

Evading next-gen AV using A.I.

<https://github.com/endgameinc/gym-malware>

Hyrum Anderson



hyrum@endgame.com



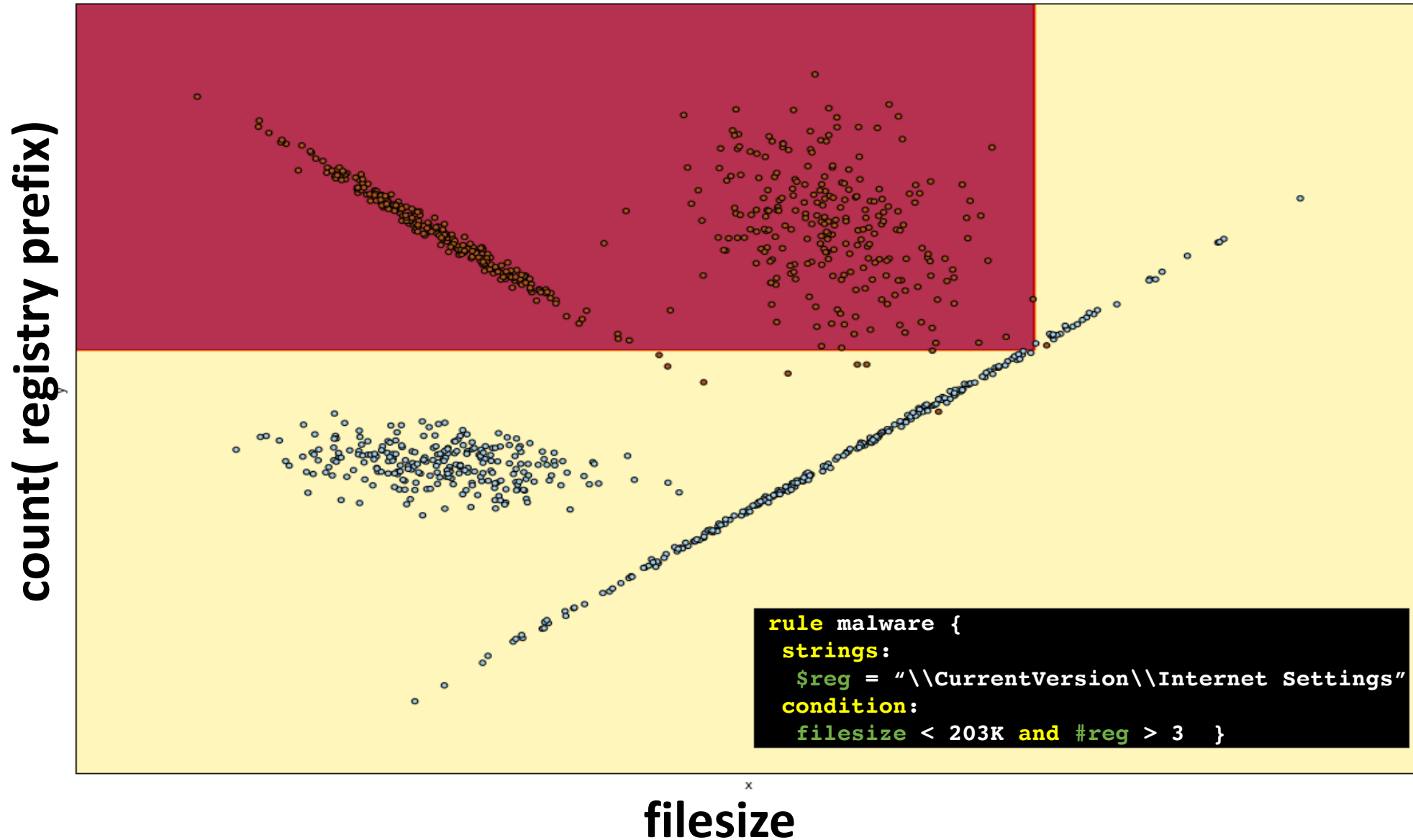
@drhyrum



/in/hyrumanderson



Rules for static analysis



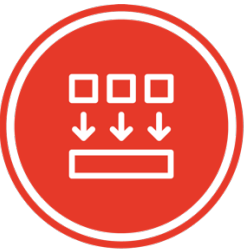
Next-gen AV using machine learning



Automated

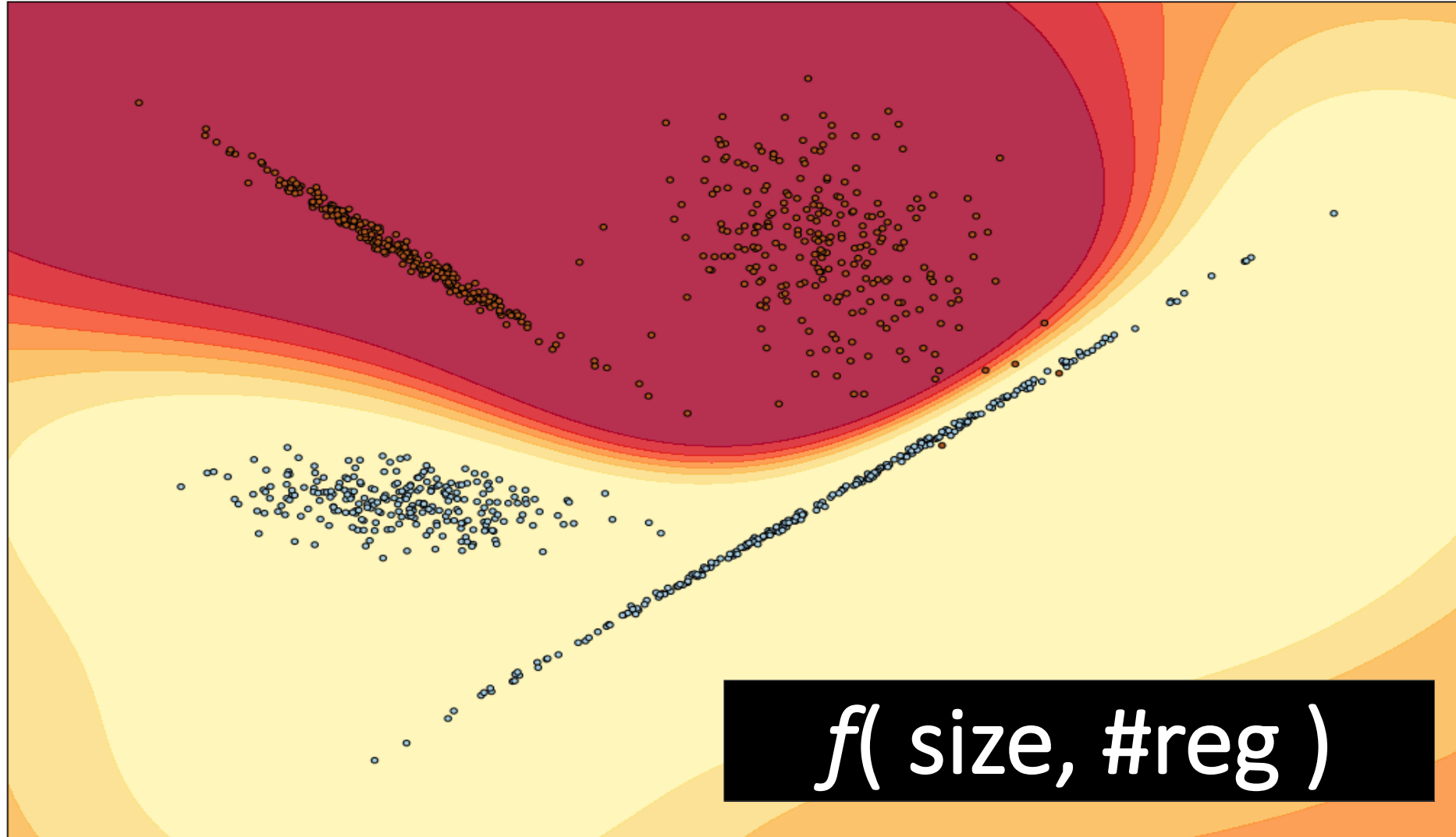


Sophisticated
relationships



Generalizes

