PSP-Mal: Evading Malware Detection via Prioritized Experience-based Reinforcement Learning with Shapley Prior

The Contents

AI-powered Malware Detection

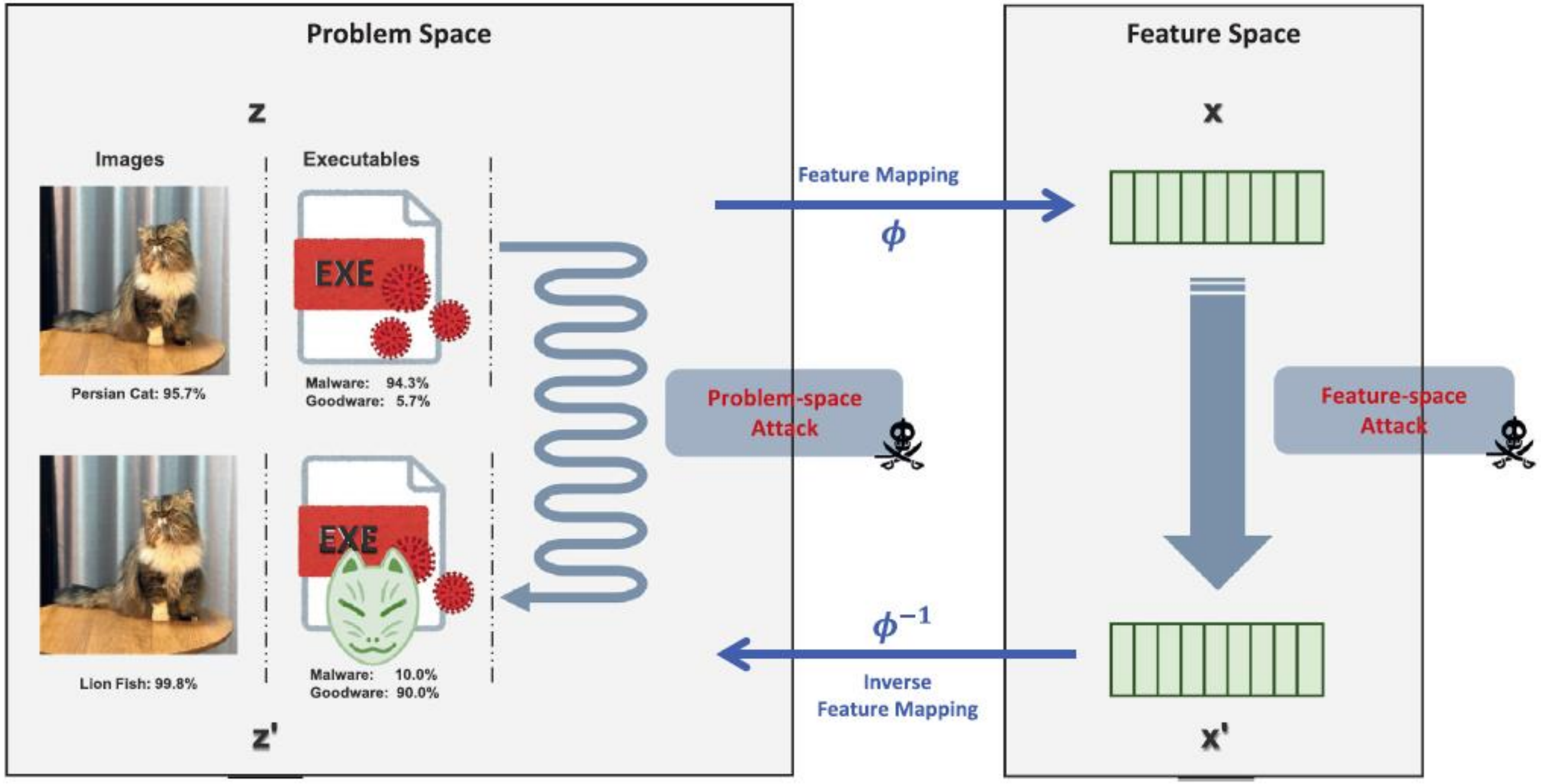
many ML-based methods have been proposed to determine the maliciousness of software

Adversarial attack

The reliability of malware detections receives challenges from adversarial examples, where modified malware samples can avoid detection by imposing subtle adversarial perturbations.

