy, black-box access to suspected policies
- Can modify environment (add adversarial mask to input states)



NOKIA BELL LABS

FLARE: Adversary Model

Effectiveness: Successful ownership verification of stolen models with high confidence

Integrity: Avoiding accidental accusations of independently trained policies

Robustness: Withstanding model modification and evasion attacks

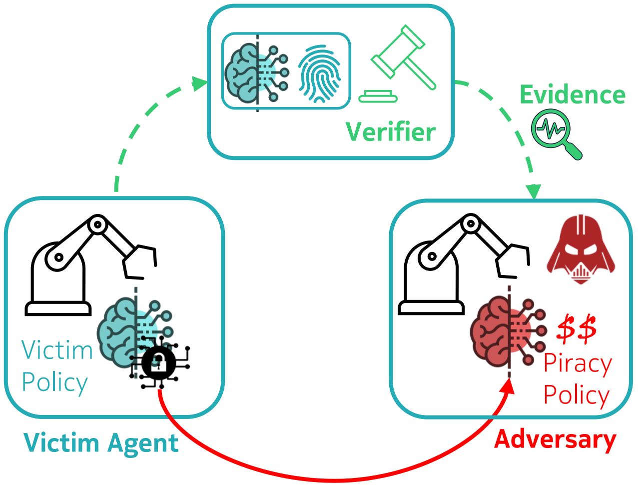
- Successful verification after model modification
- Failed verification along with a noticeable decrease in agent performance

Green: High AA stays at high value, successful verification

Blue: AA drops little, successful verification

Yellow: Low AA, failed verification, agent performs poorly

Red: Low AA, failed verification, agent performs well

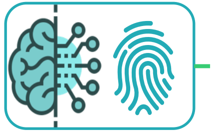


FLARE: Methodology

Fingerprint generation

- Generate a maximum confidence but non-transferable, universal masks r from randomly sampled states during a single episode eps

  using π_V and independent models $\pi_i , i \epsilon I$
- Compute non-transferability score (nts) on another eps
- $AA(\pi_i , \pi_j , s, r)$ refers to action agreement & key statistics that measures behavioral similarity
- Add valid r into fingerprint list if both nts and fooling rate δ_r are bigger than a threshold value

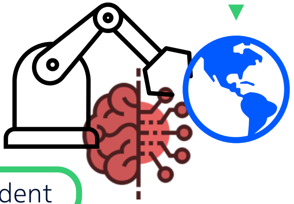


Fingerprint verification

- Suspected policy π_s in deployment
- Observe π_s to estimate the time spent to finish the task
- Add each fingerprint r to states

  during time window in verification, save $[s_t + r]_{t=i}^{t=i+N}$

  episodes, compute AA
- For each r, if $AA \geq 0.5$, it's one supporting evidence
- Final verdict is given by majority vote
- Average AA gives confidence of verdict



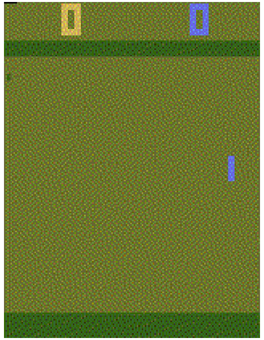
Stolen or Independent

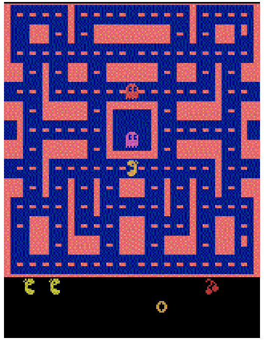
NOKIA BELL LABS

FLARE: Empirical Analysis

- Arcade Learning Environment[1]: Pong and Ms Pacman
- DRL algorithms (PyTorch, NVIDIA Quadro P5000): DDQN[2], A2C[3], PPO[4]
  - 6 victim policies in total (similar performance as OpenAI Baselines)
  - 5X3 independent policies (5 same algorithm are π_I , rest π_{others})

- 10 fingerprints for each π_V
- Verification episodes window size $M = \min(100, len(eps))$