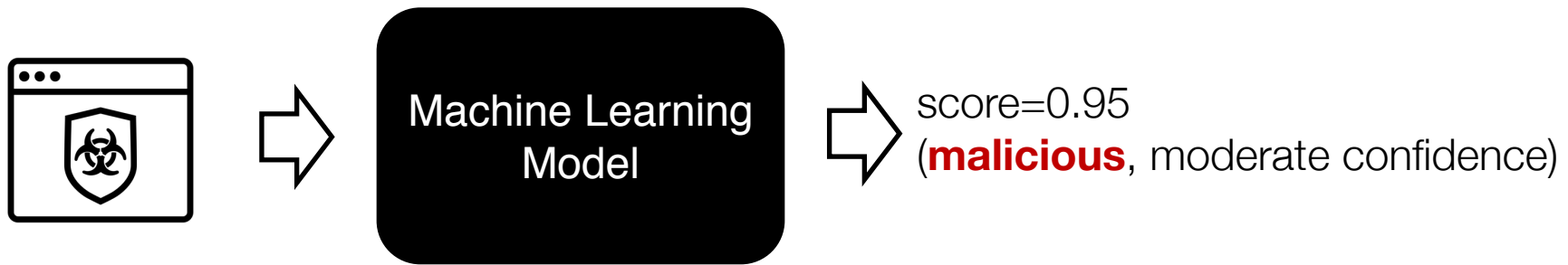
count(registry prefix)



filesize

Can We Break Machine Learning?

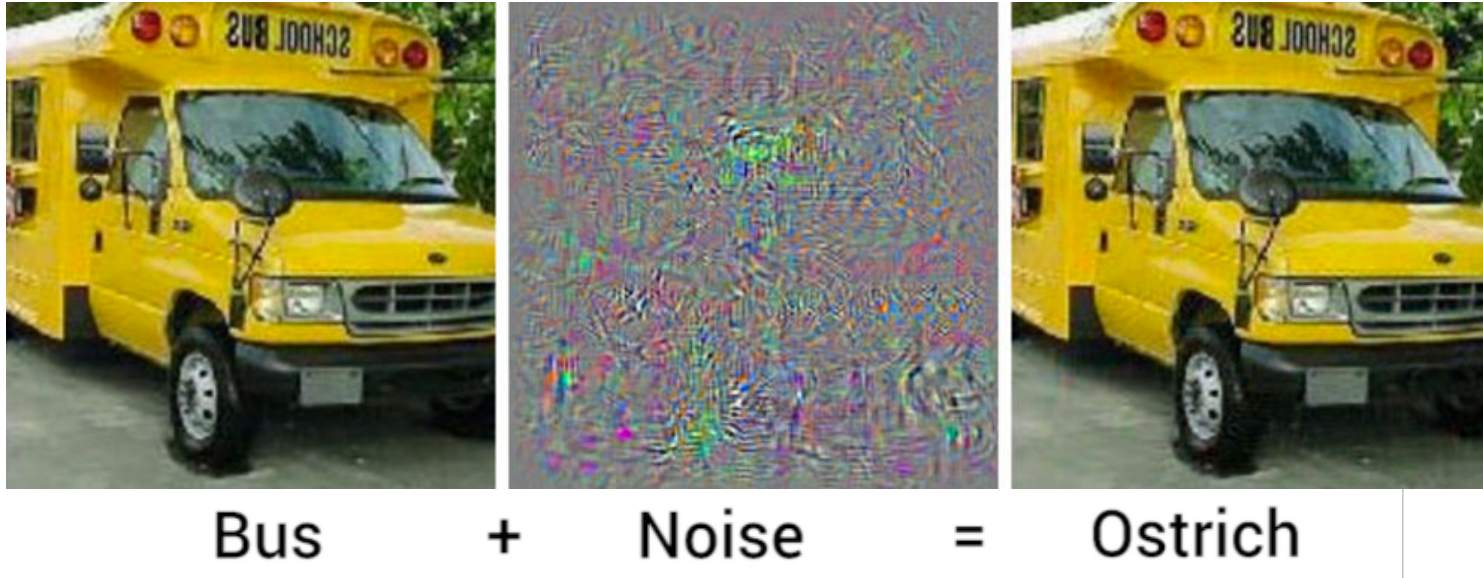
- Static machine learning model trained on millions of samples



- Simple structural changes that don't change behavior
 - upx_unpack
 - '.text' -> '.foo' (remains valid entry point)
 - create '.text' and populate with '.text from calc.exe'