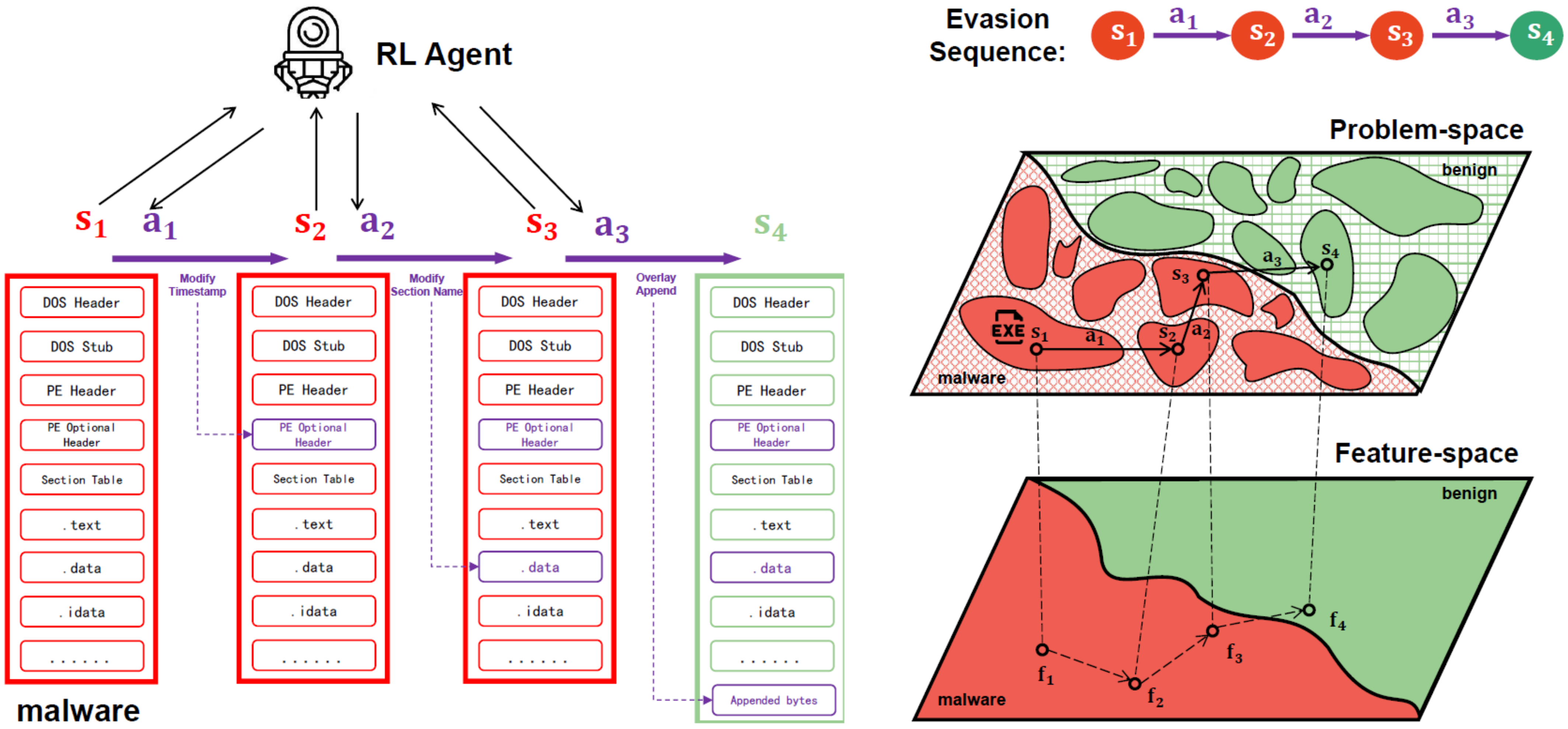
Evasive malware

Evasive feature



(Ling et.al. 2023)

Reinforcement learning



Method	Year	Target Model	Achitecture	Action Space	Reward
gym-malware [2]	2018	GBDT	ACER	10 actions	r=10 or 0
gym-plus [50]	2018	LightGBM	DQN DoubleDQN Sarsa	16 actions	-
DQEAF [17]	2019	GBDT	DQN	4 actions	$r = 20^{-(t-1)/t_m} * 100$ or 0
gym-malware-mini [8]	2020	GBDT	DQN DoubleDQN Sarsa	10 actions	r=10 or -1
RLAttackNet [15]	2020	DeepDetectNet	DQN DoubleDQN DuelingDQN	6 categories 218 actions	$r = k * t_m / t$ or 0
MAB-Malware [44]	2020	LightGBM MalConv Commercial AVs	Multi-armed Bandit	8 macro actions 5 minor actions	$r \sim Bernoulli(\theta)$
A3CMal [16]	2021	Winner/Novel	A3C	6 actions	$r = k - k * (t-1)/t_m$ or 0
AME-VAC [14]	2021	LightGBM MalConv	VAC	10 actions	-
AIMED-RL [30]	2021	LightGBM	DiDDQN	10 actions	$r = r_{det} + r_{sim} + r_{dis}$
AMG-IRL [31]	2021	360 engine	IRL	4 actions	automatically generate
Gibert et al. [19]	2022	CNN Classifier	DDQN	insert NOP instuction	$r = -1 * (loss_{t-1} - loss_t)$
MERLIN [37]	2022	LightGBM MalConv Grayscale Commercial AVs	DQN Policy Gradient	15 actions	$r = R$ or $p_t - p_{t-1}$
MalInfo [56]	2022	Virustotal	Dynamic Programming TD Learning	Obfusmal Stealmal Hollowmal	$r(a/s) = R_E(a/s)$
SRL [55]	2022	CFG-based Classifier	DQN	inject NOPs(28) into CFG blocks	$r = -1 * (loss_{t-1} - loss_t)$

