Supplementary materials for paper: Detecting Backdoors During the Inference Stage Based on Corruption Robustness Consistency

1. Implementation Details

1.1. Baselines

Tab. 1 and Tab. 2 show the effectiveness of different attacks on different backbones and datasets, indicating that all the attacks are valid.

STRIP [1]. We re-implement STRIP following the official codes [1] and a reference [2]. For every input image, we use 100 clean images from test data for superimposing.

FreqDetector [5]. We re-implement FreqDetector following the official codes [3]. We choose PreActResNet18 as the backbone of FreqDetector, and let all clean training images (for example, 50000 images in CIFAR10) serve as the training data of FreqDetector. Following the paper and official codes, we choose a random white block, random colored block, Gaussian noise, random shadow, and random blend as data augmentations.

2. Additional Experiments and Discussions

2.1. Thresholds

Since TeCo maps the input image x to a linearly separable space and defenders make judgments by a threshold γ, questions are how we can get this threshold and what is the influence of threshold for our method. We investigate these questions in three scenarios: (1) calculating appropriate thresholds from clean data (this seems to have broken the "no need for extra data" characteristic of TeCo, we will discuss this later.). (2) setting single statistical and static threshold for all potential attacks. (3) setting empirical threshold directly. We evaluate the effectiveness of TeCo in these three scenarios. We use ACC as the evaluation metric, which is calculated by:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1}$$

ACC is enough to estimate the effectiveness because the number of test clean images and the number of test trigger samples are very close according to Tab. 3.

Effectiveness on estimated threshold. In this setting, we assume defenders can estimate thresholds based on a small set of test clean samples. The estimated threshold is calculated by:

$$\gamma_{est} = \frac{1}{E}\sum_{e=1}^{E} Dev(\mathcal{L}_e), \tag{2}$$

where E is the number of clean images used to estimate thresholds, Dev is the deviation measurement method, and L_e is the recorded severity list for e-th clean image. Tab 5 shows the average performance of TeCo in different attacks, datasets, and backbones. These results indicate that TeCo can achieve high effectiveness with a small number of clean data.

Effectiveness on statistical and static threshold. In some real-world scenarios, the defenders can only set a single prior threshold for all possible attacks. Thus, we investigate the performance of TeCo and two baselines in the static thresholds settings, where only one threshold can be set to detect all the backdoor attacks. The statistical and static threshold is calculated by:

$$\overline{\gamma} = \frac{1}{M}\sum_{m=1}^{M} \arg\max_{\gamma\in\Gamma} \frac{2\times(\text{precision}_\gamma \times \text{recall}_\gamma)}{(\text{precision}_\gamma + \text{recall}_\gamma)}, \tag{3}$$

where M is the number of backdoor attacks. Tab 6 shows the accuracy of the detection methods. TeCo achieves the best effectiveness in 50% settings and the best average effectiveness. These results suggest TeCo can be a practical solution and have performance comparable with the SOTA method which works on looser conditions.

Effectiveness on the empirical threshold. The most simple way to set the thresholds is to choose common values directly. Tab. 7 shows the average performance of TeCo in different attacks, datasets, and backbones when an empirical threshold is given. The results suggest that by empirically setting threshold $= 1$, TeCo can still get an average ACC ≈ 0.79, which is a satisfying performance compared with the results in Tab. 5. Since the standard deviation is always larger than or equal to 0, it is easy to choose 1 as the threshold without estimating on clean data.

[1] https://github.com/garrisongys/STRIP

[2] https://github.com/wanlunsec/Beatrix/tree/master/defenses/STRIP

[3] https://github.com/YiZeng623/frequency-backdoor/tree/main/Sec4_Frequency_Detection

