The Victim and The Beneficiary: Exploiting a Poisoned Model to Train a Clean Model on Poisoned Data
—(*Supplementary Materials*)—

A. Appendix

A.1. Details about Datasets

The detailed information of the datasets used in our experiments is summarized in Table 1.

Dataset	Classes	Input Size	Training Images	Test Images
CIFAR-10	10	32 x 32 x 3	50000	10000
ImageNet Subset	12	224 x 224 x 3	12480	2860

Table 1. Detailed information of the datasets used in our experiments.

A.2. Detailed Settings for Backdoor Attacks

We trained all attack baselines for 200 epochs using the SGD optimizer with an initial learning rate of 0.1, a weight decay of 1e-4, and a momentum of 0.9. The learning rate was divided by 10 after every 50 epochs. We set the batch size to 128 for CIFAR-10 and 16 for the ImageNet subset. The poisoned samples crafted by different attacks are shown in Figure 1.

Settings for BadNets [3] We use a 2×2 square as the trigger on CIFAR-10 and a 32×32 Apple logo on the ImageNet subset, as suggested in previous studies [4, 8]. The triggers are added in the upper left corner of benign samples.

Settings for Blend [2] We use a 'Hello Kitty' image as the trigger on CIFAR-10 and a random noisy pattern on the ImageNet subset. The blended ratio is set to 0.1 for both datasets.

Settings for WaNet [5] Following the original settings, we set the grid size $k = 4$ and the warping strength $s = 0.5$ on CIFAR-10. But for the ImageNet subset, the grid size is set to 224 and the warping strength is set to 1 to ensure the attack works, as suggested in [4]. We set the noisy rate $\rho_n = 0.2$ for both datasets.

Settings for Dynamic [6] We use the pre-trained generator to generate triggers for each poisoned sample and cross-triggers for a small portion of benign samples. The cross-trigger mode rate ρ_c is set to 0.1, the same as the poisoning rate.

Settings for CL [7] Based on the perturbed images released by the author, we paste the BadNets trigger to them to create poisoned samples.

Settings for SIG [1] Following the original settings, we set $\triangle = 20$ and $f = 6$ to generate a sinusoidal signal as the

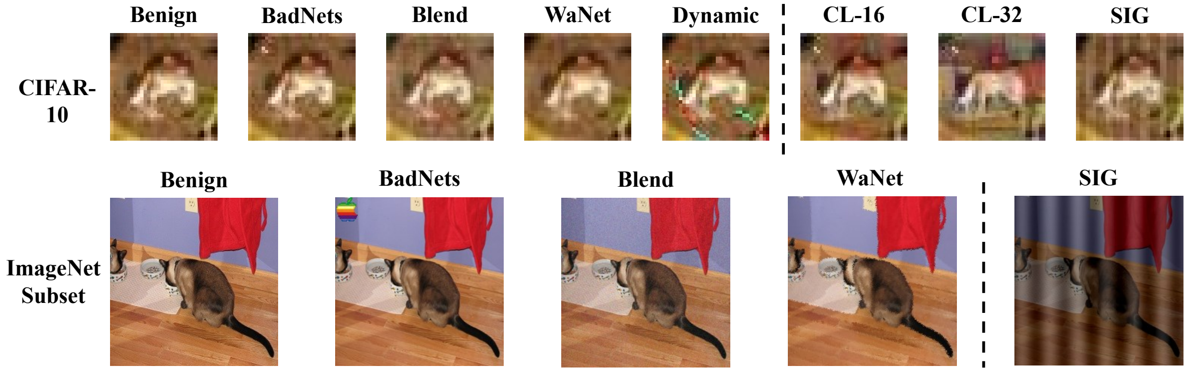

Figure 1. Poisoned samples crafted by different backdoor attacks for CIFAR-10 and ImageNet subset, including BadNets [3], Blend [2], WaNet [5], Dynamic [6], CL [7] and SIG [1]. The first sample in the two rows is benign.

trigger on CIFAR-10, and superimpose it on benign samples with a ratio of 0.1. For ImageNet, we set $\triangle = 60$, $f = 6$, a blend ratio of 0.5, and a poisoning rate of 0.5 to make the attack effective.

A.3. Warming-up Strategy Selection

In this section, we introduce the way that we select the filtering threshold t_f in the warming-up stage.