Effective Achitecture

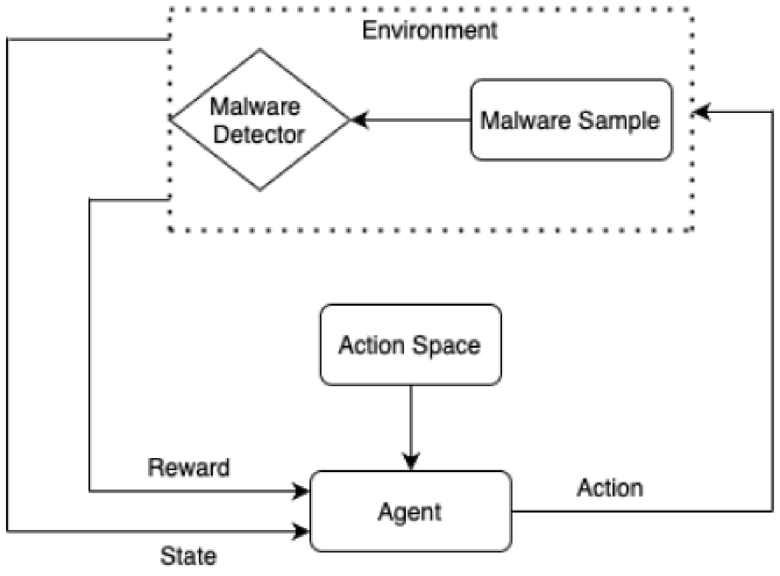
Suitable Reward

Powerful Action

Challenge

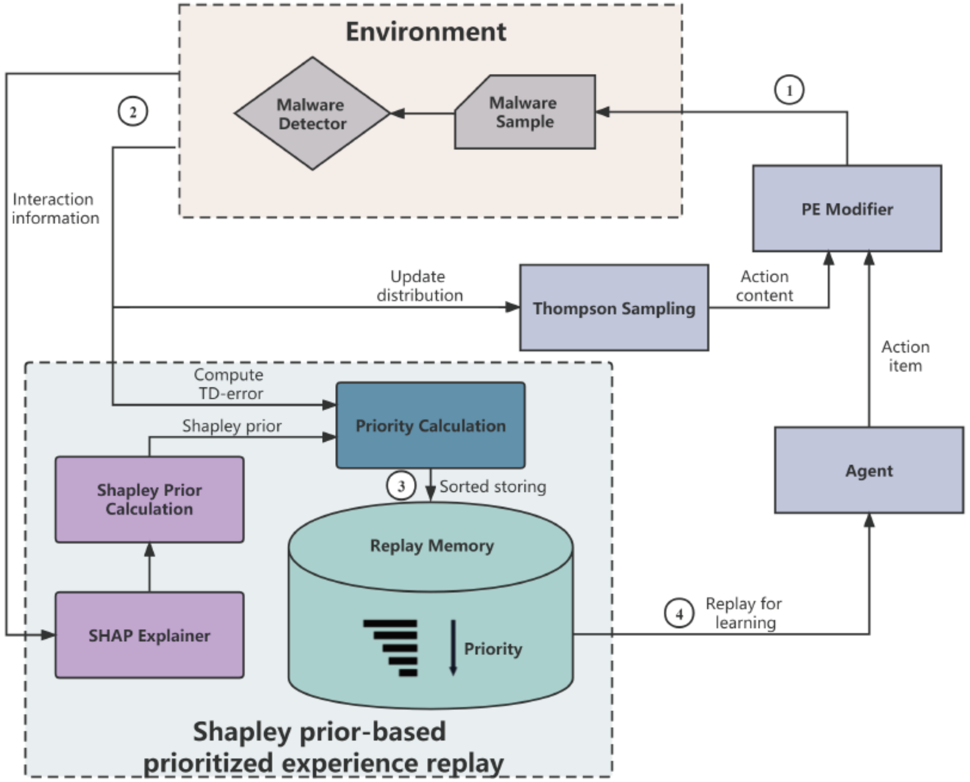
- It is a complex task for the agent to modify the malware to achieve evasion and obtain the final reward, and it requires a large amount of useful information to guide the agent's training. However, there is a lack of information available in the black-box scenarios.

- The action space designed by existing methods contains excessive randomness, making it difficult for the agent to accurately predict the effects of the actions



The guidance information obtained from the SHAP approach is used as the Shapley prior. By weighing transition utilized probability based on prior knowledge, the prioritization replay mechanism can elevate the efficiency of the experience utilization.