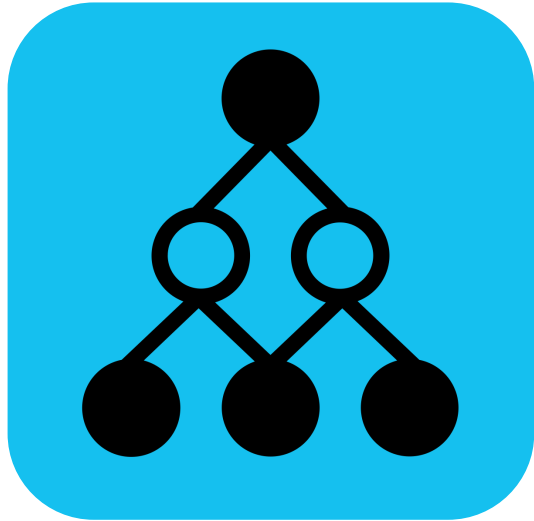
Yes! For deep learning and images...



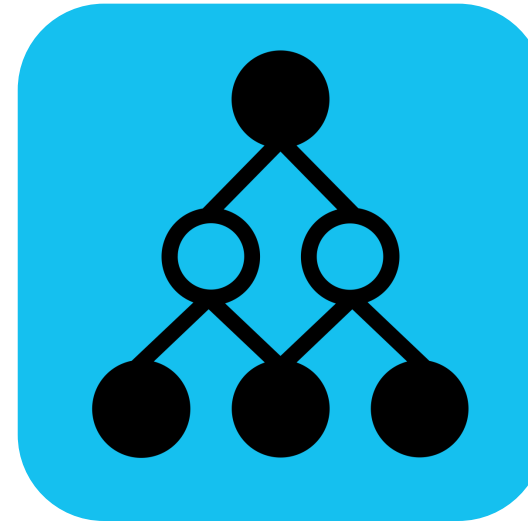
- Machine learning models have **blind spots** / **hallucinate** (modeling error)
- Depending on model and level of access, they **can be straightforward to exploit**
- Adversarial examples can **generalize across models / model types** (Goodfellow 2015)
 - blind spots in MY model may also be blind spots in YOUR model

What about for PE malware?

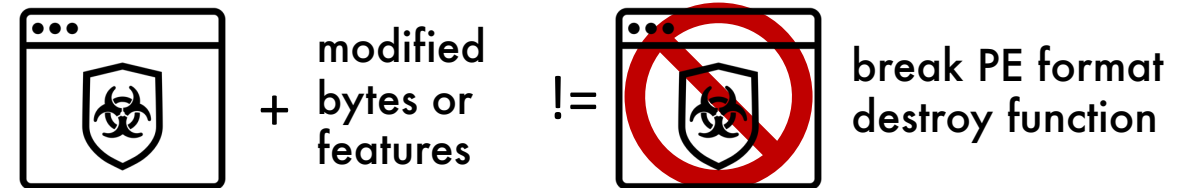
Bus (99%), Ostrich (1%)



Malware (90%), Benign (10%)



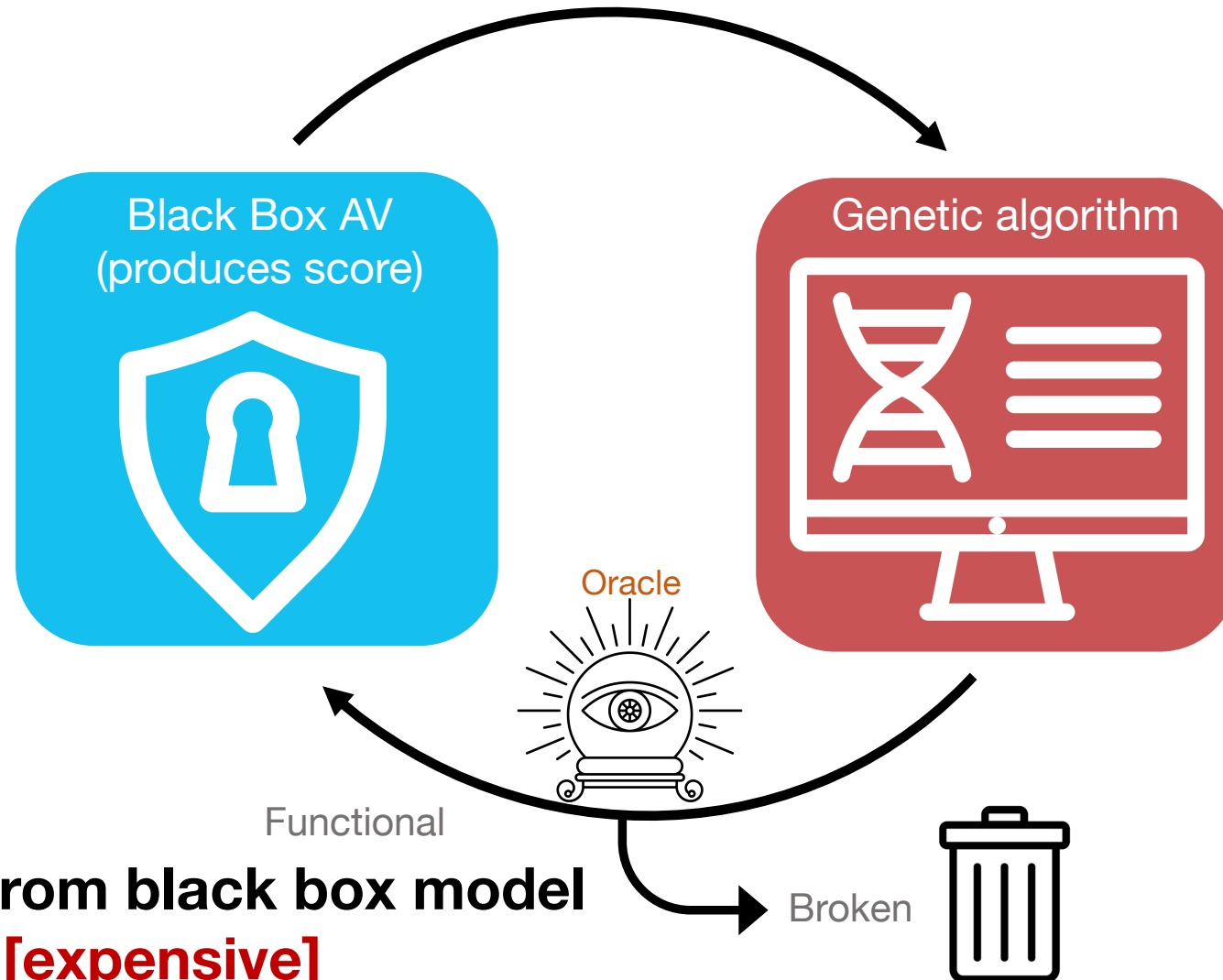
BUT...