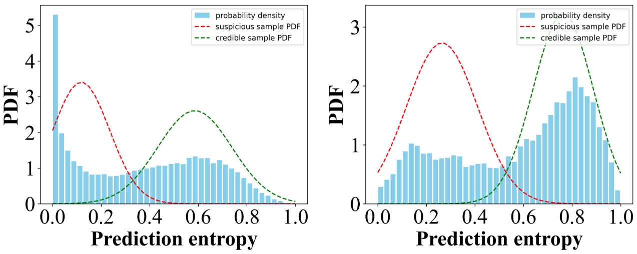

Figure 2. The distributions of prediction entropy on CIFAR-10 (left) and the ImageNet subset (right) under BadNets attack. The blue histogram is the probability density histogram of all samples' prediction entropy. The red and green dotted lines are two Gaussian distributions fitted by GMM, donating the distribution of suspicious samples and the distribution of credible samples, respectively.

During the standard training process, the prediction entropy of poisoned samples will drop faster than benign ones, leading to two clusters in the distribution of prediction entropy. Therefore, we adopt a two-component Gaussian Mixture Model (GMM) to fit the distribution of all samples' prediction entropy using the Expectation-Maximization algorithm, as shown in Figure 2. We assume the prediction entropy of suspicious samples follows $N(\mu_1, \sigma_1^2)$ and that of benign samples follows $N(\mu_2, \sigma_2^2)$. We use the GMM to distinguish suspicious samples and train a network with them until the two distributions roughly separate, *i.e.* $\mu_1 + 3\sigma_1 < \mu_2$. In order to retain more samples to train the Beneficiary network, we select $\mu_1 + \sigma_1$ as the filtering threshold. We conduct experiments on CIFAR-10 and ImageNet subset with BadNets and Blend attacks, finding that $\mu_1 + \sigma_1$ stops at about 0.2 after 2 to 3 epochs on CIFAR-10 and at about 0.4 after 5 to 6 epochs on the ImageNet subset. So we warm up the Victim network for 3 epochs on CIFAR-10 with t_f linearly decreases from 1 to 0.2 and for 6 epochs on the ImageNet subset with t_f linearly decreases from 1 to 0.4. We adopt the same settings against other attacks on both datasets and find that they still perform well. The reason why GMM has not been used for identification in stage 2 and stage 3 is that as the training progresses, the two distributions will gradually approach, thus affecting the ability of GMM to distinguish.

Inspired by that using GMM can well distinguish suspicious samples, we design an automatic warming-up strategy based on GMM to facilitate the selection of the filtering threshold. Similarly, we also adopt GMM to identify suspicious samples and train the Victim network with them until $\mu_1 + 3\sigma_1 < \mu_2$. Then we fix the t_f to $\mu_1 + 0.5\sigma_1$ in the following training stages because we find this threshold is large enough to filter out most poisoned samples. As Tabel 2 shows, the automatic warming-up strategy achieves comparable results to the original settings, which provides an alternative possibility for fast selection of the filtering threshold.

Dataset	Attack	Metric	Ours(original)	Ours(automatic)
CIFAR-10	BadNets [3]	BA	93.96%	**94.33%**
		ASR	**0.62%**	0.74%
	Blend [2]	BA	**94.37%**	93.87%
		ASR	**0.63%**	0.72%
	WaNet [5]	BA	**94.15%**	91.74%
		ASR	**0.54%**	1.00%
	CL-16 [7]	BA	**94.24%**	93.25%
		ASR	1.01%	**0.76%**
	CL-32 [7]	BA	**93.98%**	93.14%
		ASR	**0.64%**	1.17%
	SIG [1]	BA	**94.08%**	93.15%
		ASR	0.17%	**0.01%**
	Dynamic [6]	BA	**93.91%**	93.21%
		ASR	**1.13%**	2.51%
ImageNet Subset	BadNets [3]	BA	95.42%	**95.90%**
		ASR	0.28%	**0.24%**
	Blend [2]	BA	**95.03%**	92.08%
		ASR	**0.45%**	1.54%
	WaNet [5]	BA	**94.84%**	94.65%
		ASR	1.92%	**0.49%**
	SIG [1]	BA	94.65%	**95.74%**
		ASR	**0.03%**	0.17%

Table 2. The results of V&B with original warming-up strategy and automatic warming-up strategy. The automatic warming-up strategy achieves comparable results to the original settings, which provides an alternative possibility for fast selection of the filtering threshold.

