

A. Mechanisms of Different Poisons

As discussed in Section 2 in the main paper, various backdoor attacks have been developed in the literature. In this paper, we examine several of the most representative attacks: BadNets [23], blended injection attack [9], Trojan [42], sinusoidal signal backdoor [2], label-consistent backdoor [58], and hidden trigger backdoor [52]. We elucidate the mechanisms of these attacks in this section.

A.1. Dirty-Label Attacks

BadNets. BadNets is the first backdoor attack against deep neural networks, which uses a pattern (e.g., flower, colored square, boom, etc.) as the backdoor trigger. Poisoned samples are patched with the backdoor trigger and assigned an incorrect target label. The poisons generation process are expressed in Equation 7, where \mathcal{T} is the backdoor trigger and \mathcal{M} is the masking area to put the trigger.

$$\begin{aligned} x_p &= x_b + \mathcal{T} \odot \mathcal{M}, \\ y_p &= y_b \leftarrow y_t, \end{aligned} \quad (7)$$

Blended Injection Attack. Blended crafts backdoor trigger using a fixed cartoon pattern (e.g., Hello Kitty) or a random pattern (e.g., random noise). Similar to BadNets, a target dirty-label is assigned to all poisoned samples. Equation 8 depicts the generation process of Blended, where α is the blended ratio.

$$\begin{aligned} x_p &= \prod_{\alpha}^{\text{blend}} (\mathcal{T}, x_b) = \alpha \odot \mathcal{T} + (1 - \alpha) \odot x_b, \\ y_p &= y_b \leftarrow y_t, \end{aligned} \quad (8)$$

Trojan. Trojan generates the backdoor trigger by assigning value to the input variables in a given trigger mask. The value assignment process aims to make some selected internal neurons (i.e., a neuron in the last layer denoting the target dirty-label) to achieve the maximum values. The neurons selection process can be mathematically expressed by Equation 9 where L_{target} is the layer of target neurons, $L_{\text{preceding}}$ is the preceding layers, W is the weights and b is the biases between the layers.

$$\begin{aligned} L_{\text{target}} &= L_{\text{preceding}} * W + b, \\ \arg \max_{y_t} & \left(\sum_{i=0}^n \text{ABS}(W_{L(i, y_t)}) \right), \end{aligned} \quad (9)$$

Once the dirty-label and corresponding target neurons are selected, the poison samples are generated as shown in Equation 10, where lr is the learning rate and cost is the sum of the square error of target values and actual neuron values.

$$x_p = x_b + (\mathcal{T} - lr \cdot \frac{\partial \text{cost}}{\partial \mathcal{T}}). \quad (10)$$

A.2. Clean-Label Attacks

Sinusoidal Signal Backdoor (SIG). SIG devises the sinusoidal signal as the backdoor trigger, expressed as:

$$\mathcal{T}(i, j) = \Delta \sin(2\pi j f / m), 1 \leq j \leq m, 1 \leq i \leq l, \quad (11)$$

where f is the frequency of the sinusoidal signal, Δ is the strength, m and l are the numbers of columns and rows of an image, respectively. Then the backdoor trigger \mathcal{T} is superimposed with benign images x_b to form poisoned samples x_p , i.e., $x_p = x_b + \mathcal{T}$.

Label Consistent Backdoor (LCBD). LCBD places the trigger on images that are hard to classify so that the model relies more on the backdoor trigger to learn. The work proposes two approaches to make images hard-to-classify. The first generative adversarial network (GAN) based method crafts images \tilde{x}_p by interpolating a given image x_1 to another x_2 through the latent space, where x_1 is the image from the poisoned class while x_2 is an arbitrary image from a different class. The process can be expressed as:

$$\begin{aligned} \tilde{z}_1 &= \arg \min_{z_1 \in \mathbb{R}^d} \|x_1 - \mathcal{G}(z_1)\|_2, \\ \tilde{z}_2 &= \arg \min_{z_2 \in \mathbb{R}^d} \|x_2 - \mathcal{G}(z_2)\|_2, \end{aligned} \quad (12)$$

$$\tilde{x}_p = \mathcal{G}(\tau \tilde{z}_1 + (1 - \tau) \tilde{z}_2), \quad (13)$$

where \tilde{z}_1 and \tilde{z}_2 are the optimized latent space variables that produce close inputs to x_1 and x_2 , respectively. \mathcal{G} is the generator of a well-trained GAN model and τ is the parameter to direct the transition from x_1 to x_2 .