[1] https://www.gymlibrary.dev/environments/atari/
[2] Mnih et al. (Nature 2015). Human-level Control Through Deep Reinforcement Learning
[3] Mnih et al. (ICML 2016). Asynchronous Methods for Deep Reinforcement Learning
[4] Schulman et al. (2017). Proximal Policy Optimization Algorithms

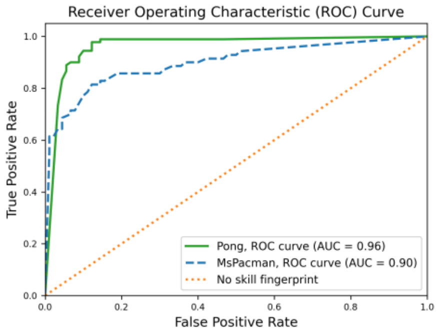
Effectiveness and Integrity

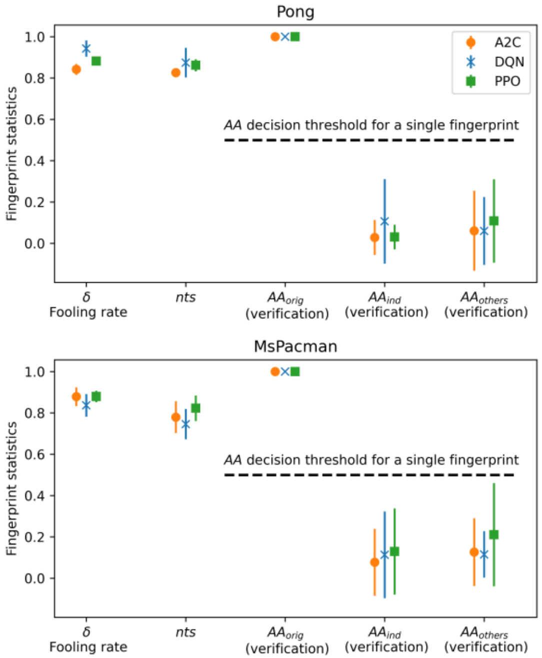
FLARE

- can distinguish stolen policies from independent ones

  while achieving high fooling rate and non-transferability score
- leads no accidental false accusation of independently trained models

Global decision threshold ($AA \geq 0.5$) is selected from ROC curves





FLARE satisfies both effectiveness and integrity requirements

Robustness against Model Modification Attacks

- Model modification attacks change the decision boundary of ML models
  - Fine-tuning[1] and weight pruning[2]

$$Impact(on\ average\ return) \leq 0.4$$

FLARE is robust against model modification attacks

Table 1: Average impact, *AA* and voting results (✓:Stolen, ✗: Independent) for piracy policies that are 1) fine-tuned over a different number of episodes and 2) pruned and then fine-tuned over 200 episodes. *AA* is averaged over 10 verification episodes, while impact is averaged over 10 test episodes. (green: Successful verification with $AA \geq 0.75$, blue: Successful verification with $0.75 \geq AA \geq 0.50$, yellow: Failed verification with high impact ≥ 0.4, red: Failed verification with low impact < 0.4)

