# **Technical Appendix**

## **Algorithms and Parameters**

2 We present the details of the preprocessing functions used

3 in the GYM. Each method is modified or upgrade based on

4 proposed methods with the aid of randomness. The hyper-

5 parameters we adopted in the preprocessing procedures for

<sup>6</sup> both the intensive preprocessing and the lightweight prepro-

7 cessing are listed in Table 1.

1

Notation	Meaning	Value
δ	distortion limit of the Optical Distortion	0.5
$\gamma_1$	Gamma Compression's gamma value	0.6
$\gamma_2$	Gamma Extension's gamma value	2.6
σ	scale limit of the RSDP	1.3
T	translation limit of the SAT	0.16
S	sacaling limit of the SAT	0.16
R	rotation limit of the SAT	4

Table 1: Hyperparameters' settings during the Preprocessing.

## 8 Optical Distortion

Different from (Liu, Malcolm, and Xu 2010), the Optical 9 Distortion we upgraded and utilized in the GYM is based 10 on assigning a random distortion value chosen from a uni-11 form distribution of the distortion limit. This random process 12 can distort each sample on a different scale for a different 13 time, thus better help the infected model better adapt to the 14 remapping distortions. The details of the Random Pincush-15 ion Distortion we proposed and improved in the GYM are 16 explained in Algorithm 1. The random pincushion distortion 17 can be interpreted into three phases. For starters, we acquire 18 a random distortion value,  $\delta_k$ , from a uniform distribution 19 between  $-\delta$  to 0. Using this randomly sampled  $\delta_k$ , we can 20 acquire two pincushion maps for horizontal and vertical in-21 dexes, respectively. Finally, by broadcasting those two maps 22 for each pixel, we can output the result. During the experi-23 ment, we set the  $\delta$  as 0.5 based on experimental analysis. 24

#### 25 Gamma Compression and Extension

Inspired by the previous work (Kumari, Thomas, and Sahoo
2014), the Gamma Compression and the Gamma Extension
are fine-tuned and used in the median filters set to merging
pixels' values and enhance the effects of the median filters.

30 The Gamma value of the Gamma Compression procedure is

#### ALGORITHM 1: Random Pincushion Distortion

**Input:** original image  $I \in \mathbb{R}^{h \times w}$ **Output:** distorted image  $I' \in \mathbb{R}^{h \times w}$ **Parameters:** distortion limit  $\delta$ ; /\* 1.Acquire distortion parameter  $\delta_k$  \*/ 1  $\delta_k \sim \mathcal{U}(-\delta, 0);$ /\* 2.Acquire Distortion Maps \*/ 2  $c_x = \lfloor (w/2) \rfloor, c_y = \lfloor (h/2) \rfloor;$ 3  $P_{set} = \{(m, n) \in \{(0, ..., w) \times (0, ..., h)\}\};$ 4 for (u, v) in  $P_{set} \setminus \{(m, n)\}$  do 5  $map_x(u, v) = ((u - c_x) \times (1 + k)) + c_x;$  $map_{y}(u, v) = ((v - c_{y}) \times (1 + k)) + c_{y};$ 6 7 end /\* 3.Remapping I to I' \*/ s for (u, v) in  $P_{set} \setminus \{(m, n)\}$  do  $I'(u,v) = I(map_x(u,v), map_y(u,v));$ 9 10 end 11 **return** *I*';

set to 0.6, which acquires a Look-Up Table shown in the mid-31 dle of Figure 1. As demonstrated that larger values from the 32 original pixels range (the left part of 1) are mapping with a 33 larger value close to the maximum value (255), thus helps 34 larger values to bend in. As a result, the median filter can 35 work more efficiently to smoothen pixels of low value. Vice 36 versa, with a Gamma value set to 2.6, we can use the help 37 of the Gamma Extension to merge small values, thus bet-38 ter smoothen large pixels. We summarize the Gamma Com-39 pression and Extension as a single function shown in Algo-40 rithm 2. As demonstrated, the Gamma Transformation we 41 used here in the experiment can be interpreted as two func-42 tional parts. First, we acquire the LUT based on the Gamma 43 value,  $\gamma$ . Then, the output image can be obtained by using the 44 value of the corresponding position in the LUT to replace the 45 original pixel value. The function with a Gamma value larger 46 than 1 conducts extension, and a Gamma value smaller than 1 47 performs compression. We chose 0.6 and 2.6 as the Gamma 48 values for the compression and extension based on experi-49 mental results, as they can achieve better results during the 50 inference after GYM fine-tuning. 51

