A. More Experimental Settings

A.1. Datasets and Classifiers

The datasets and DNN models used in our experiments are summarized in Table 4.

A.2. Details of Baseline Implementations

We implemented the baselines including FP[1], MCR[2], NAD[3] ABL[4], and DBD [5] with their open-sourced codes. For Fine-pruning (FP), we pruned the last convolutional layer of the model. For model connectivity repair (MCR), we trained the loss curve for 100 epochs using the backdoored model as an endpoint and evaluated the defense performance of the model on the loss curve. As for the Neural Attention Distillation (NAD), we finetuned the backdoored student network for 10 epochs with 5% of clean data. The distillation parameter for CIFAR-10 was set to be identical to the value given in the original paper. We cautiously selected the value of distillation parameter for GTSRB and ImageNet to achieve the best trade-off between ASR and CA. For ABL, we unlearned the backdoored model using the $\mathcal{L}_{GGA}$ loss with 1% isolated backdoor examples and a learning rate of 0.0001. For our DBD, we adopt SimCLR as the self-supervised method and MixMatch as the semi-supervised method. The filtering rate is set to 50% as suggested by the original paper.

A.3. Details of CBD Implementations

In CBD, f_C is trained on poisoned datasets for 100 epochs using Stochastic Gradient Descent (SGD) with an initial learning rate of 0.1 on CIFAR-10 and the ImageNet subset (0.01 on GTSRB), a weight decay of 0.0001, and a momentum of 0.9. The learning rate is divided by 10 at the 20th and the 70th epoch. D_ϕ is set as a MLP with 2 layers. The dimensions of the embedding r and z are set as 64.

B. More Experimental Results

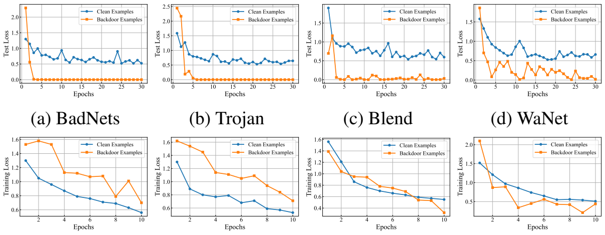
B.1. Results of Adaptive Attacks

The pseudo codes of adaptive attacks against CBD are shown in Algorithm 2. The results of adaptive attacks with different kinds of backdoor are shown in Table 5. We also show the curves of training losses on clean/backdoor examples in the optimization of added noise along with the vanilla training for reference. In Figure 4, we can observe that the training losses of backdoor examples reaches almost zero after several epochs of training (first line) while our adaptive attack strategy managed to increase the losses of backdoor examples in the optimization process (second line). The above observation indicates that the backdoor examples are much easier to learn than clean examples in vanilla training. The adaptive attack can slow the injection of backdoor and try to make the backdoor attack stealthier to bypass CBD. However, our CBD can still defend the adaptive attack successfully (Table 5). The reason is most probably that the optimized noise becomes less effective when the model is retrained and the model parameters are randomly initialized. In another word, the optimized noise is not transferable.

Algorithm 2 Adaptive Attack to CBD

Input: Model f_θ, poisoned dataset $\mathcal{D}'$, clean dataset $\mathcal{D}$, perturbation range ϵ, number of training iterations T, step size α, update steps M.
Output: optimized poisoned dataset $\mathcal{D}'$

```
Initialize $f_\theta$
for $t = 1, \cdots, T$ do
    Draw a mini-batch $\mathcal{B} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ from $\mathcal{D} \cup \mathcal{D}'$
    $\theta \leftarrow \theta - \eta \nabla_\theta \sum_{(x,y)\in\mathcal{B}} \mathcal{L}(f_\theta(x), y)$
    for $(x_i, y_i)$ in $\mathcal{D}'$ do
        for $m = 1, \cdots, M$ do
            $x_i \leftarrow \Pi_\epsilon(x_i + \alpha \cdot \nabla_x \mathcal{L}(f_\theta(x_i), y_i))$
        end for
    end for
end for
```



(a) BadNets (b) Trojan (c) Blend (d) WaNet

(e) BadNets (f) Trojan (g) Blend (h) WaNet

Figure 4. The curve of training losses on clean/backdoor examples in the vanilla training (first line) and in the optimization of adaptive attacks (second line). This experiment is conducted with WideResNet-16-1 for CIFAR-10 under poisoning rate 10%.