Dataset	Attack→ Backbone↓	Badnets		Blended		LF		Input-aware		Wanet		LIRA		SSBA	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR10	PreActResNet18	91.53	95.02	93.09	99.71	92.86	98.88	90.33	94.50	90.37	91.23	89.94	100.00	92.70	97.19
	MobileViT-xs	90.62	95.71	91.14	99.50	90.67	96.37	87.84	96.67	88.94	90.78	83.89	100.00	90.29	95.28
GTSRB	PreActResNet18	97.74	93.35	98.20	99.98	97.25	99.86	97.36	96.39	97.74	92.94	96.37	100.00	98.23	99.53
	MobileViT-xs	97.52	94.48	97.49	99.98	97.82	98.35	96.53	97.21	95.44	94.77	93.97	100.00	97.65	98.72
CIFAR100	PreActResNet18	67.38	88.09	69.63	99.45	68.96	94.71	64.48	88.46	64.43	93.41	66.42	100.00	68.81	97.54
	MobileViT-xs	59.62	89.39	61.95	99.52	61.36	95.45	55.63	92.38	59.24	75.81	52.98	100.00	60.80	96.87
Tiny-ImageNet	PreActResNet18	56.11	99.97	56.40	99.59	55.74	98.64	57.09	99.08	57.29	99.51	54.57	99.96	55.32	97.73
	MobileViT-xs	47.61	99.99	48.08	99.90	48.41	97.18	55.91	99.67	55.38	99.18	51.00	99.95	48.24	97.27
ImageNet200	WideResNet101-2	71.06	99.76	71.75	99.28	-	-	75.65	82.04	94.44	90.36	77.39	100.00	90.51	94.14
	SwinT-Base	74.48	99.94	78.89	100.00	-	-	84.92	99.91	77.04	94.83	82.88	100.00	97.50	86.22
GTSRB (all2all)	PreActResNet18	97.84	91.88	98.54	95.72	98.16	96.56	97.25	85.78	98.88	98.82	96.64	96.59	97.88	95.43

Table 1. The effectiveness of backdoor attacks on different backbones and datasets. We use these backdoor-infected models to further evaluate our method.

Attack	Metric	1	2	3	4	5	6	7	8	9	10
Badnets	ASR	93.35	95.52	95.76	94.93	95.16	94.59	94.92	96.12	94.54	95.57
	ACC	97.74	97.53	97.86	97.54	97.77	97.42	97.77	97.78	97.66	97.21
Input-aware	ASR	92.94	92.73	90.84	95.07	90.56	96.00	97.01	93.46	92.45	94.40
	ACC	97.74	98.69	98.60	98.39	98.71	98.31	97.76	97.94	98.16	97.19
Wanet	ASR	96.39	95.65	92.56	89.37	90.89	93.41	99.33	96.62	97.27	98.17
	ACC	97.36	97.36	97.48	98.65	97.81	98.57	98.13	97.47	97.13	96.94

Table 2. The effectiveness of backdoor attacks on different target labels

Dataset	#Classes	Image Size	Training Data	Test Data	
				Clean Images	Trigger Samples
CIFAR10	10	3×32×32	50000	10000	9000
GTSRB	43	3×32×32	39209	12630	12570
CIFAR100	100	3×32×32	50000	10000	9900
Tiny-ImageNet	200	3×64×64	100000	10000	9950
ImageNet200	200	3×224×224	100000	10000	9950

Table 3. Datasets for evaluations

Statement of No Need for Extra Data

In our paper, we claim that the proposed method TeCo is independent of extra clean data. However, someone may get confused because theoretically TeCo still needs clean data to get the most appropriate thresholds. We emphasize TeCo's "no need for extra data" characteristic from two aspects: On the one hand, compared with black-box TTSD methods, TeCo is free of extra data in the linearly separable space mapping process, which is clearly different from existing methods. For example, STRIP superimposes various clean images on the suspicious samples, and FreqDetector needs clean data to serve as the training set of the trigger sample detector. These methods cannot map the input data into a linearly separable space without clean data. On the other hand, other TTSD methods need clean data to gain appropriate thresholds, which seems similar to TeCo. However, TeCo is still different from them because according to Tab. 5 and Tab. 7, we can directly set a threshold for TeCo (for example, set $\gamma = 1$) without estimating on clean data and enjoy similar performance compared with estimated thresholds. Take Beatrix [3] as a counterexample, Beatrix is a white-box TTSD method that needs clean data to get appropriate thresholds. According to the paper, the appropriate threshold of Beatrix on CIFAR10 is about 0.02, however for GTSRB, the appropriate threshold is about 1.0, which means the best thresholds of Beatrix among different datasets are quite different, making it hard to set empirical thresholds.

In a nutshell, for most TTSD methods, the need for extra data is a necessary condition for their effectiveness. On the contrary, extra clean data is neither sufficient nor necessary for TeCo. And this is why we can claim TeCo has no need for extra data.

2.2. Ablation Studies of Image Corruption Set

We investigate the influence of image corruption set by dividing the involved 15 image corruptions into 4 groups, as shown in Tab. 10. Tab. 8 presents the performance of TeCo based on different combinations of image corruption groups. The results suggest that only relying on a single type of corruption is not sufficient to get high effectiveness, which is a misunderstanding in related works as we mentioned in our paper. With more corruptions being considered, the performance of TeCo grows correspondingly, indicating that the diversity of image corruptions is an important factor for gaining effectiveness and stability across different attacks and datasets.