Attacker requires full knowledge of *deep learning* model

Generated sample may not be valid PE file

Related Work: PDF attack / report score



Requires score from black box model
Oracle/sandbox [expensive]

EvadeML [for PDF malware]
(Xu, Qi, Evans, 2016)

Summary of Previous Works

Gradient-based attacks: perturbation or GAN

- Attacker **requires full knowledge** of model structure and weights
- Generated sample **may not be valid PE file**

Genetic Algorithms

- Attacker **requires score** from black box model
- Requires **oracle/sandbox [expensive]** to ensure that **functionality is preserved**

Goal: Design a machine learning agent that

- bypass **black-box** machine learning using
- **format-** and **function-preserving** mutations
- **hardest attack to pull off; difficult to learn**

Reinforcement Learning!

Atari Breakout



Nolan Bushnell, Steve Wozniak, Steve Bristow

Inspired by Atari Pong

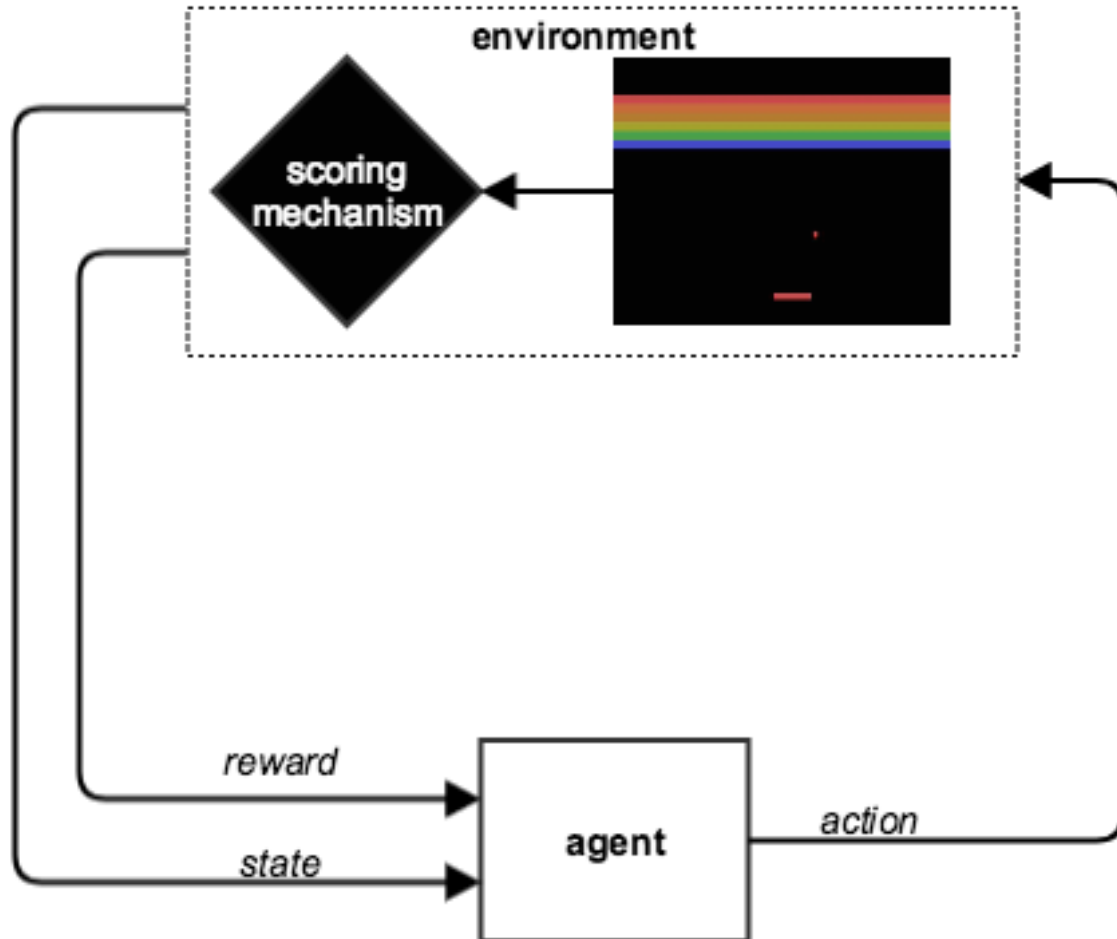
***"A lot of features of the Apple II went in
because I had designed Breakout for Atari"***

(The Woz)

Game

- Bouncing ball + rows of bricks
- Manipulate paddle (left, right)
- Reward for eliminating each brick

Atari Breakout: an AI



- **Environment**

- Bouncing ball + rows of bricks
- Manipulate paddle (*left, right, nothing*)
- Reward for eliminating each brick