The second adversarial perturbation based method leverages the principle of adversarial examples (AE) [22] and constructs hard-to-classify images \tilde{x}_p as:

$$\tilde{x}_p = \arg \max \mathcal{L}(x_b + \delta, y, \theta) \quad \text{s.t.} \quad \|\delta\|_p \leq \epsilon, \quad (14)$$

where x_b represents benign images, y represents the ground-truth labels, θ is the model parameters and δ is the adversarial perturbation constraint by the upper bound ϵ in ℓ_p -norm. Lastly, poisoned samples are generated by placing a pre-defined trigger \mathcal{T} on \tilde{x}_p :

$$x_p = \mathcal{T} \odot \mathcal{M} + \tilde{x}_p \odot (1 - \mathcal{M}), \quad (15)$$

Hidden Trigger Backdoor (HTBD). HTBD is an even stealthier attack based on feature collision. The attack has two steps to generate poisoned data. As shown in Equations (16) and (17), it generates a base image (from the source class) attached with a pre-defined trigger and then optimize poisoned data x_p (from the target class) by minimizing their ℓ_2 -norm distance with \tilde{x}_b in the feature space.

$$\tilde{x}_b = x_b \odot (1 - \mathcal{M}) + \mathcal{T} \odot \mathcal{M}, \quad (16)$$

$$x_p = \arg \max_{\|x - x_t\|_{\infty} < \epsilon} \|f(x) - f(\tilde{x}_b)\|_2^2. \quad (17)$$

B. Details of Experimental Settings

We introduce the details of experimental settings in this section. Following the open-sourced repositories of original papers [2, 9, 23, 42, 52, 58], all experiments are implemented in Tensorflow [1] (LCBD¹) and PyTorch [47] (BadNets², Blended, Trojan³, SIG⁴ and HTBD⁵), and run on NVIDIA Tesla V100 GPUs. To reproduce the attacks and achieve the best attack performance, we follow original settings introduced in original papers. The statistics of datasets and models architecture for BadNets, Blended, Trojan, SIG, LCBD, and HTBD are summarized in Table 9. Note that all the datasets used in this paper, i.e., GTSRB⁶ [55], CIFAR-10⁷, and ImageNet-ILSVRC2012⁸ [12], are public.

The detailed training settings of each attack are described as follows.

BadNets, Blended and Trojan. For the dirty-label attacks experiments, we use Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9 and a weight decay of 5×10^{-4} to train DNN models from scratch. The initial learning rate, batch size, and total training epochs are set to 0.1, 128, and 200, respectively.

SIG. We train DNN models from scratch using SGD optimizer with momentum of 0.9 in both pre-clean and post-clean phases. The initial learning rate is set to 0.01, with a weight decay of 5×10^{-4} . Models are trained for 100 epochs using a batch size of 32.

LCBD. All models are trained from scratch using SGD with momentum of 0.9, a weight decay of 2×10^{-4} and batch size of 50. We train models for 80000 steps in total with a learning rate that starts at 0.1 and schedule a learning rate decay that reduces the learning rate to 0.01 at 40000 steps and 0.001 at 60000 steps.

HTBD. Different from SIG and LCBD, the HTBD trains models in a fine-tuning fashion according to the original paper [52]. During the generation of the poisoned samples, we use the fc7 features of AlexNet (i.e., extract features from the 7th fully-connected layer of AlexNet) for feature collision. Poisoned data are generated with a learning rate of 0.01 in total 2 optimization epochs, and each epoch has 5000 iterations. During fine-tuning, models are trained using SGD with momentum of 0.9 and a learning rate of 0.001. The batch size and epoch are set to 256 and 30 for detection on the poisoned class and 1024 and 10 for the detection on the whole training dataset, respectively.

Attack	Dataset	#Training Images	DNN Model
SIG	GTSRB	4,772	ResNet
BadNet	CIFAR-10	50,000	VGG16
Trojan	CIFAR-10	50,000	VGG16
Blended (HK)	CIFAR-10	50,000	VGG16
Blended (RP)	CIFAR-10	50,000	VGG16
LCBD	CIFAR-10	50,000	ResNet
HTBD	ImageNet	>1M	AlexNet

Table 9. Dataset and model architecture statistics

C. Ablation Study

Our proposed methodology employs the natural property of DNNs and two denoising functions to detect the suspicious poisoned samples. Here, we examine the performance when applying only one of these two denoising functions to generate baseline images. Our experiment results show that different backdoor attacks have inconsistent sensitivity to various baseline images. In practical scenarios where the defender does not know the type of attack in advance, the safest strategy is to take baseline images generated by both denoising functions into consideration. Moreover, the “median + mean” version achieves better ASR reduction in general, although each single baseline version may have a higher detection rate in some cases.