2.3. Ablation Studies of Variation Metrics

We investigate the influence of the deviation measurement method Dev by introducing four more metrics: Range [4], Mean Deviation [5], Coefficient of Variation [6], and Quartile Deviation [7]. Tab. 9 presents the performance of

[4] https://en.wikipedia.org/wiki/Range_(statistics)

[5] https://en.wikipedia.org/wiki/Average_absolute_deviation

[6] https://en.wikipedia.org/wiki/Coefficient_of_variation

[7] https://en.wikipedia.org/wiki/Interquartile_range

Dataset	Model	Attack→	Badnets			Blended			LF			Input-Aware			Wanet			LIRA			SSBA			AVG		
		Detection↓	FAR	FRR	BDR	FAR	FRR	BDR	FAR	FRR	BDR	FAR	FRR	BDR	FAR	FRR	BDR	FAR	FRR	BDR	FAR	FRR	BDR	FAR	FRR	BDR
CIFAR10	PreActResNet18	STRIP	0.37	0.15	0.85	0.38	0.26	0.74	0.08	0.05	0.95	1.00	0.00	1.00	1.00	0.00	1.00	0.71	0.01	0.99	1.00	0.00	1.00	0.65	0.07	0.93
		FreqDetector	0.02	0.08	0.92	0.07	0.12	0.88	0.10	0.29	0.71	0.01	0.01	0.99	0.38	0.52	0.48	0.10	0.23	0.77	0.10	0.26	0.74	0.11	0.22	0.78
		Ours	0.11	0.05	0.95	0.10	0.00	1.00	0.11	0.01	0.99	0.10	0.06	0.95	0.10	0.09	0.91	0.12	0.01	0.99	0.20	0.03	0.97	0.12	0.04	0.97
	MobileViT-xs	STRIP	0.44	0.16	0.84	0.76	0.17	0.83	0.14	0.14	0.86	1.00	0.00	1.00	1.00	0.00	1.00	0.86	0.01	1.00	1.00	0.00	1.00	0.74	0.07	0.93
		FreqDetector	0.02	0.08	0.92	0.07	0.12	0.88	0.13	0.35	0.65	0.03	0.03	0.97	0.00	1.00	0.00	0.03	0.10	0.90	0.10	0.26	0.74	0.05	0.28	0.72
		Ours	0.44	0.10	0.90	0.14	0.01	0.99	0.14	0.04	0.96	0.22	0.21	0.79	0.10	0.09	0.91	0.07	0.07	0.93	0.12	0.06	0.95	0.17	0.08	0.92
GTSRB	PreActResNet18	STRIP	0.24	0.08	0.92	0.22	0.08	0.92	0.03	0.01	0.99	1.00	0.00	1.00	1.00	0.00	1.00	0.40	0.02	0.98	0.41	0.34	0.66	0.47	0.08	0.92
		FreqDetector	0.02	0.10	0.90	0.04	0.04	0.96	0.12	0.08	0.92	0.12	0.18	0.82	0.88	0.11	0.89	0.48	0.40	0.60	0.14	0.77	0.23	0.26	0.24	0.76
		Ours	0.18	0.15	0.85	0.12	0.05	0.95	0.07	0.01	0.99	0.05	0.04	0.96	0.01	0.07	0.93	0.03	0.00	1.00	0.06	0.01	0.99	0.07	0.05	0.95
	MobileViT-xs	STRIP	0.02	0.11	0.89	0.24	0.04	0.96	0.11	0.02	0.98	1.00	0.00	1.00	1.00	0.00	1.00	0.62	0.01	0.99	0.61	0.29	0.71	0.51	0.07	0.93
		FreqDetector	0.02	0.10	0.90	0.04	0.04	0.96	0.18	0.14	0.86	0.00	0.00	1.00	0.85	0.13	0.87	0.27	0.16	0.84	0.14	0.77	0.23	0.21	0.19	0.81
		Ours	0.15	0.04	0.96	0.13	0.00	1.00	0.01	0.02	0.98	0.18	0.06	0.94	0.03	0.05	0.95	0.05	0.05	0.95	0.07	0.01	0.99	0.09	0.03	0.97