Game, DRL method	Stats	Fine-tuning, # of episodes: 50	100	200	Pruning and fine-tuning, pruning levels (%): 25	50	75	90
Pong, A2C	Impact	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0
	AA	0.95 ± 0.14	0.95 ± 0.14	0.94 ± 0.10	0.94 ± 0.14	0.91 ± 0.25	0.67 ± 0.42	0.28 ± 0.42
	Votes	10 ✓/ 0 ✗	10 ✓/ 0 ✗	10 ✓/ 0 ✗	10 ✓/ 0 ✗	9 ✓/ 1 ✗	6 ✓/ 4 ✗	3 ✓/ 7 ✗
Pong, DQN	Impact	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.8 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	AA	0.94 ± 0.05	0.89 ± 0.14	0.90 ± 0.17	0.88 ± 0.16	0.66 ± 0.38	0.09 ± 0.17	0.27 ± 0.4
	Votes	10 ✓/ 0 ✗	10 ✓/ 0 ✗	9 ✓/1 ✗	10 ✓/ 0 ✗	7 ✓/ 3 ✗	1 ✓/ 9 ✗	3 ✓/ 7 ✗
Pong, PPO	Impact	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	AA	0.88 ± 0.23	0.89 ± 0.25	0.88 ± 0.30	0.78 ± 0.35	0.67 ± 0.35	0.65 ± 0.41	0.71 ± 0.39
	Votes	9 ✓/1 ✗	9 ✓/1 ✗	9 ✓/1 ✗	7 ✓/ 3 ✗	7 ✓/ 3 ✗	6 ✓/ 4 ✗	7 ✓/ 3 ✗
MsPacman, A2C	Impact	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.39 ± 0.19	0.03 ± 0.10	0.30 ± 0.15	0.73 ± 0.11
	AA	0.82 ± 0.16	0.75 ± 0.29	0.62 ± 0.35	0.71 ± 0.28	0.65 ± 0.39	0.72 ± 0.26	0.59 ± 0.23
	Votes	9 ✓/1 ✗	8 ✓/2 ✗	6 ✓/4 ✗	6 ✓/4 ✗	7 ✓/3 ✗	8 ✓/2 ✗	6 ✓/4 ✗
MsPacman, DQN	Impact	0.79 ± 0.11	0.83 ± 0.02	0.87 ± 0.03	0.79 ± 0.11	0.74 ± 0.09	0.86 ± 0.01	0.71 ± 0.43
	AA	0.23 ± 0.34	0.15 ± 0.28	0.16 ± 0.31	0.38 ± 0.44	0.00 ± 0.01	0.59 ± 0.46	0.42 ± 0.42
	Votes	2 ✓/8 ✗	1 ✓/9 ✗	2 ✓/8 ✗	4 ✓/6 ✗	0 ✓/10 ✗	6 ✓/4 ✗	4 ✓/6 ✗
MsPacman, PPO	Impact	0.85 ± 0.11	0.40 ± 0.26	0.51 ± 0.08	0.52 ± 0.15	0.57 ± 0.04	0.62 ± 0.05	0.66 ± 0.19
	AA	0.43 ± 0.36	0.11 ± 0.16	0.25 ± 0.32	0.26 ± 0.36	0.33 ± 0.38	0.31 ± 0.32	0.13 ± 0.20
	Votes	4 ✓/6 ✗	0 ✓/10 ✗	3 ✓/7 ✗	3 ✓/ 7 ✗	3 ✓/ 7 ✗	4 ✓/ 6 ✗	1 ✓/ 9 ✗

[1] Razavian et al. (CVPR 2014). CNN Features off-the-shelf: an Astounding Baseline for Recognition
[2] Han et al. (NeurIPS 2015). Learning both Weights and Connections for Efficient Neural Networks

NOKIA BELL LABS

Robustness against Well-informed Adversaries

Adversary can evade the verification by returning sub-optimal actions randomly or detect adversarial states and try to recover optimal (original) action

- *AA* stays at high values against these adversaries
- Final verdict do not change for good returns

Well informed adversaries can use adversarial training to make stolen policies robust against adversarial attacks (& FLARE) [2]

FLARE is robust against evasion attacks

FLARE is not robust against adversarial training, but robust when it is used with adversarially trained victim agents

Table 2: Average impact, *AA* and voting results for stolen policies modified by RADIAL-DQN. Results are reported for both the agent with the best performance during RADIAL-DQN (3rd column) and the final agent obtained after RADIAL-DQN finishes (4th column). *AA* is averaged on 10 verification episodes and impact is averaged over 10 test episodes. (*: improved policy, green: Successful verification with $AA \geq 0.75$, blue: Successful verification with $0.75 \geq AA \geq 0.50$, yellow: Failed verification with high impact ≥ 0.4, red: Failed verification with low impact < 0.4)