We also evaluate the sensitivity of the removal threshold β . We find that only SIG is sensitive to β while other attacks are not. In most cases, UltraClean can detect more than 80% poisoned samples and reduce ASR significantly where β is only at a value of 0.1. When β approaches the value of 0.3, UltraClean detects nearly 100% poisoned samples and almost completely removes the backdoor while maintaining a decent model accuracy. The experiment results provide insights into selecting β in practice (i.e., a value of 0.3 would be good enough).

C.1. Dirty-Label Attacks

We present the ablation experiments results of dirty-label attacks in Table 10. We find that BadNets is more sensitive to the type of baseline images. UltraClean fails to detect poisoned samples only based on baseline images generated by the mean denoising function, while successfully detecting most poisoned samples using baseline images generated by the median denoising function. On the other hand, UltraClean demonstrates consistently superior performance in detecting Blended and Trojan poisoned samples regardless of the type of baseline images. In general, the detection performance reaches the peak when we employ both baseline images. Meanwhile, all three attacks are not very sensitive to the change of the removal threshold. Although higher removal thresholds indeed improve the detection rate, a lower

¹<https://github.com/MadryLab/label-consistent-backdoor-code>

²<https://github.com/Koosci/BadNets>

³<https://github.com/PurduePAML/TrojanNN>

⁴https://github.com/ebagdasa/backdoor_federated_learning

⁵<https://github.com/UMBCvision/Hidden-Trigger-Backdoor-Attacks>

⁶https://benchmark.ini.rub.de/gtsrb_news.html

⁷<https://www.cs.toronto.edu/~kriz/cifar.html>

⁸<https://www.image-net.org/challenges/LSVRC/2012/index.php>

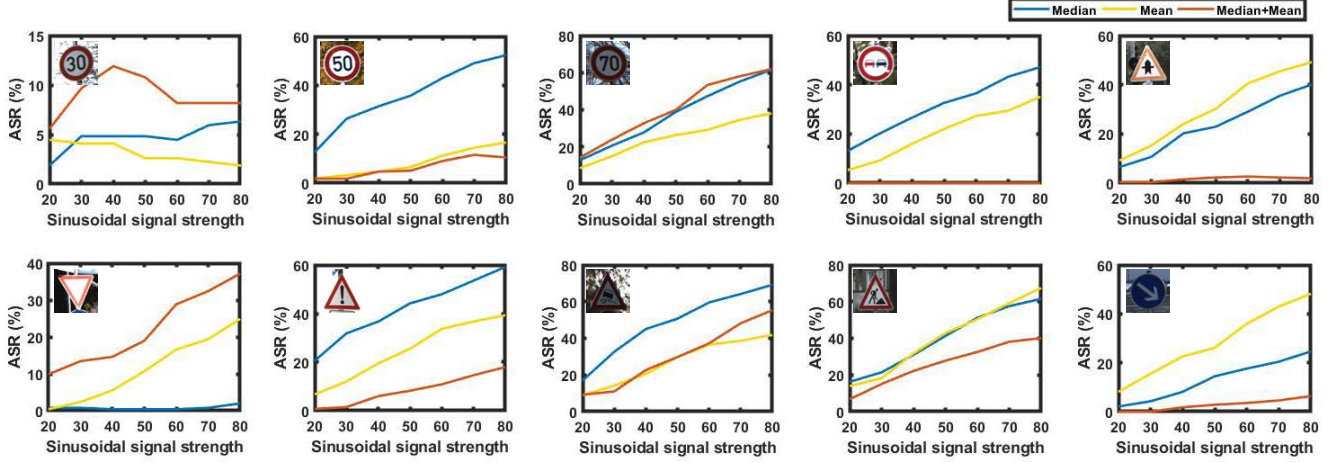


Figure 4. ASR comparison of using single “median” baseline images, single “mean” baseline images and aggregation of median and mean (SIG). The aggregation of median and mean (red curves) achieves lower post-clean ASR in most cases, demonstrating a better backdoor mitigation capability over single “median” (blue curves) and single “mean” (yellow curves).

Attack Type	BDR (Median)	BDR (Mean)	BDR (Both)
BadNets ($\beta = 0.1$)	88.86%	0.04%	83.50%
BadNets ($\beta = 0.2$)	94.52%	0.30%	92.74%
BadNets ($\beta = 0.3$)	96.84%	1.48%	94.42%
Trojan ($\beta = 0.1$)	96.62%	79.84%	97.26%
Trojan ($\beta = 0.2$)	99.10%	87.74%	99.32%
Trojan ($\beta = 0.3$)	99.28%	90.58%	99.5%
Blended (HK, $\beta = 0.1$)	80.76%	82.20%	88.60%
Blended (HK, $\beta = 0.2$)	93.18%	94.90%	95.84%
Blended (HK, $\beta = 0.3$)	95.96%	96.60%	97.38%
Blended (RP, $\beta = 0.1$)	95.30%	82.68%	96.10%
Blended (RP, $\beta = 0.2$)	99.18%	92.82%	99.46%
Blended (RP, $\beta = 0.3$)	99.74%	95.52%	99.80%

Table 10. Ablation study of denoising functions and β on dirty-label attacks

removal threshold can already effectively detect 83%~97% poisoned samples.