B.2. Results with Different Model Architectures

Note that our CBD is agnostic to the choice of model architectures. In the main text, we report the results with WideResNet-16-1 and ResNet-34. Here, in Table 6 and 7, we show experimental results on CIFAR-10 with WideResNet-40-1 [69] and th T2T-ViT [40] under poisoning rate 10%. We can observe that CBD can still greatly

[1] https://github.com/kangliucn/Fine-pruning-defense
[2] https://github.com/IBM/model-sanitization
[3] https://github.com/bboylyg/NAD
[4] https://github.com/bboylyg/ABL
[5] https://github.com/SCLBD/DBD

Table 4. Details of datasets and classifiers in the paper

Dataset	Labels	Input Size	Training Images	Classifier
CIFAR-10	10	32 x 32 x 3	50000	WideResNet-16-1
GTSRB	43	32 x 32 x 3	39252	WideResNet-16-1
ImageNet subset	12	224 x 224 x 3	12406	ResNet-34

Table 5. Attack success rate (ASR %) and clean accuracy (CA %) of Adaptive Attacks.

Defense	BadNets		Trojan		Blend		WaNet	
	ASR	CA	ASR	CA	ASR	CA	ASR	CA
None	99.62	84.55	99.85	84.32	97.63	84.45	97.24	85.47
CBD	4.31	84.19	3.77	84.37	2.57	84.49	5.19	85.33

reduce the attack success rate and keep clean accuracy with different model architectures.

Table 6. Results on CIFAR10 with WideResNet-40-1. We show attack success rates (ASR %) and clean accuracy (CA %).

Defense	BadNets		Trojan		Blend		WaNet	
	ASR	CA	ASR	CA	ASR	CA	ASR	CA
None	100	92.96	100	93.21	99.83	92.69	98.35	92.88
CBD	0.95	92.54	1.04	92.70	1.32	92.17	2.54	92.26

Table 7. Results on CIFAR10 with T2T-ViT. We show the attack success rates (ASR %) and the clean accuracy (CA %).

Defense	BadNets		Trojan		Blend		WaNet	
	ASR	CA	ASR	CA	ASR	CA	ASR	CA
None	100	85.65	100	85.86	99.42	85.66	99.30	86.27
CBD	0.89	86.05	0.97	86.61	1.59	85.82	3.80	85.94

B.3. The computational time of other defenses.

Table. 8 shows the total computational time of defense methods against BadNets. As the methods belong to different categories, we count the time to train backdoored models for FP, MCR, and NAD for a fair comparison. Generally, the time cost of CBD is acceptable.

Table 8. The total computational time (seconds) on CIFAR10 with WRN-16-1. The percentages in parentheses indicate the relative increase compared to no defence (*None*).

None	FP	MCR	NAD	ABL	DBD	CBD (ours)
1152	1515(31.5%)	3445(127.4%)	1306(13.4%)	1383(20.0%)	5280(358.3%)	1317(14.3%)

C. Details of Derivations

Here we show the details of derivation with respect to Equ. 4. Since the $p(z|x) = \mathcal{N}(\mu(x), \text{diag}\{\sigma^2(x)\})$ and $p(x) = \mathcal{N}(0,1)$ are multivariate Gaussian distributions with independent components, we only need to derive the case with univariate Gaussian distributions. For the univariate case, we have:

$$\begin{aligned}
&D_{\text{KL}}(\mathcal{N}(\mu,\sigma^2)||\mathcal{N}(0,1))\\
&= \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \left(\log \frac{e^{-(x-\mu)^2/2\sigma^2}/\sqrt{2\pi\sigma^2}}{e^{-x^2/2}/\sqrt{2\pi}} \right) dx\\
&= \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \log \left\{ \frac{1}{\sigma} \exp \left\{ \frac{1}{2} [x^2 - (x-\mu)^2/\sigma^2] \right\} \right\} dx\\
&= \frac{1}{2} \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} [-\log\sigma^2 + x^2 - (x-\mu)^2/\sigma^2] dx\\
&= \frac{1}{2}(-\log\sigma^2 + \mu^2 + \sigma^2 - 1).
\end{aligned} \tag{11}$$

The final equation of Equ. 11 holds because $-\log\sigma^2$ is a constant; the term x^2 is the second order moment of the Gaussian distribution and equals to $\mu^2+\sigma^2$ after integration; the $(x-\mu)^2$ in the third term calculates the variance and equals to σ^2 after integration ($-\frac{\sigma^2}{\sigma^2} = -1$). For the results of multivariate Gaussian distributions, we have:

$$\begin{aligned}
&D_{\text{KL}}(p(z|x)||q(z))\\
&= D_{\text{KL}}(\mathcal{N}(\mu(x), \text{diag}\{\sigma^2(x)\})||\mathcal{N}(0,1))\\
&= \frac{1}{2} \sum_d (-\log\sigma_d^2 + \mu_d^2 + \sigma_d^2 - 1)\\
&= \frac{1}{2}||\mu(x)||_2^2 + \frac{1}{2} \sum_d (\sigma_d^2 - \log\sigma_d^2 - 1).
\end{aligned} \tag{12}$$

Therefore, Equ. 4 in the main text has been proved.