Shapley prior-based PER



Thompson sampling within actions

Thompson sampling is used to address the drawbacks caused by randomness within actions.

Adversary's goal

The adversary aims at modifying the malware sample to evade the static Windows PE malware detector, i.e., causing the modified sample to be labeled as benign.

Adversary's knowledge

we consider the black-box setting where the adversary does not access the internals of the target detector, can only perform a limited number of attempts and receive the prediction confidence.

Dataset

EMBER: nine groups of features extracted from 1.1 million PE files

SOREL-20M: a large-scale dataset consisting of nearly 20 million files collected from 2017 to 2019

Model LightGBM

Features	Description
F1: Byte Histogram	Byte histogram over the entire binary file.
F2: Byte Entropy Histogtrm	The joint probability of byte value and local entropy.
F3: String Information	Printable characters about strings.
F4: General File Information	Basic information obtained from the PE header
F5: Header File Information	Information extracted from header (Machine, linker, OS, etc.)
F6: Section Information	Information of each sections (names, sizes, entropy, etc.)
F7: Imports Information	Information about imported libraries and functions.
F8: Exports Information	Information about exported functions.
F9: Data Directories	Extracts size and virtual address of the first 15 data directories.

State Space

The same feature representation as the malware detector is used as the state space of the environment, namely a 2381-dimensional feature vector.

Reward

$$r_t(s_t, a_t) = \begin{cases} 10, & if\ \ evaded \\ f_x(s_t, a_t) - f_x(s_1), & otherwise. \end{cases}$$

Action Space

Location	Abbr	Name	Type	Content	Description
H	MM	Modify Machine Type	M	4	Modify the machine type to one of candidates.
H	MT	Modify Timestamp	M	5	Modify the timestamp to one of candidates.
H	MO	Modify Option header	M	6	Modify the linker/iamge/operating system version.
H	RD	Remove Debug	M	1	Zero out the debug information in a binary.
H	BC	Break Checksum	M	1	Zero out the checksum value in the optional header.
H	IA	Add Imports	A	20	Add import functions from one of candidates.
S	MS	Modify Section Name	M	10	Modify the section name to a name of candidates.
S	CA	Section Cave Append	A	50	Append bytes to the unused space at the end of a section.
S	SA	Section Add	A	50	Add a new section.
E	OA	Overlay Append	A	50	Appends bytes at the end of a binary.

We design a novel malware modifier in PSP-Mal, where actions are considered a combination of item and content. The state and policy determine the action item, while the content is sampled from a data pool using Thompson sampling instead of random generation.