C.2. Clean-Label Attacks

SIG. The ablation study of baseline images generated by different denoising functions and β for detection on the poisoned class and the whole dataset are summarized in Tables 11 and 12, respectively. For the detection on the poisoned class, using the single “mean” baseline images achieves a slightly higher detection rate than using the single “median” function and the aggregation of mean and median for most classes. Results of the detection on the whole dataset reveal the same trend. We then compare the post-clean ASR and present the results in Figure 4. Although using the single “mean” baselines perform better in detection rate, there is

Class ID	BDR (Median)	BDR (Mean)	BDR (Both)
1	52.17%	63.48%	52.17%
2	56.41%	65.38%	56.41%
3	47.54%	65.57%	50.82%
4	67.22%	80.00%	69.01%
5	47.29%	57.43%	46.62%
6	48.71%	70.77%	52.82%
7	49.61%	70.08%	62.99%
8	51.92%	51.92%	51.92%
9	46.23%	62.81%	48.74%
10	85.42%	77.08%	91.67%

Table 11. Ablation study of denoising functions for detection on the poisoned class (SIG)

still a considerable performance gap to applying both baseline images in most cases. In general, the aggregation of median and mean denoising shows the best performance in degrading the post-clean ASR. We believe this is because the aggregation leads to a larger noise difference, which makes poisoned samples easier to detect. The ablation study of β shows that UltraClean’s performance against SIG is sensitive to the adjustment of the removal threshold. A larger β can detect more poisoned samples and better reduce ASR.

LCBD. Tables 13 and 14 present the detection performance against LCBD on the poisoned class and the whole dataset, respectively. It can be seen that the single “median” denoised baseline images outperform the single “mean” denoised baseline images by a large margin in detecting poisoned samples formed by LCBD, which is different from the SIG attacks. On the other hand, the aggregation of median and mean denoising still has a comparable detection rate to

β	BDR (Median)	BDR (Mean)	BDR (Both)
0.00	0.00%	0.00%	0.00%
0.05	3.38%	6.76%	3.38%
0.10	9.46%	11.49%	9.46%
0.15	14.86%	19.59%	15.54%
0.20	25.67%	29.73%	25.00%
0.25	29.05%	36.49%	29.73%
0.30	41.89%	42.57%	39.86%

Table 12. Ablation study of denoising functions and β for detection on the whole training dataset (SIG)

Attack type	BDR (Median)	BDR (Mean)	BDR (Both)
GAN ($\tau = 0.0$)	75.60%	51.20%	72.70%
GAN ($\tau = 0.1$)	73.25%	41.80%	70.95%
GAN ($\tau = 0.2$)	80.15%	40.20%	78.00%
GAN ($\tau = 0.3$)	83.15%	41.55%	79.75%
AE ($\ell_2, \epsilon = 300$)	77.65%	36.85%	75.55%
AE ($\ell_2, \epsilon = 600$)	89.45%	45.95%	88.65%
AE ($\ell_2, \epsilon = 1200$)	98.80%	78.75%	98.55%
AE ($\ell_\infty, \epsilon = 8$)	77.55%	42.05%	75.80%
AE ($\ell_\infty, \epsilon = 16$)	89.40%	40.95%	88.55%
AE ($\ell_\infty, \epsilon = 32$)	97.40%	44.75%	97.15%

Table 13. Ablation study of denoising functions for detection on the poisoned class (LCBD)

β	BDR (Median)	BDR (Mean)	BDR (Both)
0.00	0.00%	0.00%	0.00%
0.05	77.35%	0.00%	59.50%
0.10	89.40%	0.30%	88.80%
0.15	93.25%	0.30%	93.00%
0.20	94.90%	0.60%	94.65%
0.25	96.05%	1.15%	96.35%
0.30	97.65%	2.50%	97.40%

Table 14. Ablation study of denoising functions and β for detection on the whole training dataset (LCBD)

the single “median” denoising. This trend is even prominent for the detection on the whole dataset where employing the single “mean” denoising images can barely detect any poisoned sample. Meanwhile, we found the post-clean ASR of employing the single “median” and the aggregation version are close, since they detect almost the same poisoned samples. In contrast to SIG, defending against LCBD is not too sensitive to the selection of β . As can be seen in Table 14, UltraClean still detects 88.80% poisoned samples when β is small.