CIFAR100	PreActResNet18	STRIP	0.25	0.12	0.88	0.36	0.20	0.80	0.11	0.10	0.90	1.00	0.00	1.00	1.00	0.00	1.00	0.76	0.06	0.94	0.41	0.29	0.71	0.56	0.11	0.89
		FreqDetector	0.02	0.13	0.87	0.09	0.11	0.89	0.08	0.35	0.65	0.02	0.03	0.97	0.00	1.00	0.00	0.07	0.14	0.86	0.12	0.26	0.74	0.06	0.29	0.71
		Ours	0.04	0.12	0.88	0.06	0.05	0.95	0.07	0.25	0.75	0.08	0.17	0.83	0.02	0.06	0.94	0.21	0.14	0.86	0.04	0.02	0.98	0.07	0.12	0.88
	MobileViT-xs	STRIP	0.29	0.11	0.89	0.31	0.20	0.80	0.09	0.14	0.86	0.88	0.09	0.91	1.00	0.00	1.00	0.60	0.13	0.87	0.24	0.26	0.74	0.49	0.13	0.87
		FreqDetector	0.02	0.13	0.87	0.09	0.11	0.89	0.09	0.23	0.77	0.01	0.01	0.99	0.00	1.00	0.00	0.08	0.16	0.84	0.12	0.26	0.74	0.06	0.27	0.73
		Ours	0.06	0.12	0.88	0.07	0.02	0.98	0.02	0.05	0.95	0.07	0.06	0.94	0.08	0.16	0.84	0.04	0.03	0.97	0.05	0.04	0.96	0.06	0.07	0.93
Tiny-ImageNet	PreActResNet18	STRIP	0.14	0.29	0.71	0.14	0.08	0.92	0.03	0.02	0.98	0.98	0.02	0.98	0.31	0.41	0.59	0.91	0.00	1.00	0.36	0.19	0.81	0.41	0.14	0.86
		FreqDetector	0.25	0.44	0.56	0.01	0.01	0.99	0.19	0.16	0.84	0.00	0.00	1.00	0.49	0.28	0.72	0.03	0.15	0.85	0.03	0.05	0.95	0.15	0.15	0.85
		Ours	0.03	0.00	1.00	0.04	0.01	0.99	0.01	0.01	0.99	0.04	0.01	0.99	0.04	0.18	0.82	0.05	0.00	1.00	0.02	0.03	0.97	0.03	0.03	0.97
	MobileViT-xs	STRIP	0.22	0.40	0.60	0.24	0.14	0.86	0.04	0.03	0.97	0.93	0.04	0.96	0.32	0.45	0.55	0.62	0.07	0.93	0.36	0.21	0.79	0.39	0.19	0.81
		FreqDetector	0.23	0.50	0.50	0.01	0.02	0.98	0.10	0.17	0.83	0.00	0.00	1.00	0.48	0.32	0.68	0.18	0.43	0.57	0.05	0.08	0.92	0.15	0.22	0.78
		Ours	0.04	0.00	1.00	0.05	0.00	1.00	0.03	0.02	0.98	0.03	0.00	1.00	0.04	0.01	0.99	0.11	0.14	0.86	0.03	0.02	0.98	0.05	0.03	0.97
ImageNet200	WideResNet101-2	STRIP	0.02	0.04	0.96	0.13	0.12	0.88	-	-	-	0.11	0.15	0.85	0.28	0.37	0.63	0.03	0.02	0.98	0.31	0.37	0.63	0.15	0.18	0.82
		FreqDetector	0.40	0.56	0.44	0.01	0.02	0.98	-	-	-	0.00	0.00	1.00	0.11	0.88	0.12	0.04	0.08	0.92	0.02	0.04	0.96	0.09	0.27	0.73
		Ours	0.04	0.00	1.00	0.04	0.00	1.00	-	-	-	0.02	0.00	1.00	0.04	0.02	0.98	0.00	0.00	1.00	0.03	0.02	0.98	0.03	0.01	0.99
	SwinT-Base	STRIP	0.08	0.19	0.81	0.10	0.10	0.90	-	-	-	0.98	0.01	0.99	0.34	0.53	0.47	0.64	0.03	0.97	0.38	0.23	0.77	0.42	0.18	0.82
		FreqDetector	0.40	0.56	0.44	0.01	0.02	0.98	-	-	-	0.00	0.00	1.00	0.19	0.78	0.22	0.04	0.07	0.93	0.02	0.04	0.96	0.11	0.25	0.75
		Ours	0.04	0.00	1.00	0.02	0.01	0.99	-	-	-	0.04	0.12	0.88	0.04	0.01	0.99	0.01	0.00	1.00	0.07	0.05	0.95	0.04	0.03	0.97