Game, DRL method	Stats	Best Agent	Final Agent
Pong, RADIAL-DQN	Impact	0.0 ± 0.0	0.0 ± 0.0
	AA	0.04 ± 0.06	0.04 ± 0.06
	Votes	0 ✓/ 10 ✗	0 ✓/ 10 ✗
MsPacman, RADIAL-DQN	Impact	$-0.16 \pm 0.03^*$	0.39 ± 0.03
	AA	0.59 ± 0.40	0.29 ± 0.31
	Votes	6 ✓/ 4 ✗	4 ✓/ 6 ✗
Pong, RADIAL-RDQN	Impact	0.0 ± 0.0	0.0 ± 0.0
	AA	0.84 ± 0.21	0.89 ± 0.17
	Votes	8 ✓/ 2 ✗	9 ✓/ 1 ✗
MsPacman, RADIAL-RDQN	Impact	0.15 ± 0.04	0.55 ± 0.06
	AA	0.61 ± 0.34	0.09 ± 0.18
	Votes	7 ✓/ 3 ✗	1 ✓/ 9 ✗

[1] Lin et al. (2017). Detecting Adversarial Attacks on Neural Network Policies with Visual Foresight

[2] Oikarinen et al. (NeurIPS 2021). Robust Deep Reinforcement Learning through Adversarial Loss

Robustness against False Claims

Malicious accusers can produce fake fingerprints to pass the ownership verification test against independent (and victim) policies [1]

Verifier can reject the claim by

- checking the amount of perturbation ϵ
- Search for other independent policies (PPO)

Table 3: AA values (averaged over 10 verification episodes) and voting results for false claims against victim $\mathcal{V}$, and independent $\mathcal{I}$ policies with different perturbation constraint ϵ values. (The cases where a false claim succeeds are shown as follows: green: False claim with $AA \geq 0.75$, purple: False claim with $0.75 \geq AA \geq 0.50$)

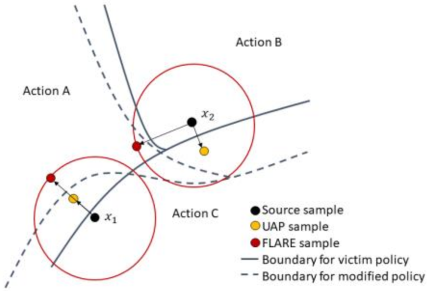
Game, DRL method		ϵ vs. AA (Votes) 0.05	0.1	0.2	0.5
Pong, A2C	$\mathcal{V}$	0.45 ± 0.47 (5 ✓/ 5 ✗)	0.49 ± 0.49 (5 ✓/ 5 ✗)	0.40 ± 0.49 (4 ✓/ 6 ✗)	0.40 ± 0.49 (4 ✓/ 6 ✗)
	$\mathcal{I}$, avg.	0.32 ± 0.36 (3 ✓/ 7 ✗)	0.38 ± 0.45 (3 ✓/ 7 ✗)	0.30 ± 0.41 (3 ✓/ 7 ✗)	0.28 ± 0.43 (3 ✓/ 7 ✗)
Pong, DQN	$\mathcal{V}$	0.37 ± 0.42 (4 ✓/ 6 ✗)	0.37 ± 0.45 (3 ✓/ 7 ✗)	0.33 ± 0.45 (3 ✓/ 7 ✗)	0.40 ± 0.49 (4 ✓/ 6 ✗)
	$\mathcal{I}$, avg.	0.01 ± 0.18 (1 ✓/ 9 ✗)	0.07 ± 0.22 (1 ✓/ 9 ✗)	0.05 ± 0.19 (1 ✓/ 9 ✗)	0.05 ± 0.19 (1 ✓/ 9 ✗)
Pong, PPO	$\mathcal{V}$	0.56 ± 0.39 (5 ✓/ 5 ✗)	0.68 ± 0.42 (7 ✓/ 3 ✗)	0.76 ± 0.38 (8 ✓/ 2 ✗)	0.78 ± 0.39 (8 ✓/ 2 ✗)
	$\mathcal{I}$, avg.	0.56 ± 0.36 (6 ✓/ 4 ✗)	0.59 ± 0.38 (6 ✓/ 4 ✗)	0.59 ± 0.38 (6 ✓/ 4 ✗)	0.52 ± 0.41 (6 ✓/ 4 ✗)
MsPacman, A2C	$\mathcal{V}$	0.00 ± 0.00 (0 ✓/10 ✗)	0.03 ± 0.05 (0 ✓/10 ✗)	0.14 ± 0.29 (1 ✓/9 ✗)	0.09 ± 0.22 (1 ✓/9 ✗)
	$\mathcal{I}$, avg.	0.15 ± 0.56 (1 ✓/9 ✗)	0.14 ± 0.21 (1 ✓/9 ✗)	0.13 ± 0.30 (2 ✓/8 ✗)	0.21 ± 0.36 (2 ✓/8 ✗)
MsPacman, DQN	$\mathcal{V}$	0.23 ± 0.36 (2 ✓/8 ✗)	0.0 ± 0.0 (0 ✓/10 ✗)	0.0 ± 0.0 (0 ✓/10 ✗)	0.0 ± 0.0 (0 ✓/10 ✗)
	$\mathcal{I}$, avg.	0.26 ± 0.24 (2✓/8 ✗)	0.19 ± 0.26 (2✓/8 ✗)	0.15 ± 0.29 (1✓/9✗)	0.24 ± 0.26 (3✓/7✗)
MsPacman, PPO	$\mathcal{V}$	0.19 ± 0.18 (1 ✓/ 9 ✗)	0.26 ± 0.31 (3 ✓/ 7 ✗)	0.38 ± 0.37 (4 ✓/ 6 ✗)	0.07 ± 0.21 (1 ✓/ 9 ✗)
	$\mathcal{I}$, avg.	0.10 ± 0.11 (0 ✓/10 ✗)	0.50 ± 0.39 (5 ✓/5 ✗)	0.74 ± 0.40 (8 ✓/2 ✗)	0.80 ± 0.20 (8 ✓/2 ✗)