HTBD. Unlike SIG and LCBD, for the detection on the poisoned class, HTBD attacks are sensitive to the single

Target Class		BDR (Median)	BDR (Mean)	BDR (Both)
1	Terrier	92.00%	86.00%	97.00%
2	Bee	84.00%	86.00%	86.00%
3	Plunger	84.00%	79.00%	86.00%
4	Partridge	85.00%	78.00%	87.00%
5	Ipod	91.00%	92.00%	96.00%
6	Deerhound	100.00%	72.00%	96.00%
7	Cockatoo	99.00%	91.00%	99.00%
8	Toyshop	71.00%	70.00%	72.00%
9	Tiger beetle	100.00%	94.00%	100.00%
10	Goblet	79.00%	69.00%	86.00%

Table 15. Ablation study of denoising functions for detection on the poisoned class (HTBD)

β	BDR (Median)	BDR (Mean)	BDR (Both)
0.00	0.00%	0.00%	0.00%
0.01	65.50%	12.00%	57.25%
0.02	79.25%	19.75%	68.75%
0.03	87.00%	25.25%	78.00%
0.04	91.00%	29.50%	83.50%
0.05	93.25%	33.25%	88.25%

Table 16. Ablation study of denoising functions and β for detection on the whole training dataset (HTBD)

“median” and the single “mean” denoised baseline images. However, the aggregation version consistently achieves the highest detection rate for all target classes, as shown in Table 15. We can observe from Figure 5 that the performance of the post-clean ASR is similar to the detection rate, where the aggregation version always performs the best. For the detection on the whole dataset, applying both the aggregation version and the single “median” denoised baseline images detect considerable poisoned samples while applying the single “mean” denoising only detects a few of them. UltraClean is resistant to the selection of β in defending against HTBD, only a removal threshold of 5% is capable of entirely mitigating HTBD from models.

In sum, all six dirty-label and clean-label attacks reveal inconsistent sensitivity to different denoised baseline images. Only the aggregation of both functions demonstrates effectiveness against all the attacks. Overall, the differences of exploiting each single denoised baseline image in detecting different backdoor attacks further validate the necessity of integrating both denoising functions in the proposed methodology. Similarly, UltraClean’s sensitivity to removal threshold is different against different attacks. However, a value of 0.3 is good enough to handle all attacks.

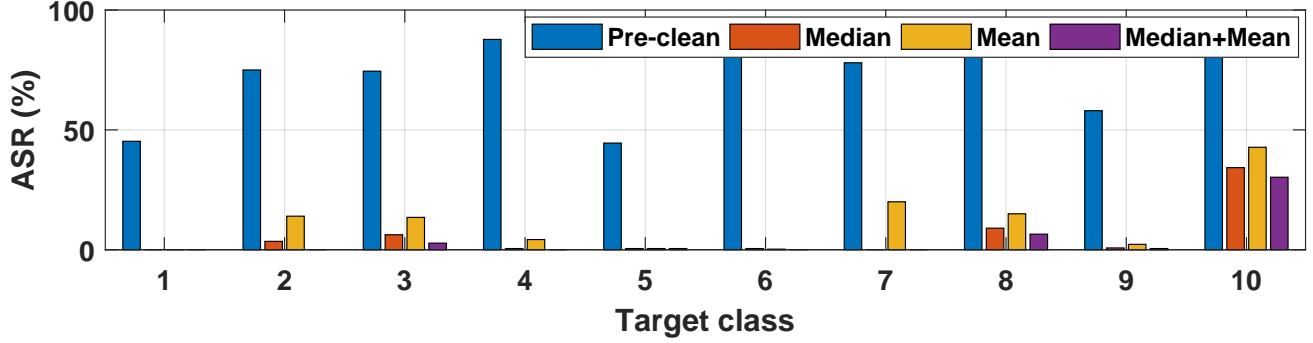


Figure 5. ASR comparison of using the single “median” baseline images, single “mean” baseline images and aggregation of median and mean (HTBD).