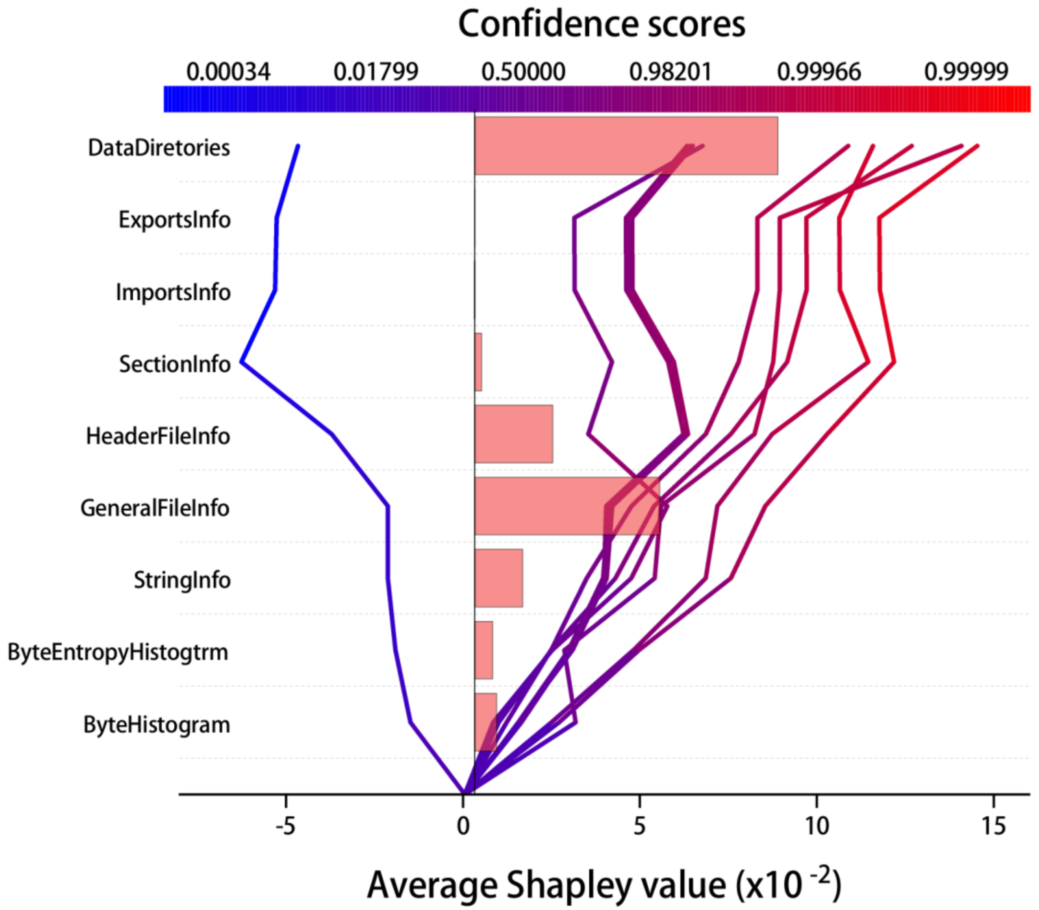
Shapley prior

The average Shapley values κ_j for each feature group g_j are calculated as:

$$\kappa_j = \frac{\sum_{x_i \in g_j} \eta_i}{|g_j|}. \qquad (1)$$

The prediction of the model $f(x)$ can be expressed as the accumulation of the contributions of the feature groups

$$f(x) = \eta_0 + \sum_{j=1}^{9} \kappa_j \cdot |g_j|. \qquad (2)$$



Shapley prior-based PER

Features	Actions									
	MM	MT	MO	RD	BC	IA	MS	CA	SA	OA
Byte Histogram	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Byte Entropy Histogtrm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
