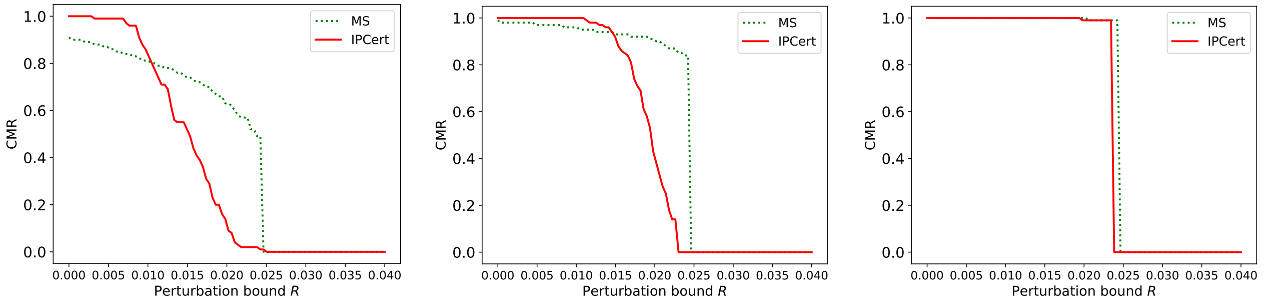
A. Appendix

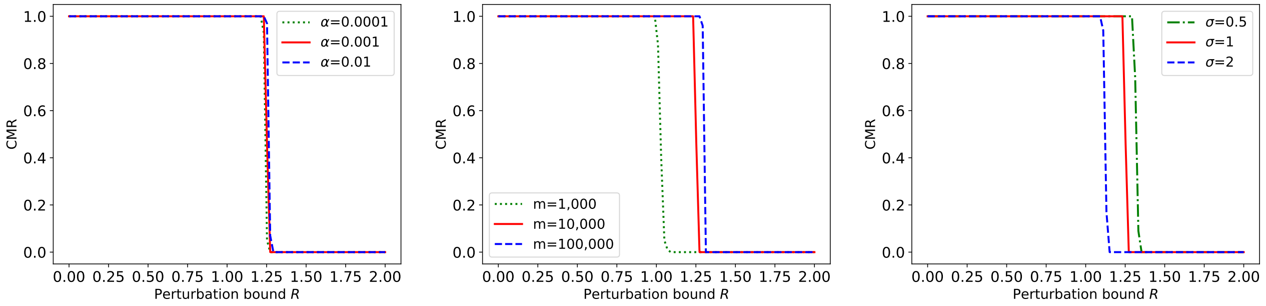
A.1. Parameter Settings of Fine-tuning and Distillation

For both fine-tuning and distillation, we use *cross-entropy* loss and SGD optimizer with a learning rate of 0.001/0.01 for 50 epochs. We use another half of training data for post-processing. The training batch size is set to 256. For fine-tuning, the attacker uses the labeled training data. For distillation, the attacker uses the target model to label the training inputs, and then fine-tunes the target model using the labeled training data.



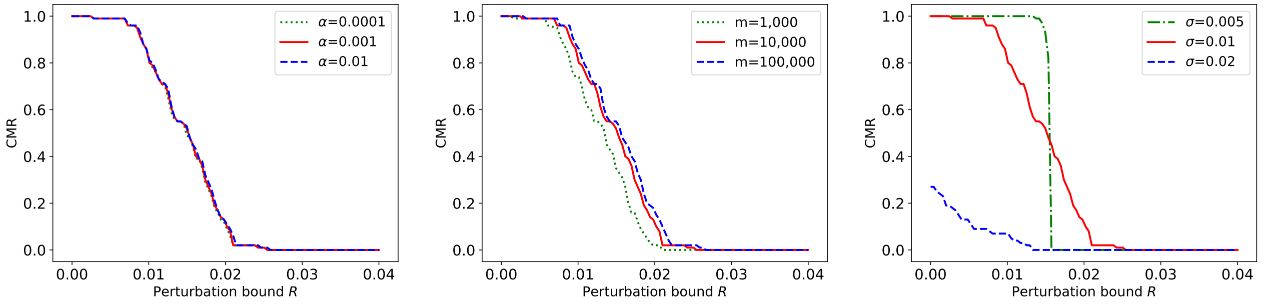
(a) CIFAR-10 (b) CIFAR-100 (c) MNIST

Figure 5. CMR of MS and IPCert for fingerprint on the three datasets.



(a) α (b) m (c) σ

Figure 6. Impact of parameters α, m, and σ on IPCert for watermark on CIFAR-10 dataset.

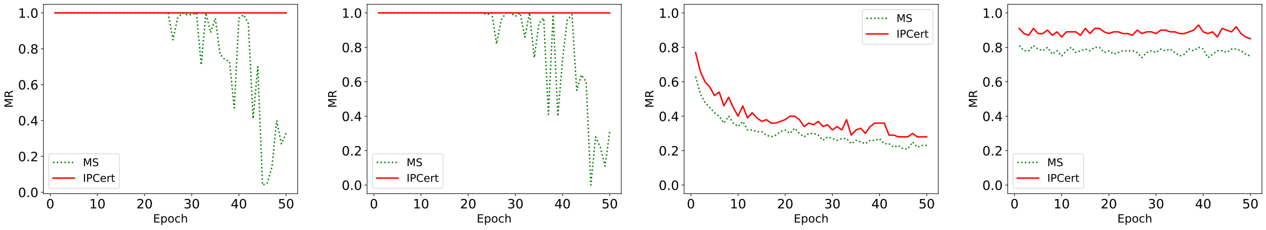


(a) α (b) m (c) σ

Figure 7. Impact of parameters α, m, and σ on IPCert for fingerprint on CIFAR-10 dataset.