Attack Type		Acc. (UC)	BDR (UC)	ASR (UC)
BadNets	$\mathcal{T}_s=1 \times 1$	83.49%	92.34%	0.59%
	$\mathcal{T}_s=2 \times 2$	84.34%	100.00%	0.82%
	$\mathcal{T}_s=3 \times 3$	83.91%	94.42%	0.83%
Trojan	$\mathcal{T}_t=0.1$	84.83%	32.34%	0.36%
	$\mathcal{T}_t=0.3$	84.83%	98.66%	0.40%
	$\mathcal{T}_t=0.5$	84.73%	99.50%	1.61%
Blended (HK)	$\alpha=0.1$	84.34%	96.00%	1.30%
	$\alpha=0.2$	85.08%	97.38%	3.06%
	$\alpha=0.3$	85.00%	98.44%	10.23%
Blended (RP)	$\alpha=0.1$	84.23%	99.32%	0.70%
	$\alpha=0.2$	84.23%	99.80%	0.15%
	$\alpha=0.3$	83.92%	99.78%	0.11%

Table 17. UltraClean against adaptive attacks (dirty-label)

D. Robustness against Adaptive Attacks

Attackers may perform adaptive attacks by adjusting the poisoning ratio, changing trigger size, varying blended or transparency ratio, or altering adversarial perturbation to evade defenses. We extensively evaluate UltraClean’s robustness to multiple adaptive attacks and find none of the attacks successfully bypass the detection of UltraClean. UltraClean consistently detects most poisoned samples and mitigates ASR by a large margin against both dirty-label and clean-label adaptive attacks. Detailed results are presented as follows.

D.1. Adaptive Attacks of Dirty-Label Attacks

We first evaluate UltraClean’s performance against adaptive attacks of dirty-label attacks. In order to bypass the defense of UltraClean, attackers can reduce backdoor trigger size or trigger blended/transparency ratio to craft harder-to-detect poisoned samples. Note that attackers tend not to reduce the poisoning ratio for dirty-label attacks since it is usually below 10% to evade label sanitization defense [48].

A lower poisoning ratio may render a failure of attacks. We conduct the adaptive attacks by adjusting the trigger size (\mathcal{T}_s) of BadNets from 1×1 to 3×3 , the trigger transparency (\mathcal{T}_t) of Trojan from 0.1 to 0.5 and blended ratio (α) of Blended from 0.1 to 0.3. The results are summarized in Table 10. We can see that while most attacks can achieve high ASR even with harder-to-detect triggers, UltraClean demonstrates exceptional robustness to various adaptive attacks. UltraClean identifies at least 92% poisoned samples for almost all adaptive attacks and successfully mitigates backdoors in all models. The only exception is the Trojan attack with $\mathcal{T}_t = 0.1$, where the attack achieves 9.43% ASR even without a defense. Although UltraClean only detects 32% of the poisoned data, it still completely remove the backdoor. **The experiment results validates UltraClean’s robustness against dirty-label adaptive attacks.**

Poisoning Ratio	Acc. (PC)	ASR (PC)	BDR (UC)	Acc. (UC)	ASR (UC)
0.1	88.48%	99.01%	95.80%	88.27%	1.69%
0.2	87.98%	99.76%	97.30%	87.89%	1.49%
0.3	88.02%	99.98%	97.06%	87.81%	1.38%
0.4	87.73%	99.98%	98.55%	87.26%	1.10%
Trigger Amplitude	Acc. (PC)	ASR (PC)	BDR (UC)	Acc. (UC)	ASR (UC)
16	87.44%	1.12%	90.35%	86.45%	1.04%
32	87.39%	99.90%	98.70%	87.32%	0.88%
64	87.73%	99.98%	98.55%	87.26%	1.10%

Table 18. UltraClean against adaptive attacks (LCBD)

D.2. Adaptive Attacks of Clean-Label Attacks

LCBD. In the main manuscript, we extensively evaluated UltraClean against LCBD with GAN-based and AE-based approaches. UltraClean has shown superior performance in detecting and mitigating the LCBD backdoor. Here, we further study the effectiveness of UltraClean against adaptive attacks. Attackers may change poisoning ratio and trigger

Target Class		BDR ($\epsilon=8$) ($\mathcal{T}_s=30$)	BDR ($\epsilon=32$) ($\mathcal{T}_s=30$)	BDR ($\epsilon=16$) ($\mathcal{T}_s=15$)	BDR ($\epsilon=16$) ($\mathcal{T}_s=60$)
1	Terrier	100.00%	100.00%	99.50%	99.75%
2	Bee	95.25%	96.50%	93.75%	93.50%
3	Plunger	94.00%	94.75%	94.25%	94.00%
4	Partridge	97.50%	96.25%	96.00%	97.00%
5	Ipod	99.25%	99.00%	99.00%	99.50%
6	Deerhound	99.75%	98.75%	100.00%	97.25%
7	Cockatoo	99.75%	99.25%	96.00%	99.00%
8	Toyshop	89.50%	88.25%	92.75%	88.25%
9	Tiger beetle	98.00%	97.75%	99.00%	96.25%
10	Goblet	94.75%	95.00%	92.75%	93.75%