[*] LF is computationally infeasible on ImageNet200.

Table 4. The evaluation results on different attacks, datasets, and backbones. We observe that the results in additional metrics (FAR, FRR, and *Backdoored Data Rejection Rate* (BDR)) with optimal thresholds are aligned with the conclusions in the paper.

Avg, of	CIFAR10			GTSRB			CIFAR100			Tiny-ImageNet			ImageNet200			AVG		
ACC	$E=1$	$E=10$	$E=50$	$E=1$	$E=10$	$E=50$	$E=1$	$E=10$	$E=50$	$E=1$	$E=10$	$E=50$	$E=1$	$E=10$	$E=50$	$E=1$	$E=10$	$E=50$
CNNs	0.7766	0.7802	0.8078	0.8931	0.9011	0.8968	0.8850	0.8730	0.8823	0.9618	0.9618	0.9618	0.9773	0.9773	0.9773	0.8987	0.8987	0.9052
ViTs	0.7066	0.8000	0.7801	0.8349	0.8779	0.8687	0.9097	0.8998	0.8957	0.9492	0.9336	0.9377	0.9145	0.9639	0.9639	0.8630	0.8950	0.8892

Table 5. The accuracy of TeCo in the settings where defenders can estimate the thresholds based on n clean images

Avg, of	CIFAR10			GTSRB			CIFAR100			Tiny-ImageNet			ImageNet200			AVG		
ACC	STRIP	FreqDetector	Ours	STRIP	FreqDetector	Ours	STRIP	FreqDetector	Ours	STRIP	FreqDetector	Ours	STRIP	FreqDetector	Ours	STRIP	FreqDetector	Ours
CNNs	0.6188	0.8245	**0.8939**	0.7008	0.7395	**0.8899**	0.5868	**0.8053**	0.7434	0.6735	**0.8200**	0.8101	0.8135	0.8135	**0.9760**	0.6787	0.8006	**0.8627**
ViTs	0.5917	**0.8233**	0.7665	0.4988	**0.7687**	0.7668	0.6349	**0.8066**	0.7381	0.6896	0.7920	**0.8778**	0.6735	0.8153	**0.9639**	0.6177	0.8012	**0.8226**

Table 6. The accuracy of TeCo and two baselines in the settings where only one statistical threshold can be set to detect all attacks

Avg, of	CIFAR10			GTSRB			CIFAR100			Tiny-ImageNet			ImageNet200			AVG		
ACC	$\gamma=0$	$\gamma=0.5$	$\gamma=1$	$\gamma=0$	$\gamma=0.5$	$\gamma=1$	$\gamma=0$	$\gamma=0.5$	$\gamma=1$	$\gamma=0$	$\gamma=0.5$	$\gamma=1$	$\gamma=0$	$\gamma=0.5$	$\gamma=1$	$\gamma=0$	$\gamma=0.5$	$\gamma=1$
CNNs	0.6672	0.7824	0.8521	0.6604	0.7802	0.9242	0.8111	0.8367	0.7735	0.7351	0.7933	0.6504	0.6125	0.7309	0.7613	0.6973	0.7847	0.7923
ViTs	0.6130	0.7345	0.8018	0.6132	0.7236	0.8366	0.7816	0.8440	0.7778	0.7460	0.8590	0.7569	0.6313	0.7435	0.7610	0.6770	0.7809	0.7868

Table 7. The accuracy of TeCo in the settings where only one empirical threshold can be set to detect all attacks

Group	$\mathcal{G}_1$		$\mathcal{G}_2$		$\mathcal{G}_3$		$\mathcal{G}_4$		$\mathcal{G}_{1+2}$		$\mathcal{G}_{1+3}$		$\mathcal{G}_{1+4}$	
Metric	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score
Avg. of AVG(↑)	0.780	0.782	0.661	0.677	0.637	0.669	0.536	0.543	0.906	0.902	0.907	0.908	0.900	0.901
Avg. of STD(↓)	0.184	0.178	0.171	0.156	0.226	0.172	0.081	0.081	0.082	0.084	0.104	0.084	0.095	0.092

$\mathcal{G}_{2+3}$		$\mathcal{G}_{2+4}$		$\mathcal{G}_{3+4}$		$\overline{\mathcal{G}_1}$		$\overline{\mathcal{G}_2}$		$\overline{\mathcal{G}_3}$		$\overline{\mathcal{G}_4}$		ALL	
AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score
0.756	0.760	0.734	0.743	0.708	0.713	0.771	0.775	0.935	0.931	0.923	0.920	0.938	0.929	**0.945**	**0.940**
0.183	0.170	0.187	0.171	0.199	0.183	0.189	0.175	0.050	0.052	0.060	0.064	0.042	0.041	**0.035**	**0.034**

Table 8. The performance of TeCo based on different image corruption sets. Results are averaged from different attacks, datasets, and backbones.

TeCo based on different deviation measurement methods.