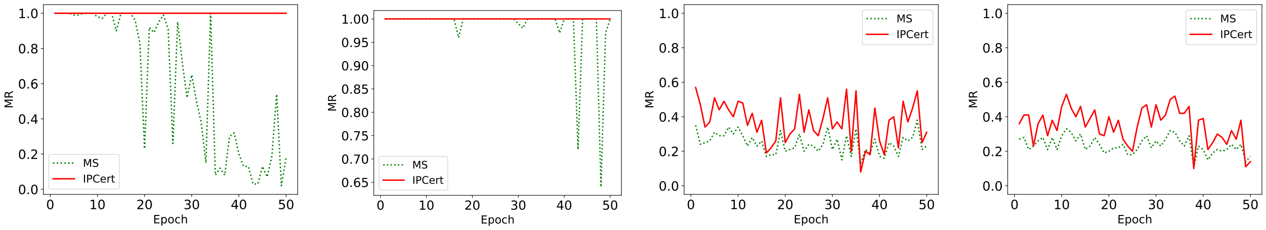
A.2. Additional Results

We also compare IPCert with MS on different datasets. The results in Figure 8 and 9 show that for each dataset, IPCert performs better than MS for both watermark and fingerprint against fine-tuning and distillation.



(a) Watermark with fine-tuning (b) Watermark with distillation (c) Fingerprint with fine-tuning (d) Fingerprint with distillation

Figure 8. MR of MS and IPCert against fine-tuning and distillation on CIFAR-100.



(a) Watermark with fine-tuning (b) Watermark with distillation (c) Fingerprint with fine-tuning (d) Fingerprint with distillation

Figure 9. MR of MS and IPCert against fine-tuning and distillation on MNIST.

A.3. Adaptive Post-processing to IPCert

In Section 3, we design a post-processing method for existing IP protection methods. We adapt this method to IPCert. We call the adaptive method *ensemble post-processing*. We note that IPCert is provably robust, which means that the CMR is the lower bound of MR when the perturbation added to the target model is bounded, no matter what post-processing (including adaptive ones) is used to find the bounded perturbation. In IPCert, multiple noisy models are used to compute MR. Therefore, we design a method that attacks multiple noisy models simultaneously. Specifically, attacker adds s (we use 16 in our experiments) Gaussian noises to the target model θ and gets noisy models $\theta_1, \theta_2, \cdots, \theta_s$. Then, the attacker replaces the term $l(\theta + \delta, x, y)$ as the term $\sum_{k=1}^{s} l(\theta_k + \delta, x, y)$ in the optimization problem in Eq. 2, and solves the optimization problem to find the perturbation δ. Table 4 shows the perturbation bounds that our post-processing method in Section 3 and our ensemble post-processing can reduce MR of IPCert to 0. The post-processing in Section 3 requires adding perturbation 12.57 to reduce MR of IPCert to 0, while the adaptive ensemble post-processing only requires perturbation 4.56 to reduce MR of IPCert to 0. Our results show that the adaptive post-processing is stronger than the post-processing in Section 3 to attack IPCert. The reason is that the post-processing in Section 3 is designed to attack watermark/fingerprint, but not IPCert.

Table 4. Perturbation bound R that reduces the MR of IPCert to be 0 for watermark on CIFAR-10. R is the perturbation bound that our post-processing in Section 3 can reduce MR of IPCert to 0; and Ensemble R is the perturbation bound that our ensemble post-processing can reduce MR of IPCert to 0.

R	Ensemble R
12.57	4.56

Algorithm 2 Training with noise

```
Input: Training dataset $D$, IP dataset $D_{IP}$, learning rate $lr$, noise level $\sigma$, training epochs $N$, the number of noise $k$, and the
    times of training with noise $t$.
Output: Target model parameter $\theta$
for epoch = 1 to $N$ do
    StandardTrain($\theta$, $D$)
    for $i$ = 1 to $t$ do
        $\epsilon \leftarrow \frac{i}{t}\sigma$
        $\epsilon_1, \epsilon_2, \cdots, \epsilon_k \sim (0, \epsilon^2 I)$
        $g_\theta \leftarrow \frac{1}{k|D_{IP}|} \sum_{j=1}^{k} \sum_{(x,y)\in D_{IP}} [\nabla_\theta l(\theta + \epsilon_j, x, y)]$
        $\theta \leftarrow \theta - lr \cdot g_\theta$
    end for
end for
```

Algorithm 3 Selecting an IP data point in fingerprinting

```
Input: Model $\theta$, learning rate $lr$, trade-off hyperparameter $\beta$, noise level $\sigma$, hinge parameter $\gamma$, label $j$, and maximum number
    of iterations max_iter.
Output: IP data point $x$
$x \leftarrow$ random initialization
for $iter$ = 1 to max_iter do
    if $||\nabla_x L(x)||_2 \leq$ 1e-7 then
        break
    end if
    $x \leftarrow x - lr \cdot \nabla_x L(x)$
end for
```