Table 19. Detection performance of UltraClean against adaptive attacks (HTBD)

amplitude to make LCBBD more stealthy. In the previous experiments, we injected 40% poisoned sample into the target class. Although the poisoning ratio (i.e., the fraction of poisoned data injected into the training dataset) over the entire dataset is small, 40% is a relatively high poisoning ratio over the poisoned class. The original LCBBD paper [58] shows that with lower poisoning ratios (lower poisoning ratios usually render higher stealthiness), it can still achieve a high attack success rate. Thus, we investigate if UltraClean is still effective with lower poisoning ratios. We perform the experiment with the setting “AE (ℓ_2 , $\epsilon = 1200$)” since it achieves the best attack result. As shown in Table 18, even with 10% poisoned data in the target class (only 1% poisoning ratio over the entire training dataset), LCBBD still achieves $\sim 99\%$ ASR. Even under this circumstance, UltraClean still demonstrates a stable detection performance, reaching at least a 95% detection rate and reducing the ASR by over 98%. Another critical factor to make the attack more stealthy is the trigger amplitude. Smaller amplitude means less trigger visibility. In the previous experiments, we set the amplitude to 64 to secure the attack success rate. Here we reduce the amplitude to 16 and 32 to examine if UltraClean can still detect poisoned samples. The experiment results are summarized in the last two rows in Table 18. We find that UltraClean still detects at least 90% poisoned samples, even with an extremely low trigger amplitude. Note that with a 16 trigger amplitude, the attack has failed to inject an effective backdoor since the pre-clean ASR is only 1.12%. **The results further demonstrate the effectiveness of UltraClean against different attack settings, indicating its robustness in protecting neural networks against various adaptive attacks.**

HTBD. For HTBD attacks, there are two crucial parameters: the maximum perturbation ϵ and the trigger size (\mathcal{T}_s), which affect the performance and stealthiness of poisoned samples. Attackers may try to evade or break (i.e., achieve a higher ASR in the presence of detection) the defense of

UltraClean by adjusting these parameters. Here we vary these two parameters to study their effects on UltraClean. Results are presented in Tables 19 and 20. Recall that HTBD generates poisoned samples by adding perturbations to poisoned images to match source images’ representations in the feature space. Therefore, ϵ in the HTBD is akin to the adversarial perturbations of the LCBBD, which is the upper bound of perturbation imposed on the image. The trigger patched on source images affects their representations in the feature space. Thus the trigger size is also an important parameter for poisoned data generation. According to [52], a larger trigger size means larger perturbations added on poisoned images and always leads to better attack efficiency. As shown in the results, **UltraClean shows highly consistent performance against all adaptive attacks of different perturbation and trigger size.** In all experiments, UltraClean detects almost all poisoned samples and reduces the post-clean ASR to 0% in most cases.

E. Comparison to Other SOTA Defenses

We have shown that UltraClean outperforms STRIP and SVD, which are poison-filtering-based defenses. To better demonstrate the effectiveness of UltraClean, we compare it to other SOTA defenses ABL [39] and ANP [63], which are **not** poison-filtering-based. Results are summarized in Table 21. ABL is a poison suppression-based defense that eliminates backdoors by unlearning the poisoned samples in the dataset. ANP is a model reconstruction-based defense that prunes some sensitive neurons to purify the injected backdoor. We conduct the experiment against BadNets on CIFAR-10. It can be seen that UltraClean achieves comparable clean accuracy and ASR while cleansing the dataset for reusability, as opposed to ABL and ANP that do not remove poisoned data.

F. Effectiveness against Sleeper Agent

We evaluate the effectiveness of UltraClean against various prestigious attacks. We notice the recently proposed clean-label backdoor attack Sleeper Agent (SA) [54] employs a different attack mechanism from previous attacks, which crafts poisoned samples via gradient matching [20]. We evaluate UltraClean’s performance against this latest type of attack in Table 22. We carry out the experiment on CIFAR-10 using ResNet-18. It can be seen that UltraClean still significantly reduces the ASR and successfully mitigate the backdoor effect, which again validates that UltraClean is effective against both clean and dirty-label attacks regardless of attack mechanisms.