2.4. Discussion of Outliers

There are some interesting results about baselines. Since the Low-frequency (LF) attack is designed to avoid FreqDetector [5], FreqDetector should have low effectiveness against this attack. However, we implement them following the official codes and find that if we let FreqDetector work in a binary classification manner and make judgments based on thresholds, it will perform well on LF at-

Measure	Standard Deviation		Range		Mean Deviation		Coefficient of Variation		Quartile Deviation	
Metric	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score
Avg. of AVG(↑)	0.944	**0.939**	0.912	0.906	**0.945**	**0.939**	0.895	0.906	0.708	0.710
Avg. of STD(↓)	**0.035**	**0.034**	0.068	0.069	**0.035**	**0.034**	0.075	0.062	0.186	0.180

Table 9. The performance of TeCo based on different measures of variation. Results are averaged from different attacks, datasets, and backbones.

Group	Type	Corruptions
$\mathcal{G}_1$	Noise	Gaussian Noise, Shot Noise, Impulse Noise
$\mathcal{G}_2$	Blur	Defocus Blur, Glass Blur, Motion Blur, Zoom Blur
$\mathcal{G}_3$	Nature	Snow, Frost, Fog, Brightness
$\mathcal{G}_4$	Digital	Contrast, Elastic Transform, Pixelate, Jpeg Compression

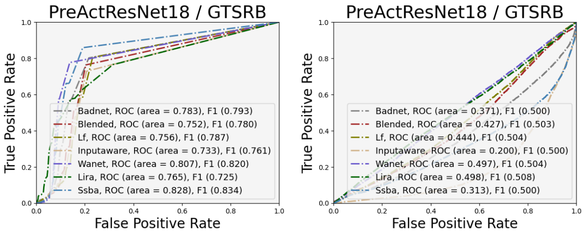
Table 10. Images corruptions in different groups.

Runs	Input-aware				LIRA			
	SwinT-B		WideResNet		SwinT-B		WideResNet	
	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score
Run#1	0.936	0.86	0.423	0.504	0.994	0.975	0.696	0.645
Run#1	0.939	0.864	0.383	0.502	0.994	0.976	0.684	0.667

Table 11. The additional random runs of STRIP on ImageNet200

Dataset	Accuracy	Badnets	Blended	LF	Input-aware	Wanet	LIRA	SSBA
CIFAR10	Trigger Samples	93.38	79.40	54.41	99.89	3.74	64.84	59.29
	Clean Images	96.91	96.91	96.91	96.91	96.91	96.91	96.91
GTSRB	Trigger Samples	91.62	97.05	82.01	73.18	3.98	10.50	12.97
	Clean Images	94.28	94.28	94.28	94.28	94.28	94.28	94.28
CIFAR100	Trigger Samples	87.98	77.54	58.47	98.58	1.87	81.42	54.02
	Clean Images	96.17	96.17	96.17	96.17	96.17	96.17	96.17
Tiny-ImageNet	Trigger Samples	17.94	99.32	54.80	99.94	4.75	82.64	92.23
	Clean Images	98.41	98.41	98.41	98.41	98.41	98.41	98.41
ImageNet200	Trigger Samples	1.39	98.03	-	100.00	0.72	86.90	94.67
	Clean Images	99.07	0.00	-	37.20	99.07	37.20	99.07

Table 12. The effectiveness of FreqDetector in its original mode



(a) Ours

(b) STRIP

Figure 1. ROC of detecting all-to-all attacks

tack. So we believe it is not unfair to involve LF attack and FreqDetector simultaneously in our experiments. To prove the implementation correctness of FreqDetector, we share the performance of FreqDetector on its original mode in Tab. 12, which indicates that LF attack does avoid its detection if FreqDetector works on its original mode. In addition, Wanet can avoid detection, which is aligned with the results in our paper.

Another interesting phenomenon is the success of STRIP against Input-aware and LIRA attacks on SwinTransformer-base/ImageNet200, while STRIP fails on other datasets and backbones. We re-run these experiments by setting different random seeds to ensure the stability of the results. As shown in Tab. 11, the results from different random seeds are similar, indicating that the performance of STRIP is somehow influenced by the choice of datasets and backbones.

2.5. Additional Experiments

Effectiveness against all-to-all attacks. In practice, a backdoor-infected model may have multiple labels embedded with Trojans, *i.e.*, the multiple classes scenario. Here, we consider the worst multiple classes scenario (all-to-all attack) where every class in the victim model is attacked and the backdoor trigger can cause a specific transition of trigger samples' labels (e.g., turning y_i to $y_i + 3$). We investigate TeCo and two baselines against the all-to-all attack on PreActResNet18/GTSRB[8]. As depicted in Fig. 1(d)-(e), the performance of our method drops about 20% in this scenario but still maintains stability across different attacks. STRIP has lost its performance totally and even makes contrary predictions. Since FreqDetector makes judgments only based on the images, it maintains its performance as the same as which Tab. 4 shows. Fortunately, TeCo is still comparable with FreqDetector in this worst-case setting as demonstrated in Tab. 13.