FLARE is not susceptible to false claims with a simple additional countermeasure on ϵ and non-transferability check based on the DRL algorithm

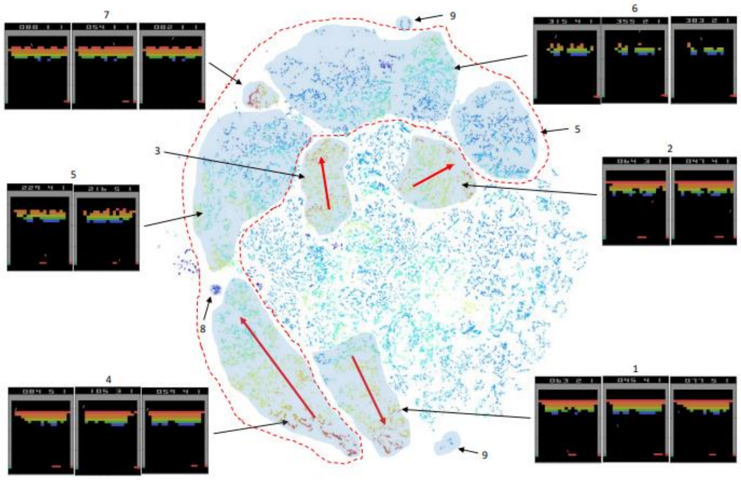
[1] Liu et al. (https://arxiv.org/abs/2304.06607, 2023). False claims against Model Ownership Resolution.

Universality vs. Transferability

- Input space embeddings in DRL are not as separable as in DNN
- In DRL, input states have spatio-temporal abstractions, and policies are hierarchal[1]
- UAP[1] (minimum-distance method) finds the smallest high-sensitivity directions belonging to closest incorrect class
- FLARE identifies spatially similar pockets that are distant from each other in temporal dimension



[2]



Can we find better fingerprints/adversarial examples that break temporal abstractions?