G. Study of Other Denoising Methods

We evaluate other denoising methods and present the results in Table 23. We perform the experiment against BadNets on

$\epsilon=16$ $\mathcal{T}_s=30$	Class ID									
	1	2	3	4	5	6	7	8	9	10
Acc.(%) (Pre)	96.00	97.00	95.00	97.00	95.00	95.00	96.00	95.00	98.00	95.00
Acc.(%) (UC)	100.00	99.00	97.00	100.00	100.00	100.00	98.00	98.00	100.00	99.00
ASR(%) (Pre)	45.25	75.00	74.5	87.75	44.50	83.75	78.00	80.50	58.00	89.75
ASR(%) (UC)	0.00	0.00	4.75	0.00	0.00	0.25	0.00	4.50	0.00	27.50
$\epsilon=8$ $\mathcal{T}_s=30$	Class ID									
	1	2	3	4	5	6	7	8	9	10
Acc.(%) (Pre)	98.00	99.00	95.00	98.00	96.00	90.00	96.00	96.00	99.00	98.00
Acc.(%) (UC)	100.00	100.00	97.00	100.00	100.00	99.00	99.00	96.00	100.00	99.00
ASR(%) (Pre)	33.75	78.75	78.25	85.25	80.00	41.75	32.00	79.50	87.00	89.75
ASR(%) (UC)	0.00	0.00	2.75	0.00	0.50	0.00	0.00	6.50	0.50	3.25
$\epsilon=32$ $\mathcal{T}_s=30$	Class ID									
	1	2	3	4	5	6	7	8	9	10
Acc.(%) (Pre)	97.00	99.00	96.00	99.00	93.00	91.00	98.00	95.00	100.00	95.00
Acc.(%) (UC)	100.00	100.00	97.00	100.00	100.00	99.00	99.00	96.00	100.00	99.00
ASR(%) (Pre)	29.00	80.75	81.25	88.00	77.50	72.25	35.50	83.25	85.25	87.00
ASR(%) (UC)	0.00	0.00	3.50	0.00	0.50	0.00	0.00	7.75	0.00	1.50
$\epsilon=16$ $\mathcal{T}_s=15$	Class ID									
	1	2	3	4	5	6	7	8	9	10
Acc.(%) (Pre)	97.00	98.00	96.00	99.00	96.00	92.00	97.00	97.00	100.00	96.00
Acc.(%) (UC)	100.00	100.00	97.00	100.00	99.00	100.00	98.00	96.00	100.00	99.00
ASR(%) (Pre)	40.25	66.25	80.75	89.00	75.00	30.75	86.00	77.75	84.75	97.00
ASR(%) (UC)	0.00	8.50	4.50	0.00	0.50	0.00	0.00	8.75	0.25	6.00
$\epsilon=16$ $\mathcal{T}_s=60$	Class ID									
	1	2	3	4	5	6	7	8	9	10
Acc.(%) (Pre)	96.00	99.00	96.00	98.00	97.00	93.00	97.00	96.00	99.00	96.00
Acc.(%) (UC)	100.00	100.00	97.00	100.00	100.00	99.00	99.00	96.00	100.00	99.00
ASR(%) (Pre)	47.50	78.25	77.25	63.75	53.75	83.00	77.75	83.75	85.75	91.25
ASR(%) (UC)	0.00	0.00	6.00	1.50	0.50	0.00	8.25	10.50	1.00	30.00

Table 20. Accuracy and ASR of UltraClean against adaptive attacks (HTBD)

	Acc.	ASR	BDR
Pre-Clean	87.73%	99.98%	-
ABL [39]	89.03%	0.00%	-
UltraClean	85.00%	1.59%	97.40%
Pre-Clean	93.51 %	99.92%	-
ANP [63]	90.20%	0.45%	-
UltraClean	93.61%	0.00%	100%

Table 21. Comparison of UltraClean to ABL against LCBF attack and ANP against BadNets on CIFAR-10 using ResNet.

	Acc.	ASR	BDR
Sleeper Agent [54]	92.31%	85.27%	-
UltraClean	91.50%	5.66%	50.60%

Table 22. Performance of UltraClean against Sleeper Agent attack on CIFAR-10 using ResNet-18.

CIFAR-10 using VGG16. Compared to our current method that employs local median and non-local mean denoising, incorporating more denoising methods only slightly improves or even degrades the performance. Therefore, we argue that it is unnecessary to introduce more denoising methods that may significantly increase the complexity of our defense.

	Ours	Ours+Gaussian	Ours+Bilateral
BDR	94.42%	95.16%	83.80%
ASR	0.89%	0.76%	100%

Table 23. Comparison of denoising methods against BadNets on CIFAR10 using VGG16.

	GTSRB	CIFAR-10	ImageNet
Total Time	~398 (s)	~262 (s)	~57 (hrs)
#Training Images	4,772	50,000	1,081,167

Table 24. Time consumption of detection over different datasets

H. Time Cost of Detection

We also analyze the efficiency of UltraClean and present the time consumption over each attack and dataset on our machine in Table 24. It can be seen that the proposed denoising operations are fast to process images on different datasets. Even for the ImageNet dataset, it only takes less than 0.2 seconds to process one image and about 57 hours to process the entire dataset (there are way more images than other datasets). Note that all the data are processed with each denoising operation in sequence; applying parallel processing may significantly reduce the time cost.