Method	Avg. of AUROC	Avg. of F1 score	Std. of AUROC	Std. of F1 score
STRIP	0.3930	0.5026	0.0997	0.0027
FreqDetector	0.7911	0.7671	0.2235	0.2027
Ours	0.7749	0.7856	0.0306	0.0336

Table 13. Quantization results of detecting all-to-all attacks

Smaller triggers. The results show that TeCo is effective and even more strong against the backdoor attack with smaller triggers.

Size	% of image	AUROC(↑)	F1 score(↑)	ACC(↑)	FAR(↓)	FRR(↓)	BDR(↑)
7*7	0.10	0.9963	0.9969	99.69	0.60	0.01	99.99
14*14	0.39	0.9973	0.9974	99.74	0.49	0.02	99.98
21*21	0.88	0.9784	0.9782	97.82	4.12	0.23	99.77

Table 14. ImageNet200 / SwinT-Base

Transferability to unseen attacks. The results show in Tab. 15 that TeCo is effective against unseen attacks after optimizing a threshold using a known attack.

Corruptions as data augmentations. The results show

[8] We find all-to-all backdoor attacks are not stable enough on other datasets and cause difficulties to do evaluations.

		Badnets	Blended	LF	Input-Aware	Wanet	LIRA	SSBA	AVG
Badnets	ACC	-	0.9451	0.9339	0.8647	0.8559	0.9322	0.8475	0.8966
	FAR	-	0.1017	0.1155	0.2096	0.1987	0.1183	0.1641	0.1513
	FRR	-	0.0029	0.0112	0.0528	0.0833	0.0117	0.1396	0.0502
	BDR	-	0.9971	0.9888	0.9472	0.9167	0.9883	0.8604	0.9498
Blended	ACC	0.9169	-	0.9366	0.8703	0.8629	0.9215	0.8324	0.8901
	FAR	0.1119	-	0.1099	0.1987	0.1849	0.1138	0.1590	0.1464
	FRR	0.0511	-	0.0117	0.0531	0.0840	0.0392	0.1771	0.0694
	BDR	0.9489	-	0.9883	0.9469	0.9160	0.9608	0.8229	0.9306

Table 15. CIFAR10 / PreActResNet18

that the degradation of performance is not large when the backdoor attacks use corruption for data augmentation.

Augmentation	Metric	Badnets	Blended	SSBA	AVG
Aug, 50%	AUROC(↑)	0.9040	0.9038	0.8968	0.9015
	F1 score(↑)	0.8890	0.8863	0.8922	0.8892
	BDR(↑)	93.77	95.51	95.31	94.86

Table 16. GTSRB / MobileViT-xs

More recent attacks. We show additional results on detecting Sleeper Agent [4].

Attack	AUROC(↑)	F1 score(↑)	ACC(↑)	FAR(↓)	FRR(↓)	BDR(↑)
Sleeper	0.8897	0.9325	93.25	10.80	2.70	97.30

Table 17. CIFAR10 / PreActResNet18

Insightful discussion of TeCo. As we discussed in Sec.6, the explanation of TeCo is very likely to be the dual-target training function of backdoor attacks which leads to the huge bias of victim models. The bias makes victim models focus on the trigger patterns rather than the original information of trigger samples. When the trigger patterns encounter different corruptions, since some corruptions are in texture information while others are in structure information, the trigger will be robust against certain corruptions while not robust against others. And since clean images have more complex texture and structure information compared with trigger patterns which need to be simple and repetitive for causing bias, the clean images will have consistent robustness. In this paper, our main goal is to discover and introduce this phenomenon to the community with comprehensive empirical studies. A formal theoretical study will be our future work.

Use TeCo to detect backdoor-infected models. As we have mentioned in our introduction, TeCo is a *test-time trigger sample detection* (TTSD) method that can seamlessly integrate into existing model diagnosis defenses for defense. In practice, defenders can first use model diagnosis defenses (e.g., AEVA [2], which also works in hard-label black-box settings) to judge whether the target model is a backdoor model. Then the defenders can use TeCo to detect the trigger samples. On the other hand, TeCo can be used to diagnose the model. Our study shows that for the clean samples on clean models, the FAR of TeCo will be high when applying thresholds calculated from the backdoor model (Avg. FAR$\approx 55\%$ on GTSRB/PreActResNet18 clean model). So defenders may feed a batch of clean images into the target model, and calculate the FAR of TeCo to judge whether the target model is a backdoor model.