[1] Moosavi-Dezfooli et al. (CVPR 2017) Universal Adversarial Perturbations

[2] Zahavy et al. (ICML 2016). Graying the black box: Understanding DQNs

Conclusion & Takeaways

FLARE: **the first fingerprint mechanism that** verifies the ownership of illegitimate DRL policies using universal adversarial masks

FLARE **satisfies**

- Effectiveness (100% action agreement on stolen policies),
- Integrity (no false positives)
- Robustness (successful verification of stolen policies when the impact on performance ≤ 0.4)

The choice of universal adversarial mask method is crucial due to **inherent characteristics** of DRL policies

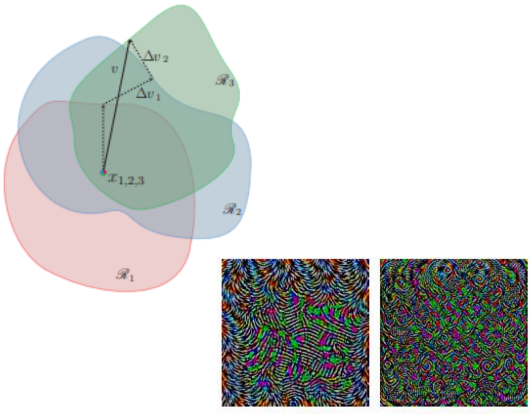
NOKIA BELL LABS

Additional Slides

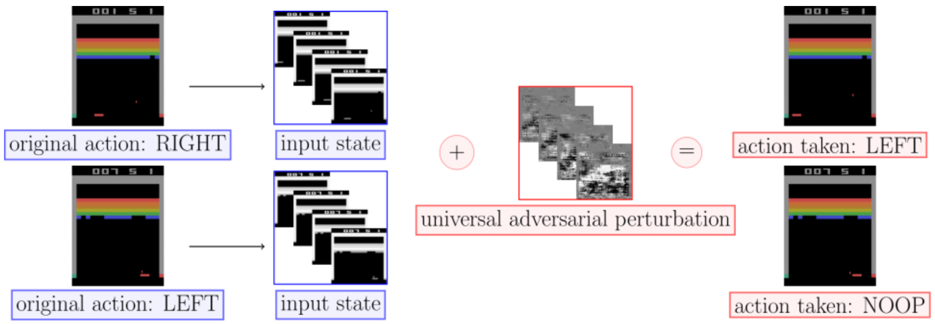
Universal Adversarial Perturbations

- A single, minimum amount of perturbation r that fools the victim model on almost all data points[1]
  - Constrained via ϵ, i.e., $||r||_p \leq \epsilon$
  - Effectiveness of r measured via fooling rate, δ_r on a test set

[1]



[2]



[1] Moosavi-Dezfooli et al. (CVPR 2017) Universal Adversarial Perturbations
[2] Tekgul et al . (ESORICS 2022) Real-time Adversarial Perturbations against Deep Reinforcement Learning Policies: Attacks and Defenses

FLARE: Methodology

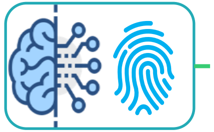
Fingerprint generation

- Generate a maximum confidence but non-transferable, universal masks r from randomly sampled states during a single episode eps

  using π_V and independent models $\pi_i, i \in I$
- Compute non-transferability score on another eps

  $$nts(r, eps) = \delta_{r,eps} \times max_{i \in I}(1 - AA(\pi_V, \pi_i, s, r))$$
- AA refers to action agreement & key statistics that measures behavioral similarity

  $$AA(\pi_i, \pi_j, s, r) = {}^{1}/_{N} \sum_{t=0}^{t=N} \mathbf{1}_{\hat{\pi}_i(s_t+r)=\hat{\pi}_j(s_t+r)}$$
- Add valid r into fingerprint list

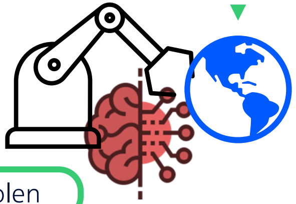


Fingerprint verification

- Suspected policy π_s in deployment
- Observe π_s to estimate the time spent to finish the task
- Add each fingerprint r to states

  during time window in verification, save $[s_t + r]_{t=i}^{t=i+N}$

  episodes, compute AA
- For each r, if $AA \geq 0.5$, it's one supporting evidence
- Final verdict is given by majority vote
- Average AA gives confidence of verdict



Stolen or Not stolen

NOKIA BELL LABS

  | Nokia internal use

Robustness against Well-informed Adversaries (I)