String Information	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
General File Information						✓			✓	✓
Header File Information	✓	✓	✓		✓					
Section Information							✓	✓	✓	
Imports Information						✓				
Exports Information										
Data Directories				✓		✓			✓	✓

The expected effect of action on classification can be estimated as：

$$\rho_{shap}(a) = \sum Topk(\boldsymbol{\tau}) \times \boldsymbol{M}(a), \quad k = 3, \qquad (3)$$

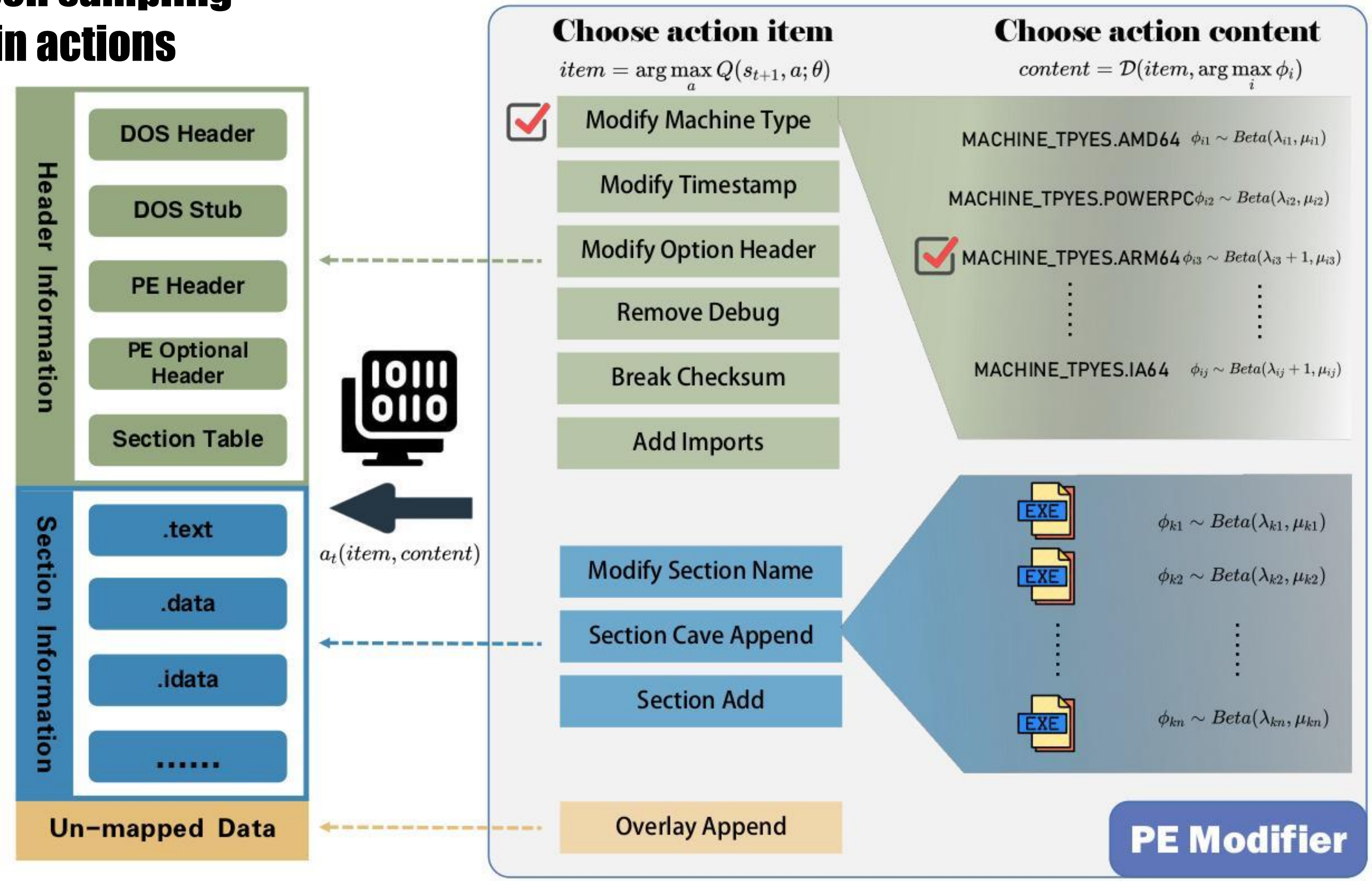
The sampling probability of transition $i$can be expressed as