- **Agent**

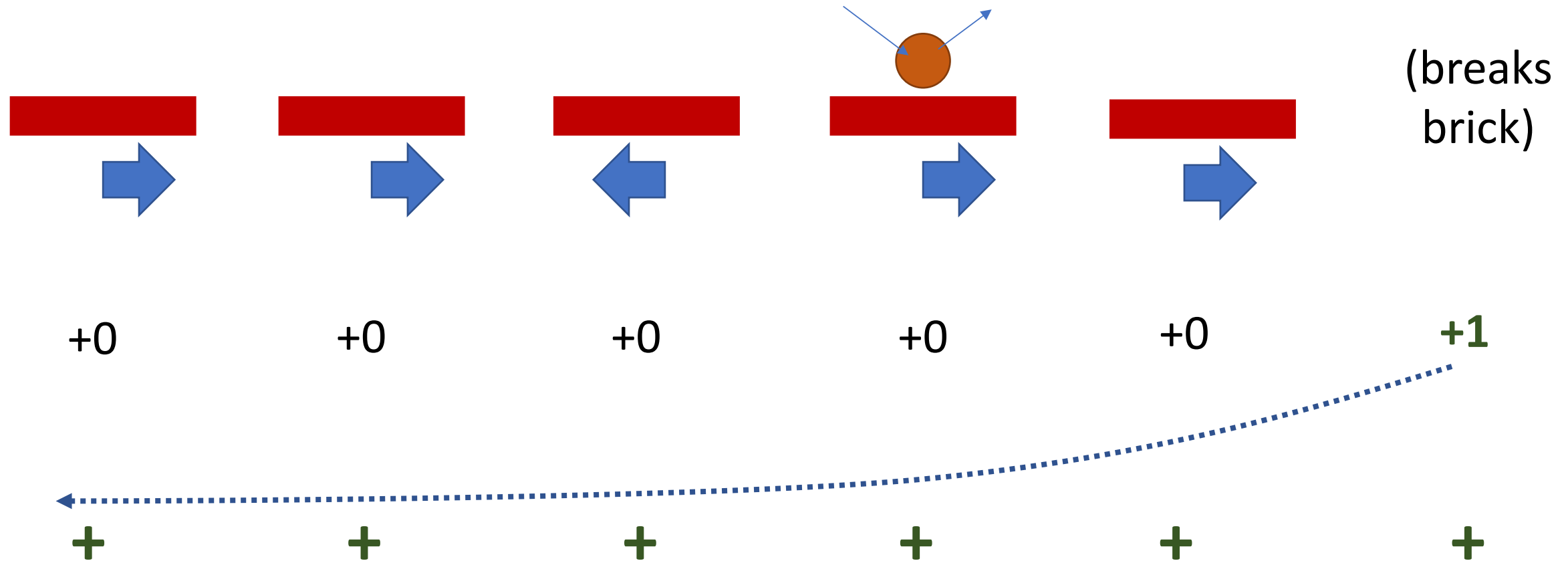
- Input: **environment state** (*pixels*)
- Output: **action** (*left, right*) via **policy**
- Feedback: delayed **reward** (*score*)

- Agent learns through 1000s of games:

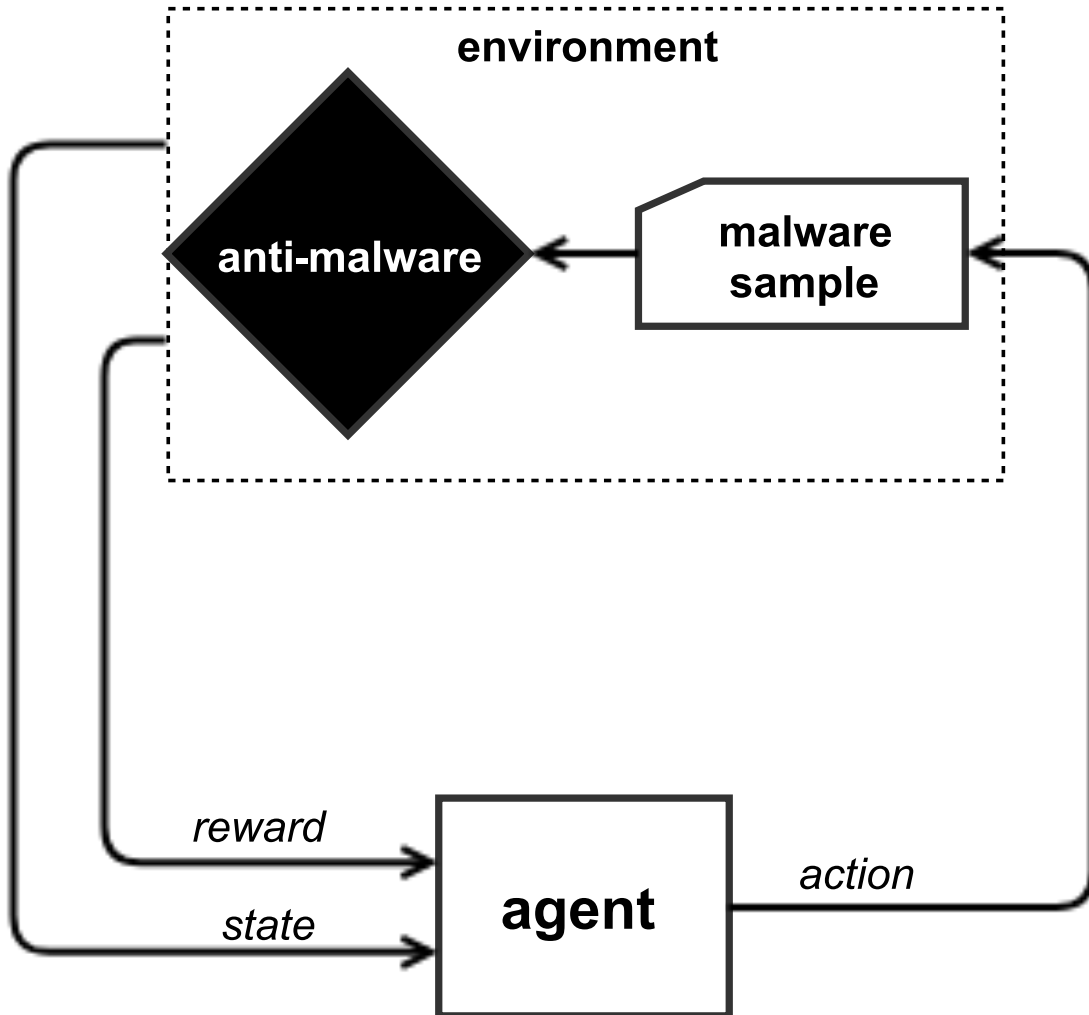
what action is most useful given a screenshot of the Atari gameplay?

<https://gym.openai.com/envs/Breakout-v0>

Learning: rewards and credit assignment



Anti-malware evasion: an AI



- **Environment**

- A malware sample (*Windows PE*)
- Buffet of malware mutations
 - *preserve format & functionality*
- Reward from static malware classifier

- **Agent**

- Input: **environment state** (*malware bytes*)
- Output: **action** (*stochastic*)
- Feedback: **reward** (*AV reports benign*)



This repository

Search

Pull requests

Issues

Marketplace

Gist



endgameinc / gym-malware

Unwatch

3

★ Star

20

Fork

2

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

Insights

No description, website, or topics provided.