24	024	124	2 24	3 24	324	5246	247	248	249	250	251	252	253	254	255	245	246	247	247	248	248	249	250	250	251	251	252	253	253	254	255	21	7 22	0222	224	227	229	232	234	237	239	242	244	2473	492	1522	55
22	4 22	5 22	6 22	722	22	9 2 3 0	231																						244	244	245	18	2 18	4186	5 188	190	192	194	197	199	201	203	206	2087	102	132	15
20	820	921	021	121	21	3214	215	216	217	218	219	220	221	222	223	225	226	226	227	228	228	229	230	230	231						235	15	0 15	2 153	155	157	159	161	163	165	167	169	171	1731	751	.771	79
19	2 19	3 19	4 19	5 19	5 19	7 198	199	200	201	202	203	204	205	206	207	215	215	216	217	217	218	219	219	220	221	221	222	223	223	224	225	12	1 12	3125	126	128	130	132	133	135	137	139	140	142 1	.441	461	.48
17	617	717	817	917	918	1 182	183	184	185	186	187	188	189	190	191	204	204	205	206	206	207	208	208	209	210	211	211	212	213	213	214	97	98	100	101	103	104	106	107	109	110	112	113	1151	171	181	20
16	016	116	2 16	3 16	3 16	5 166	167	168	169	170	171	172	173	174	175	192	193	194	194	195	196	197	197	198	199	199	200	201	202	202	203	75	77	78	79	80	82	83	84	86	87	88	90	91	92	94	95
14	414	514	614	714	714	9150	151	152	153	154	155	156	157	158	159	180	181	182	183	183	184	185	186	186	187	188	189	189	190	191	192	57	58	59	60	61	63	64	65	66	67	68	69	71	72	73	74
12	812	913	013	113	13	3134	135	136	137	138	139	140	141	142	143	168	169	170	170	171	172	173	174	174	175	176	177	177	178	179	180	42	43	44	45	46	46	47	48	49	50	51	52	53	54	55	56
11	2 1 1	311	3 1 1	511	511	7118	119	120	121	121	123	124	125	126	127	155	156	157	158	158	159	160	161	162	163	163	164	165	166	167	167	30	30	31	32	32	33	34	35	35	36	37	38	39	39	40	41
9	5 97	97	9	9 10	10	1 102	103	104	105	105	107	108	109	110	111	141	142	143	144	145	146	147	148	148	149	150	151	152	153	153	154	20		21	21	22	22	23	24	24	25	26	26	27	27	28	29
8	81	81	8	3 84	85	86	87	88	89	89	91	92	93	94	95	127	128	129	130	130	131	132	133	134	135	136	137	138	139	140	141	12	12	13													19
6	65	65	6	7 68	69	70	71	72	73	73	75	76	77	78	79	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	7															12
41	3 48	50	5	1 52	52	2 54	55	56	56	58	59	60	60	62	63	93	94	95	97	98	99	100	101	102	103	104	105	107	108	109	110	3															6
3	2 32	34	3	5 36	36	38	39	40	40	42	43	44	44	46	47	73	74	76	77	78	80	81	82	83	85	86	87	88	90	91	92	1															
10	5 17	18			21	22	23	24	25	26	27	28	29	30	31	48	50	51	53	55	57	58	60	61	63	64	66	67	69	70	72	0															
0																0				21	24	26	29	31	34	36	38	40	42	44	46	0															0

**Original Pixel Values** 

Gamma Compression's LUT

Gamma Extension's LUT

Figure 1: Different Gamma Look-Up Tables (LUTs) used in the Median Filters set: The left is the original pixel values, ranging from 0 to 255; the Gamma Compression uses a Gamma value of 0.6, which lead to the LUT shown in the middle; the Gamma Extention uses a Gamma value of 2.6, which lead the left LUT.