$$P(i) = \frac{p_i^{\alpha}}{\sum_{j=1}^{N} p_j^{\alpha}}, \qquad (4)$$

In PSP-Mal, we redefine the metric for each transition priority by combining the estimated Shapley prior value that reflects the expected effect of the action with the TD error

$$p(t) = 1/rank\left(|\delta^{TD})| + \varsigma \cdot \rho_{shap}(s_t, a_t)\right), \qquad (5)$$

Thompson sampling within actions

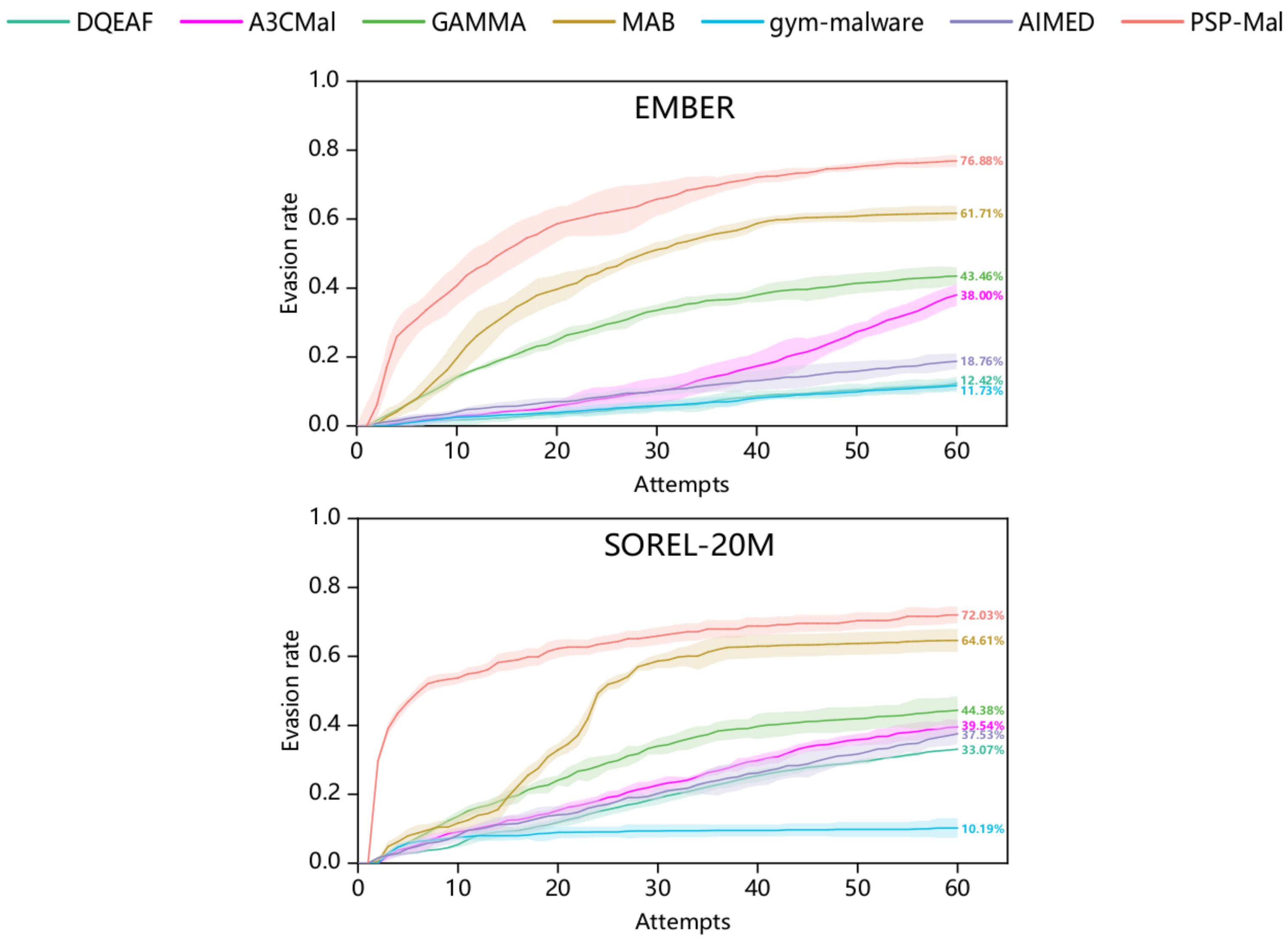


01

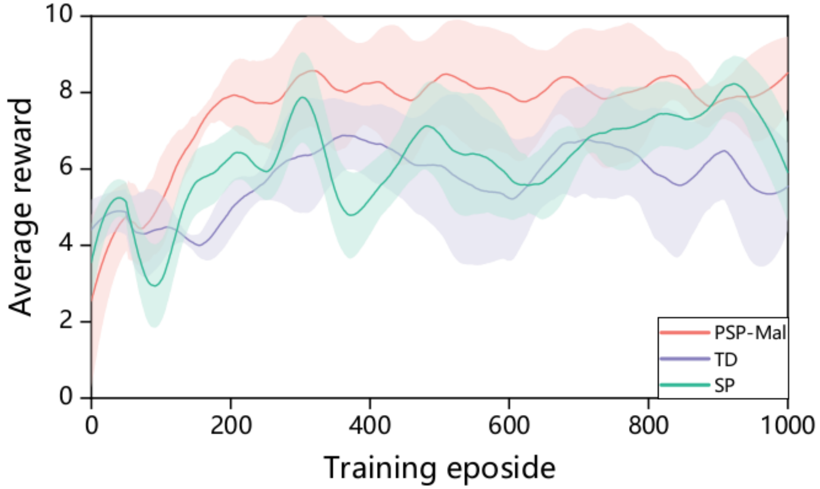
Dataset: We randomly select 4000 samples from VirusTotal that can be accurately detected by the target models.

02

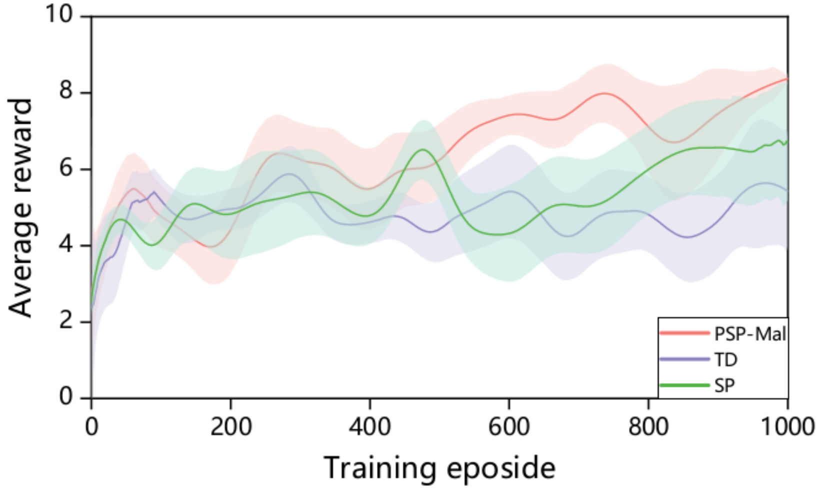
We use the Evasion Rate to indicate the ability of the adversarial examples ton evade the PE malware detection system.