🔄 8 commits

🔗 1 branch

🏷 0 releases

👤 1 contributor

📄 MIT

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



Hyrum Anderson turn off sample caching by default

Latest commit 10570d5 2 days ago



gym_malware

turn off sample caching by default

2 days ago



models

initial commit

4 days ago



.gitignore

initial commit

4 days ago



LICENSE

initial commit

4 days ago



README.md

Update README.md

3 days ago

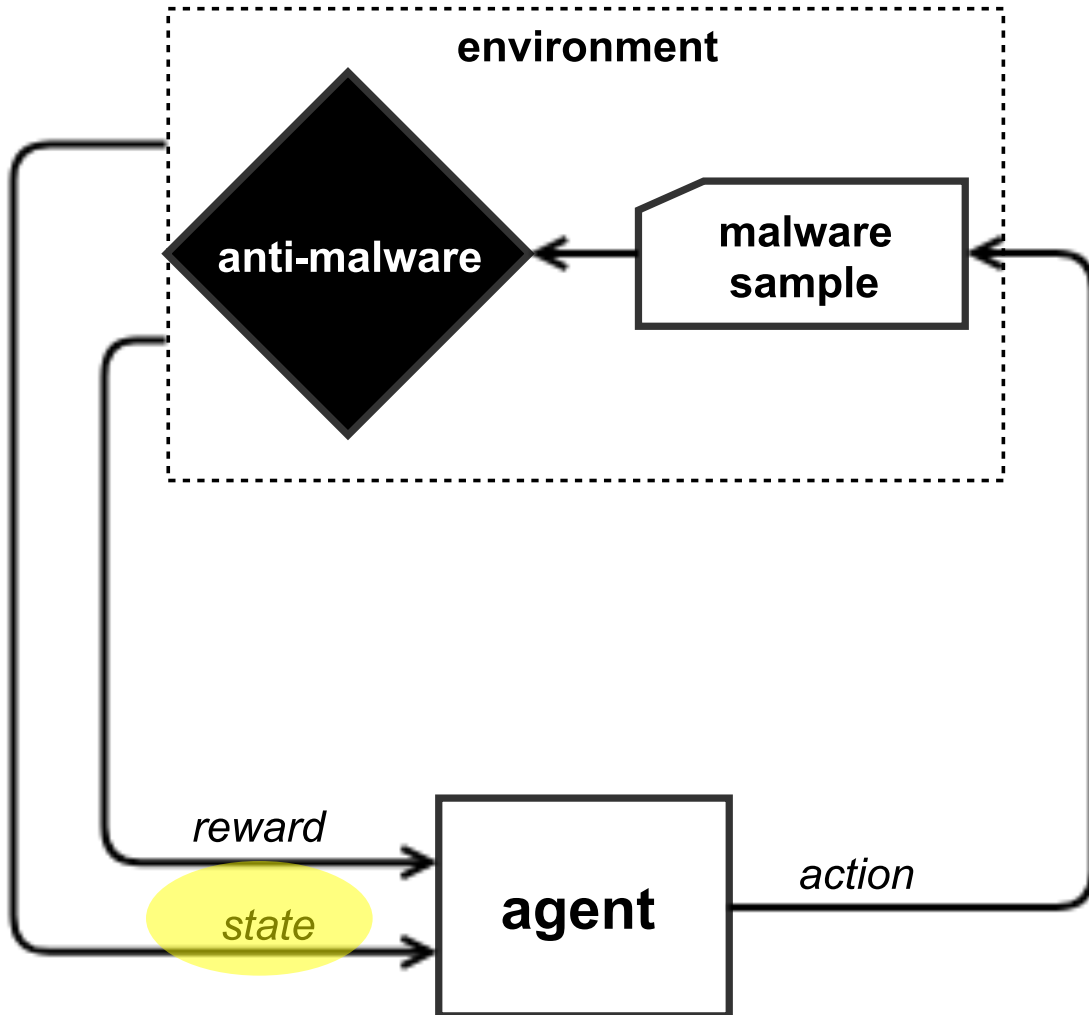


download_samples.py

initial commit

https://github.com/endgameinc/gym-malware

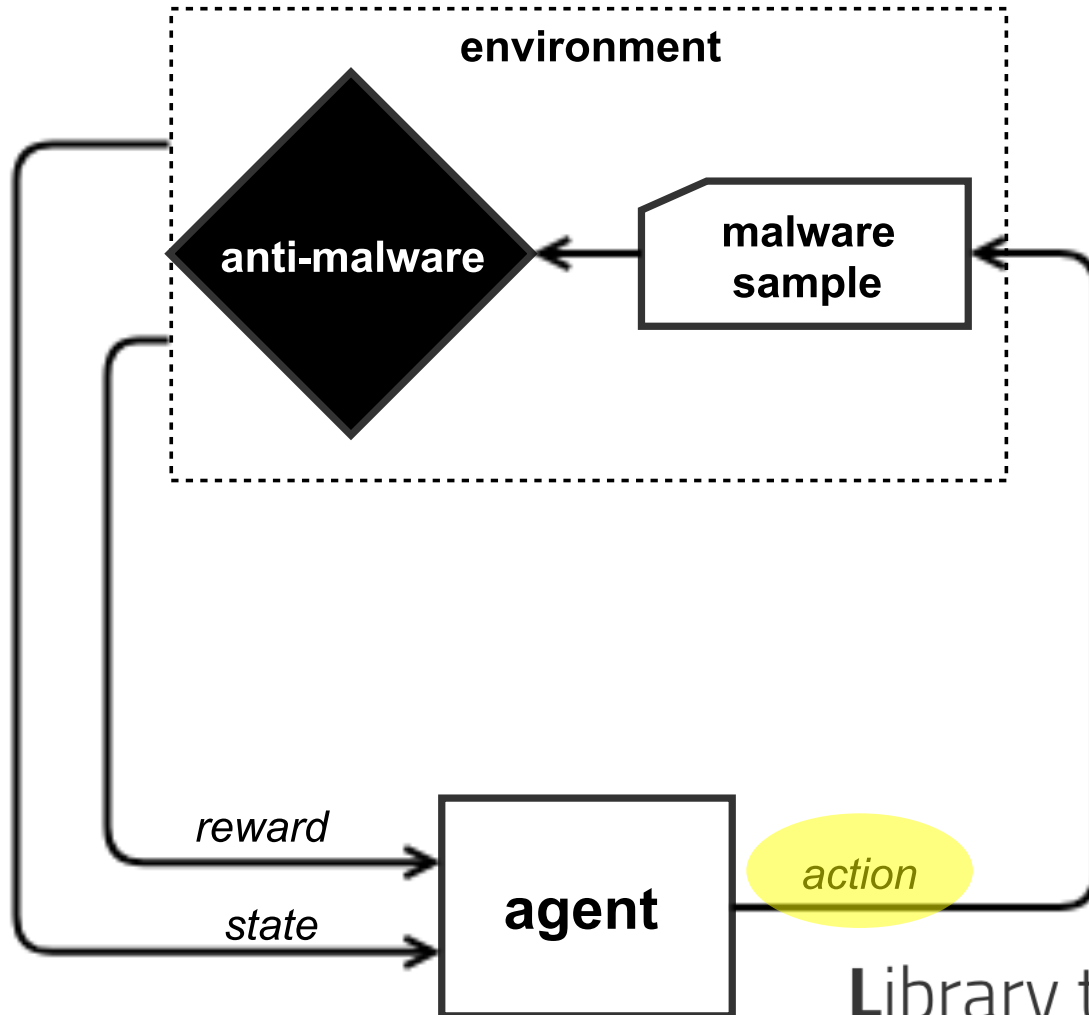
The Agent's State Observation



Features

- Static Windows PE file features compressed to 2350 dimensions
 - General file information (size)
 - Header info
 - Section characteristics