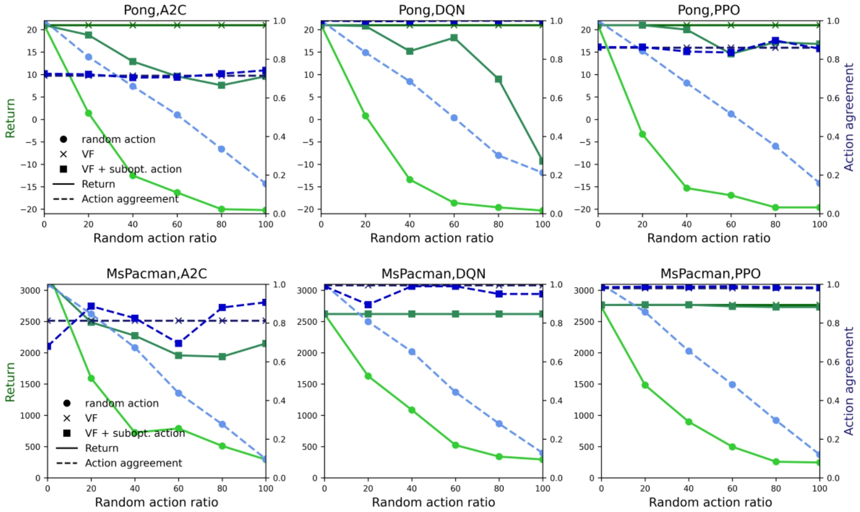
Adversary can evade the verification by:

- performing sub-optimal actions with a random action ratio
- detecting adversarial states and trying to recover the optimal (original action) using a history of saved [states + actions]
- detecting adversarial states and performing a random action to those inputs

Random action

Visual Foresight (VF)[1]

VF + sub-optimal action



AA stays at high values, and the final verdict do not change for good returns: FLARE is robust against evasion attacks.

[1] Lin et al. (2017). Detecting Adversarial Attacks on Neural Network Policies with Visual Foresight

NOKIA BELL LABS

Robustness against Well-informed Adversaries (II)

- Well informed adversaries can use adversarial training to make stolen model robust against adversarial attacks (& FLARE)
- Stolen policy + RADIAL-DQN[1]

FLARE is not robust against adversarial training, but robust when it is used with adversarially trained victim agents.

Table 2: Average impact, AA and voting results for stolen policies modified by RADIAL-DQN. Results are reported for both the agent with a best performance during RADIAL-DQN (3rd column) and the final agent obtained after RADIAL-DQN finishes (4th column). AA is averaged on 10 verification episodes and impact is averaged over 10 test episodes. (*: improved policy, (green) : Successful verification with $AA \geq 0.75$, (blue) : Successful verification with $0.75 \geq AA \geq 0.50$, (yellow) : Failed verification with high impact ≥ 0.5, (red) : Failed verification with low impact < 0.5)

Game, DRL method	Stats	Best Agent	Final Agent
Pong, RADIAL-DQN	Impact	0.0 ± 0.0	0.0 ± 0.0
	AA	0.04 ± 0.06	0.04 ± 0.06
	Votes	0 ✓/ 10 ✗	0 ✓/ 10 ✗
MsPacman, RADIAL-DQN	Impact	$-0.16 \pm 0.03^*$	0.39 ± 0.03
	AA	0.59 ± 0.40	0.29 ± 0.31
	Votes	6 ✓/ 4 ✗	4 ✓/ 6 ✗
Pong, RADIAL-RDQN	Impact	0.0 ± 0.0	0.0 ± 0.0
	AA	0.84 ± 0.21	0.89 ± 0.17
	Votes	8 ✓/ 2 ✗	9 ✓/ 1 ✗
MsPacman, RADIAL-RDQN	Impact	0.15 ± 0.04	0.55 ± 0.06
	AA	0.61 ± 0.34	0.09 ± 0.18
	Votes	7 ✓/ 3 ✗	9 ✓/ 1 ✗

[1] Oikarinen et al. (NeurIPS 2021). Robust Deep Reinforcement Learning through Adversarial Loss