A.4. Effectiveness under Different Poisoning Rate

We verified the effectiveness of our framework with poisoning rates ranging from 0.1 to 0.5, and the results are shown in Figure 3. Against most attacks, our framework can reduce the attack success rate to below 1% at various poisoning rates, while maintaining a satisfactory benign accuracy. With the poisoning rate increasing, the Victim network can better learn trigger patterns and filter out a larger proportion of poisoned samples for the Beneficiary network. This is why our framework works fine even with a poisoning rate of 0.5. Although the proportion of filtered poisoned samples becomes larger, the number of missed poisoned samples may also increase as the poisoning rate increases. Especially when the poisoning rate is not high enough (*e.g.* 0.2 for SIG and 0.4 for WaNet), the Victim network will miss more poisoned samples, which may cause the attack success rate to increase. When the poisoning rate reaches 0.5, all benign accuracy drops because fewer benign samples can be used. We can also observe that the benign accuracy of some attacks has fluctuated (*i.e.* 0.2 for BadNets, 0.3 for Blend, and 0.4 for WaNet), which is possibly due to selecting too many benign samples of a certain class for poisoning, resulting in the model being unable to correctly relabel the poisoned samples under this class.

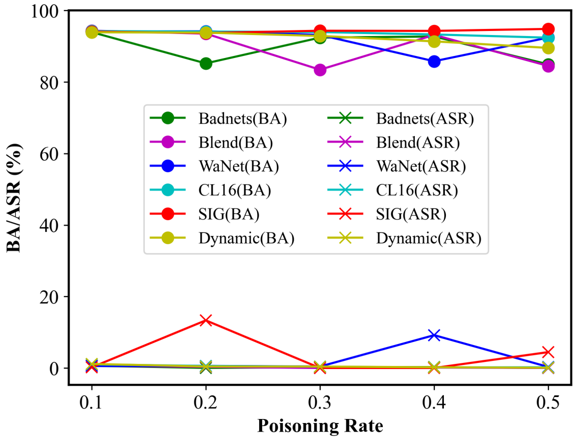

Figure 3. The benign accuracy and attack success rate under different poisoning rates.

A.5. Detailed Results of Our V&B

Dataset	Attack	Metric	After stage 2	After stage 3
CIFAR-10	BadNets [3]	BA	89.87%	**93.96%**
		ASR	1.94%	**0.62%**
	Blend [2]	BA	90.03%	**94.37%**
		ASR	4.94%	**0.63%**
	WaNet [5]	BA	91.46%	**94.15%**
		ASR	75.33%	**0.54%**
	CL-16 [7]	BA	91.69%	**94.24%**
		ASR	1.71%	**1.01%**
	CL-32 [7]	BA	91.09%	**93.98%**
		ASR	1.91%	**0.64%**
	SIG [1]	BA	91.27%	**94.08%**
		ASR	3.40%	**0.17%**
	Dynamic [6]	BA	90.03%	**93.91%**
		ASR	76.34%	**1.13%**
ImageNet Subset	BadNets [3]	BA	94.42%	**95.42%**
		ASR	0.52%	**0.28%**
	Blend [2]	BA	91.31%	**95.03%**
		ASR	2.13%	**0.45%**
	WaNet [5]	BA	90.87%	**94.84%**
		ASR	4.55%	**1.92%**
	SIG [1]	BA	91.76%	**94.65%**
		ASR	2.38%	**0.03%**

Table 3. Detailed results of our V&B against different attacks.

In Table 3, we show the results after stage 2 and stage 3 (final results) of our framework for all attack cases. It is obvious that the attack success rates after stage 2 are all higher than the final results, while the benign accuracy is the opposite. This demonstrates that our semi-supervised suppression training can both erase existing backdoors and improve model performance on benign samples.

References

[1] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A New Backdoor Attack in CNNS by Training Set Corruption Without Label Poisoning. In *2019 IEEE International Conference on Image Processing*, pages 101–105. IEEE, 2019.

[2] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[3] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv preprint arXiv:1708.06733*, 2017.

[4] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor Defense via Decoupling the Training Process. In *The Tenth International Conference on Learning Representations*, 2022.

[5] Anh Nguyen and Anh Tran. WaNet–Imperceptible Warping-based Backdoor Attack. *arXiv preprint arXiv:2102.10369*, 2021.

[6] Tuan Anh Nguyen and Anh Tran. Input-Aware Dynamic Backdoor Attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[7] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-Consistent Backdoor Attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[8] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *2019 IEEE Symposium on Security and Privacy*, pages 707–723. IEEE, 2019.