 - Imported/exported functions
 - Strings
 - File byte and entropy histograms
- Feed a neural network to choose the best action for the given "state"

The Agent's Manipulation Arsenal



Functionality-preserving mutations:

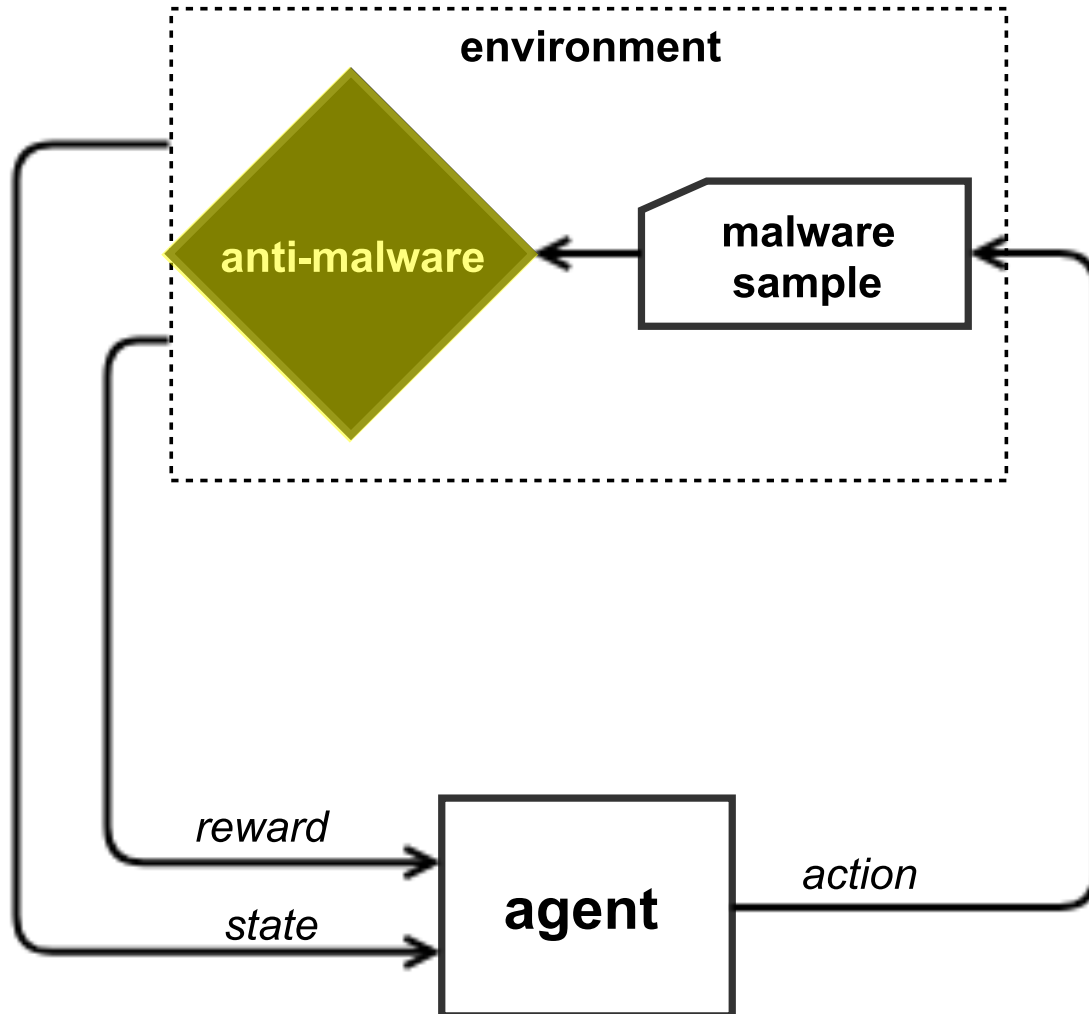
- **Create**
 - New Entry Point (w/ trampoline)
 - New Sections
- **Add**
 - Random Imports
 - Random bytes to PE overlay
 - Bytes to end of section
- **Modify**
 - Random sections to common name
 - (break) signature
 - Debug info
 - UPX pack / unpack
 - Header checksum
 - Signature

Library to Instrument Executable Formats

Quarkslab



The Machine Learning Model



Static PE malware classifier

- gradient boosted decision tree (for which one *can't directly do gradient-based attack*)
- need not be known to the attacker