References

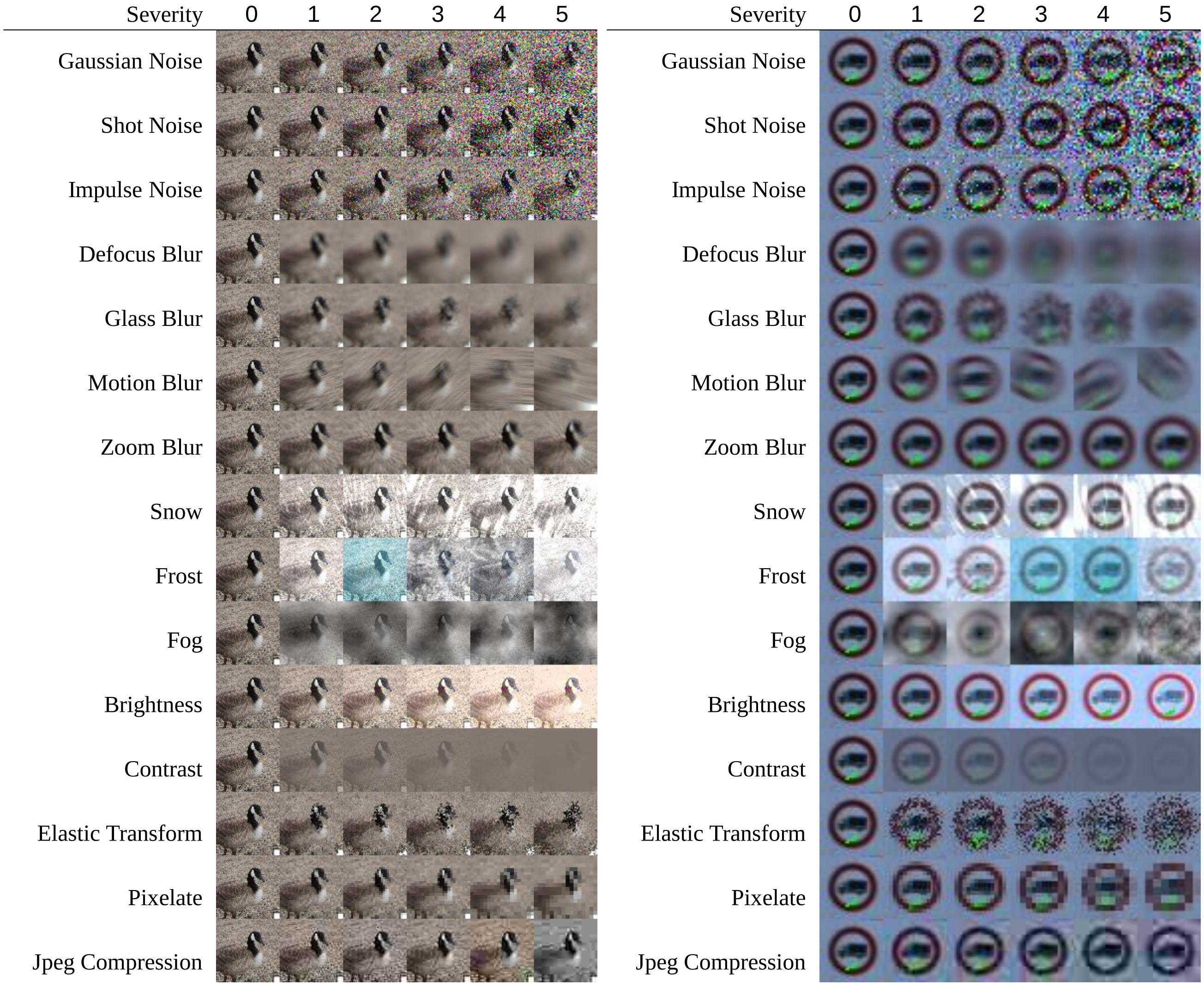
[1] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC'19)*, pages 113–125, 2019. 1

[2] Junfeng Guo, Ang Li, and Cong Liu. AEVA: black-box backdoor detection using adversarial extreme value analysis. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*, 2022. 5

[3] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "Beatrix" resurrections: Robust backdoor detection via gram matrices. *arXiv preprint arXiv:2209.11715*, 2022. 2

[4] Hossein Souri, Micah Goldblum, Liam Fowl, Rama Chellappa, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *arXiv preprint arXiv:2106.08970*, 2021. 5

[5] Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*, pages 16473–16481, 2021. 1, 3



(a) Badnets / Tiny-ImageNet.

(b) Input-aware / GTSRB.

Figure 2. Visualization of trigger samples and their corrupted versions