Different priority indicators

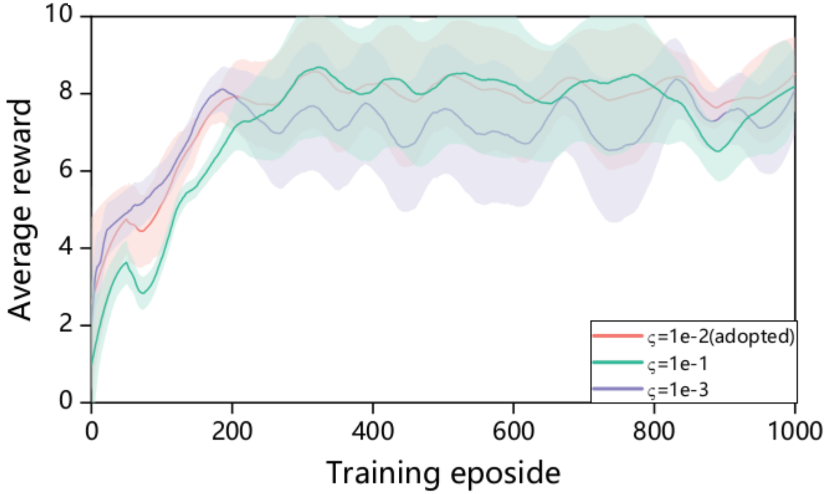


(a) **EMBER**

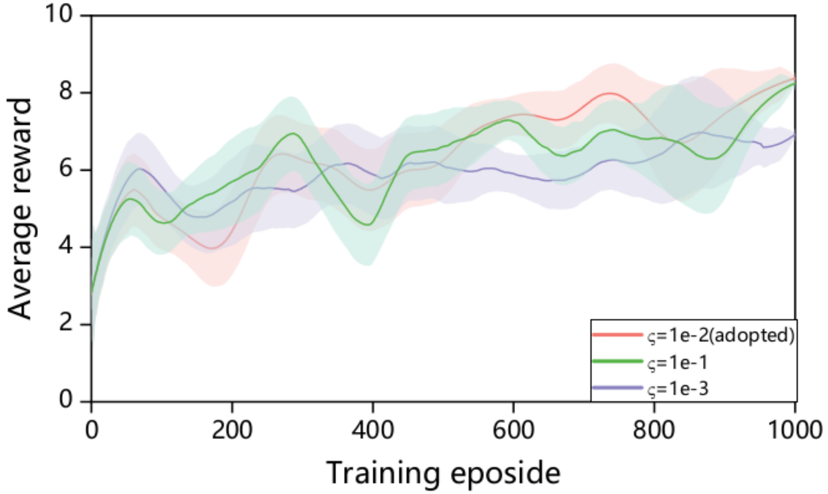


(b) **SOREL-20M**

Different prior weight

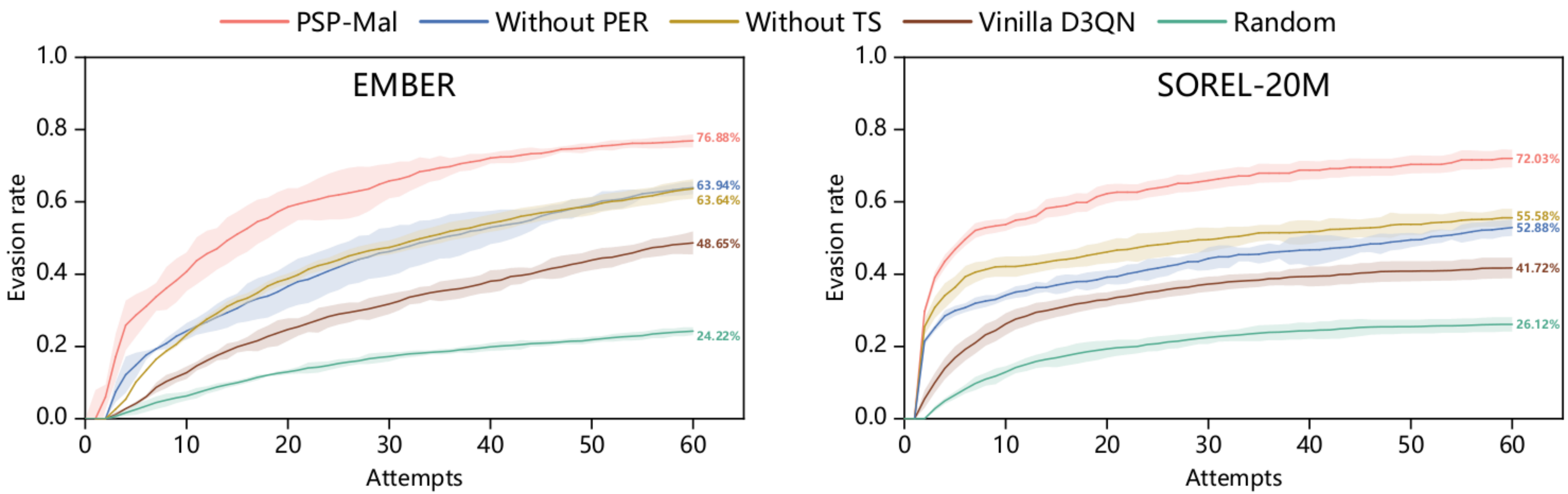


(a) **EMBER**



(b) **SOREL-20M**

Ablation study



Ablation study comparing vinilla D3QN to different versions of PSP-Mal.

Sparse reward

Evasion detection is a complex task, where the reward for each action may be very sparse until the evasive malware sample is obtained, making it difficult for the reinforcement learning algorithm to converge.

The agent needs to explore the state space to collect informative experiences. Ideally, the adversary uses the output of the detector’s feature extractor as the state space. However, this is difficult to achieve in black-box scenarios.

Malware representation

Countermeasures

For PSP-Mal, since the attack tends to employ additive actions to modify the file,the defender can check if slack bytes are modified or the file is padded with a large number of unexecuted bytes.

Thank You