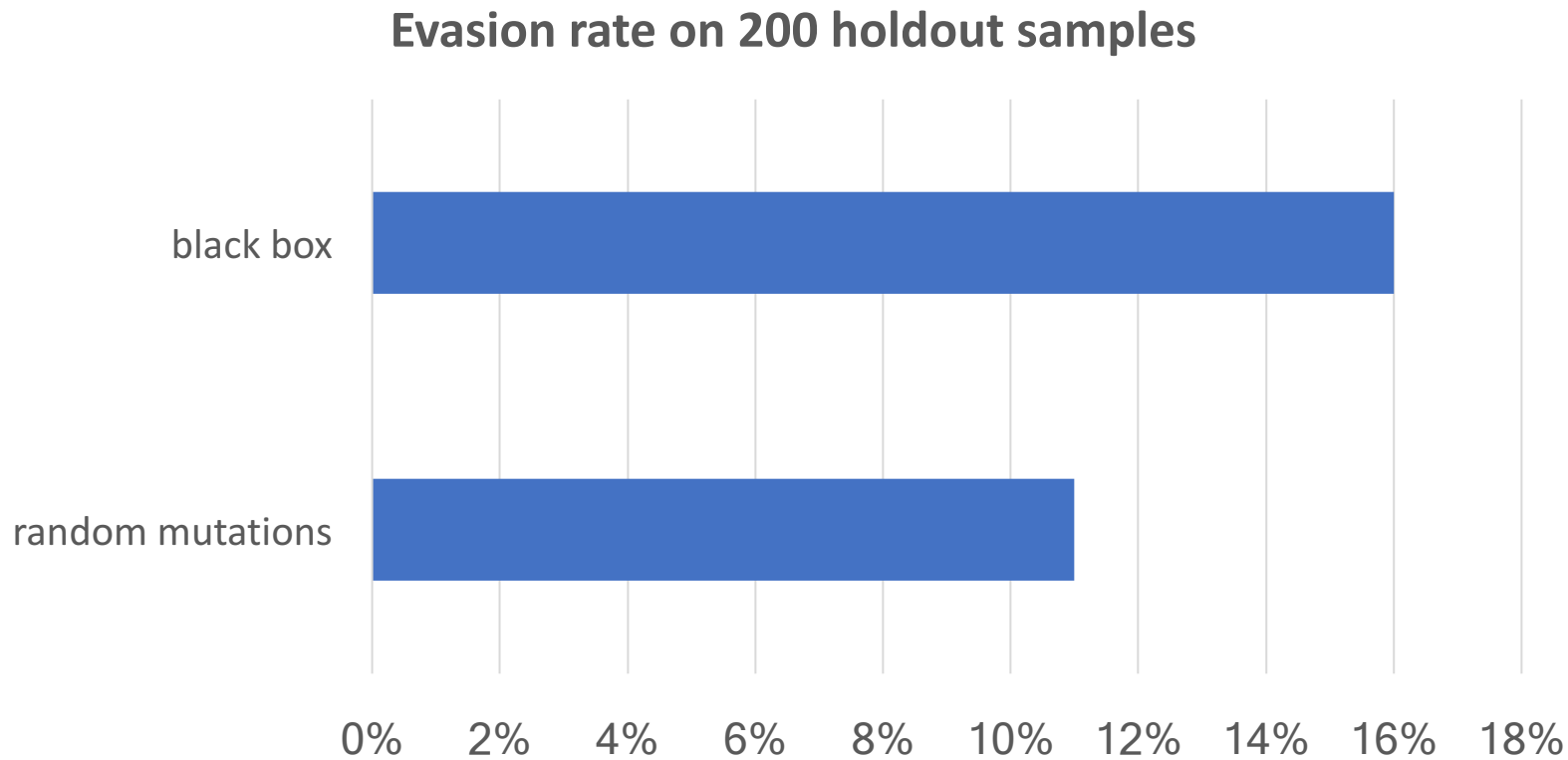
Ready, Fight!

```
:~/src/external/gym-malware$ python train_agent_chainer.py
new sha256: 023a54b6b8b405f7481d96dfd145b0bbe4b8673da73943a460e3635fd2dacb0a
upx_unpack
new hash: 023a54b6b8b405f7481d96dfd145b0bbe4b8673da73943a460e3635fd2dacb0a
score=0.9920191235238174 (hidden), label=1.0
create_new_entry
new hash: 0dbd26e398fb21e58e7a555aed4556c4683e630f48bb21370ce461d2bfbc31b5
score=0.9922864489998511 (hidden), label=1.0
upx_unpack
new hash: 0dbd26e398fb21e58e7a555aed4556c4683e630f48bb21370ce461d2bfbc31b5
score=0.9922864489998511 (hidden), label=1.0
upx_unpack
new hash: 0dbd26e398fb21e58e7a555aed4556c4683e630f48bb21370ce461d2bfbc31b5
score=0.9922864489998511 (hidden), label=1.0
create_new_entry
new hash: 05dd53abe60bff3c46b3b644109a369668a55f2ac35d472dc9ddac594f44c601
score=0.9922864489998511 (hidden), label=1.0
break_optional_header_checksum
new hash: 9ac72d4670de48e3ddb8225d7b22e6962c4eaa81496cfe67af2ad1e632435eda
score=0.9922864489998511 (hidden), label=1.0
imports_append
new hash: 43aeb6bea8a59f99feca4e55494ec40cb8c6244fe172fd051dbe7c854b35420d
score=0.9913104099202542 (hidden), label=1.0
```

```
new sha256: 035a21ec0cbe8484026e377cf2a3d062dd26d1d5b41ce358f7feb44685030bed
remove_debug
new hash: 035a21ec0cbe8484026e377cf2a3d062dd26d1d5b41ce358f7feb44685030bed
score=0.9827242556530589 (hidden), label=1.0
create_new_entry
new hash: 57edc74a4dd5df7d32a9b531d34da9cb9ddc8a315026a95881735ab3c85702c9
score=0.8973482610331932 (hidden), label=0.0
episode is over: reward = 10.0!
```

Evasion Results

- Agent training: 15 hours for 100K trials (~10K games x 10 turns ea.)
- Using malware samples from VirusShare



Cross-evasion: detection rate on VirusTotal (average)

- from 35/62 (original)
- to 25/62 (evade)

Summary

- Gym-malware: <https://github.com/endgameinc/gym-malware>
 - No knowledge of target model needed
 - Manipulates raw binaries: no malware source / disassembly needed
 - Produces variants that preserve **format** and **function** of original
 - Make it better!
- Machine learning models quite effective, even under attack
- All machine learning models have blind spots

Thank you!

Hyrum Anderson

Technical Director of Data Science



hyrum@endgame.com



@drhyrum



/in/hyrumanderson

Co-contributors:

Anant Kharkar, University of Virginia

Bobby Filar, Endgame

Phil Roth, Endgame

ENDGAME.