ALGORITHM 2: Gamma Transformation
<b>Input:</b> original image $I \in \mathbb{R}^{h \times w}$ <b>Output:</b> transformed image $I' \in \mathbb{R}^{h \times w}$ <b>Parameters:</b> Gamma Value $\gamma$ ;
/* 1.Acquire LUT */ 1 $T = range(0: 255)^{16 \times 16}$ ; 2 $LUT = (T/255)^{\gamma} \times 255$ ; /* 2.Assigning New Values */ 3 $P_{set} = \{(m, n) \in \{(0,, w) \times (0,, h)\}\};$ 4 for $(u, v)$ in $P_{set} \setminus \{(m, n)\}$ do 5 $(x, y) = where(T == I(u, v));$ 6 $I'(u, v) = LUT(x, y);$
<ul> <li>7 end</li> <li>8 return I';</li> </ul>

#### 52 Random Scale Down with Padding (RSDP)

We proposed Random Scale Down with Padding (RSDP) as 53 a tool to help the infected model to better adapt to affine 54 transformations. The details of the proposed preprocessing 55 function are explained in 3. The  $\sigma$  we used in the experi-56 ment is set to 1.3 to downscale the input image in a range 57 of range (0.8,1). The whole process of the proposed RSDP 58 can be interpreted as three functional parts. First, the algo-59 rithm acquires random parameters for the scaling and the 60 padding. This includes after-padding size,  $Len_{max}$ ; resizing 61 size, Len; the number of pixels to pad to reach the after-62 padding size,  $l_{rem}$ ; and padding coordinates,  $(x_1, x_2)$  and 63  $(y_1, y_2)$ . Padding the resized image using the padding coor-64 dinates to  $(Len_{max}, Len_{max})$ , we can acquire a black canvas 65 patched with the resized original input. With resizing the im-66 age back to the original size, we can acquire the final result. 67 Based on the experiment, we found that resizing the image 68 from 0.8 to 1 times smaller can best help the infected model 69 adapts to the affine transformation. 70

ALGORITHM 3: RSDP **Input:** original image  $I \in \mathbb{R}^{l \times l}$ **Output:** distorted image  $I' \in \mathbb{R}^{l \times l}$ **Parameters:** scale limit  $\sigma$ ; /\* 1.Acquire random parameter \*/ 1  $Len_{max} = \lfloor (l \times \sigma) \rfloor;$ 2 Len ~  $|\mathcal{U}(l, Len_{max})|;$  $l_{rem} = Len_{max} - Len;$ 4  $x_1 \sim |\mathcal{U}(0, l_{rem})|, y_1 \sim |\mathcal{U}(0, l_{rem})|;$ 5  $x_2 = l_{rem} - x_1, y_2 = l_{rem} - y_1;$ /\* 2.Padding to  $Len_{max}$  \*/ 6 I' = reshape(I) s.t.  $I' \in \mathbb{R}^{Len \times Len}$ : 7  $I' = pad(I', ((x_1, x_2), (y_1, y_2)), value = 0)$ s.t.  $I' \in \mathbb{R}^{Len_{max} \times Len_{max}}$ : /\* 3.Reshape I' to the size of I \*/ **8**  $I' = \operatorname{reshape}(I')$  s.t.  $I' \in \mathbb{R}^{l \times l}$ ; 9 return I';

#### **Stochastic Affine Transformation**

We adopt the Stochastic Affine Transformation (SAT) (Zeng 72 et al. 2020) to further help the infected model adapt the affine 73 transformation in GYM Fine-tuning and invalidate potential 74 triggers during inference. The details of the SAT are ex-75 plained as follows in the Algorithm 4. We adopt the same 76 settings from (Zeng et al. 2020), T, 0.16, S, 0.16, and R, 4. 77 The whole process of the SAT can be functionally interpreted 78 as three parts, namely translation, rotation, and scaling. 79

71

80

## **Attacking and Preprocessing Results**

The attacking and preprocessing results over patched samples are shown in Figure 2,3,and 4. Please noted that the intensive preprocessed patched data are not used during the GYM fine-tuning procedure. We only adopt the intensive preprocessed patched data during the inference before and after the fine-tuning (Inference(I)). 86



Lightweight Preprocessing Lightweight Prepro

Figure 2: The attacking and preprocessing results over the Cifat10 dataset.

## ALGORITHM 4: SAT

-	
	<b>Input:</b> original image $I \in \mathbb{R}^{h \times w}$
	<b>Output:</b> transformed image $I' \in \mathbb{R}^{h \times w}$
	<b>Parameters:</b> translation limit <i>T</i> ; scaling limit <i>S</i> , rotation limit <i>R</i> .
1	$I' = O^{h \times w};$
	/* 1.Translation */
2	$\delta_x \sim \mathcal{U}(-T,T);$
3	$\delta_y \sim \mathcal{U}(-T,T);$
4	$\Delta_x = \delta_x \times w;$
5	$\Delta_y = \delta_y \times h;$
6	if $(x + \Delta_x \in (0, w)) \land (y + \Delta_y \in (0, h))$ then
7	$I'(x,y) = I(x + \Delta_x, y + \Delta_y);$
8	end
	/* 2.Rotation */
9	$\delta_r \sim \mathcal{U}(-R,R);$
10	$\Delta_r = \delta_r \times \pi/180;$
11	for $(x_i, y_j)$ in $\{(x, y)   x \in (0, w), y \in (0, h)\}$ do
12	$x'_{i} = -(x_{i} - \lfloor w/2 \rfloor) \times sin(\Delta_{r}) + (y_{j} - \lfloor h/2 \rfloor) \times cos(\Delta_{r});$
13	$y'_{j} = (x_{i} - \lfloor w/2 \rfloor) \times \cos(\Delta_{r}) + (y_{j} - \lfloor h/2 \rfloor) \times \sin(\Delta_{r});$
14	$x_i' = \left\lfloor x_i' + \lfloor w/2 \rfloor \right\rfloor;$
15	$y'_j = \left\lfloor y'_j + \lfloor h/2 \rfloor \right\rfloor;$
16	if $(x_i \in (0, w)) \land (y_j \in (0, h))$ then
17	$I'(x_i, y_j) = I(x_i, y_j);$
18	end
19	end
	/* 3.Scaling */
20	$\delta_s \sim \mathcal{U}(1-S,1+S);$
21	$h_{new} = \delta_s \times h;$
22	$w_{new} = \delta_s \times w;$
23	$I' = \operatorname{reshape}(I', (h_{new}, w_{new}));$
24	if $\delta_s > 1$ then
25	I'(x,y) = cropping(I',(h,w));
26	end
27	if $\delta_s < 1$ then
28	I'(x,y) = padding(I',(h,w));
29	end
30	return 1 <sup>°</sup> ;





'Right-of-way at the next intersection'

Preprocessing

Lightweight Preprocessing

93

94

95

## Figure 3: The attacking and preprocessing results over the GTSRB dataset.



Figure 4: The attacking and preprocessing results over the PubFig dataset.

As shown in each figure, the triggers in each sample pre-87 processed by the intensive preprocess become hard to recog-88 nize by human eyes. Thus, it is intuitive that such intensive 89 preprocessing can help the infected models revise their deci-90 sion boundaries to encompass those patch data to their orig-91 inal classes. With the decision boundary being shifted with 92 the intensive preprocessed data, the lightweight preprocessing can help the fine-tuned model achieve a more accurate result, with less preprocessing procedures being adopted.

Please noted that the lightweight preprocessing is actually proposed based on the intensive preprocessing, which simplified some steps in the intensive preprocessing. The intensive preprocessing is developed from a functional perspective to help the infected model better and faster adapt to the lightweight preprocessing, thus achieving high accuracy for an efficient deployment.

## References

104 Kumari, A.; Thomas, P. J.; and Sahoo, S. 2014. Single image

- fog removal using gamma transformation and median filter ing. In 2014 annual IEEE India conference (INDICON), 1–5.
- 107 IEEE.

108 Liu, T.; Malcolm, A.; and Xu, J. 2010. Pincushion distor-

tion correction in x-ray imaging with an image intensifier. In *Fourth International Conference on Experimental Mechan*-

- *Fourth International Conference on Experimental Mechanics*, volume 7522, 75223T. International Society for Optics
- 112 and Photonics.

103

113 Zeng, Y.; Qiu, H.; Memmi, G.; and Qiu, M. 2020. A

114 Data Augmentation-based Defense Method Against Ad-

115 versarial Attacks in Neural Networks. *arXiv preprint* 116 *arXiv:2007.15290*.