## A. Proof

Here we define the margin loss.

$$l_\gamma(f(x),y) = \phi_\gamma(f(x)_y - \max_{i\neq y} f(x)_i), \quad \text{where } \phi_\gamma(t) = \begin{cases} 1, & t<0 \\ 1-\gamma t, & 0 \le t \le 1/\gamma \\ 0, & t > 1/\gamma \end{cases} \tag{9}$$

**Lemma 1.** *Let $l_\gamma(x,y) : \mathbb{R}^c \times Y \to \mathbb{R}$ and $\phi_\gamma(x) : \mathbb{R} \to \mathbb{R}$ where*

$$\begin{aligned} & l_\gamma(x,y) = \phi_\gamma[f(x)_y - \max_{i\neq y} f(x)_i], \\ & \textit{where } \phi_\gamma(x) = \begin{cases} 1, & x<0 \\ 1-\gamma x, & 0 \le x \le 1/\gamma \\ 0, & x > 1/\gamma \end{cases} . \end{aligned} \tag{10}$$

*For any $x_1, x_2 \in R^c$ and $y \in Y$, we have*

$$|l_\gamma(x_1,y) - l_\gamma(x_2,y)| \le 2\gamma ||x_1 - x_2||_\infty \tag{11}$$

*Proof.* It is apparent that $\phi_\gamma(x) : \mathbb{R} \to \mathbb{R}$ is $\gamma$ Lipschitz. And we have

$$\begin{aligned} & |l_\gamma(x_1,y) - l_\gamma(x_2,y)| \\ =& \left|\phi_\gamma\Big[(f(x_1)_y - f(x_2)_y) - (\max_{i\neq y} f(x_1)_i - \max_{j\neq y} f(x_2)_j)\Big]\right| \\ \le& \gamma\big|(f(x_1)_y - f(x_2)_y) - (\max_{i\neq y} f(x_1)_i - \max_{j\neq y} f(x_2)_j)\big| \\ \le& \gamma\big|(f(x_1)_y - f(x_2)_y)\big| + \gamma\big|\max_{i\neq y} f(x_1)_i - \max_{j\neq y} f(x_2)_j\big| \\ \le& 2\gamma||f(x_1) - f(x_2)||_\infty \end{aligned} \tag{12}$$

□

**Lemma 2.** *Assume the softmax function $s : \mathbb{R}^c \to \mathbb{R}^c$ is defined as follows.*

$$s(x)_i = \frac{\exp(x_i)}{\sum_i \exp(x_i)} \tag{13}$$

*Then we have*

$$||s(x) - s(y)||_\infty \le \frac{1}{2}||x-y||_\infty . \tag{14}$$

*Moreover we have*

$$\begin{aligned} & \big\|(s(x) - s(y)) - (s(x') - s(y'))\big\|_\infty \le \\ & \frac{1}{2}(\|x-x'\|_\infty + \|y-y'\|_\infty) \end{aligned} \tag{15}$$

*In other words, $s(x)$ and its residual form $s(x) - s(y)$ are both $1/2$-Lipschitz with respect to $L_\infty$-norm.*

*Proof.* From the definition of Lipschitz continualty, we have $|s(x) - s(y)|_\infty \le L|x-y|_\infty$. Note that $|s(x) - s(y)|_\infty \le \sup_i |s_(x)_i - s(y)_i|$.

The gradient is hence the following.

$$\frac{ds(x)_i}{dx_j} = \begin{cases} s(x)_i(1-s(x)_i), & \text{i = j} \\ -s(x)_i s(x)_j, & \text{i} \neq \text{j} \end{cases} \tag{16}$$

Note that $||\nabla_i s(x)||_1 = 2s(x))_i(1-s(x))_i) \leq 1/2$. Let $\preceq$ represents the generalized inequality of the nonnegative orthant. By the mean value theorem, for some $\delta$, s.t. $x \preceq \delta \preceq y$, we have

$$s(x)_i - s(y)_i \leq \nabla_i s(\delta)^T (x-y)\,. \tag{17}$$

By Holder's inequality,

$$\nabla_i s(\delta)^T(x-y) \leq ||\nabla_i s(\delta)||_1 ||x-y||_\infty \leq \frac{1}{2}||x-y||_\infty \tag{18}$$

To prove the second goal, we can find that

$$\begin{aligned}
&||(s(x)-s(y))-(s(x')-s(y'))||_\infty \\
=&||(s(x)-s(x'))-(s(y)-s(y'))||_\infty \\
\leq&||(s(x)-s(x'))||_\infty + (s(y)-s(y')||_\infty \\
\leq&\frac{1}{2}(||x-x'|_\infty + ||y-y'||_\infty)
\end{aligned} \tag{19}$$

□

**Lemma 3** ( [36]). *We add the proof of this lemma here for completeness. Let $f : \mathbb{R} \to \mathbb{R} \in \mathcal{F}$ be a class of real-value functions, $D$ be a set of samples. We define a pseudo-metric $\|\cdot\|_{p,D}$ on functions $\mathcal{F}$ with respect to vector norm $\|\cdot\|_p$ and $n$ samples $|D| = n$, where*

$$\|f\|_{p,D} = \left[\frac{1}{n}\sum_{x\in D}|f(x)|^p\right]^{\frac{1}{p}}. \tag{20}$$

*We define the Covering number $\mathcal{N}(\mathcal{F}, \|\cdot\|_{p,D}, \epsilon)$ as the size of the minimal $\epsilon$-cover of $\mathcal{F}$ with respect to pseudo-norm $\|\cdot\|_{p,D}$. Given*

$$\sup_{f\in F}\|f\|_{2,D} \leq s_D\,, \tag{21}$$

*we have*

$$\hat{R}(\mathcal{F}|D) \leq \inf_{\epsilon\in[0,s_D/2]}\left\{\epsilon + \frac{\sqrt{2}s_D}{\sqrt{n}}\sqrt{\log \mathcal{N}(\mathcal{F}, \|\cdot\|_{1,D}, \epsilon)}\right\} \tag{22}$$

*Proof.* For any $\epsilon$ and $D$, let $\mathcal{C}$ be the minimal-$\epsilon$ cover of $\mathcal{F}$, which means for each function $f$, there exists $f_\epsilon \in \mathcal{C}$ such that $\|f - f_\epsilon\|_{1,D} \leq \epsilon$

$$\begin{aligned}
&\hat{R}(\mathcal{F}|D) \\
=&E_\sigma\left[\frac{1}{n}\sup_{f\in\mathcal{F}}\sum_i \sigma_i f(x_i)\right] \\
\leq&E_\sigma\left[\frac{1}{n}\sup_{f\in\mathcal{F}}\sum_i \sigma_i\left(f(x_i)-f_\epsilon(x_i)\right)\right] + \\
&E_\sigma\left[\frac{1}{n}\sup_{f_\epsilon\in\mathcal{C}}\sum_i \sigma_i f_\epsilon(x_i)\right] \\
\leq&\epsilon + E_\sigma\left[\frac{1}{n}\sup_{f_\epsilon\in\mathcal{C}}\sum_i \sigma_i f_\epsilon(x_i)\right] \\
\leq&\epsilon + \sup_{f\in\mathcal{F}}\sqrt{\sum_{x_i\in D} f(x_i)^2}\,\frac{\sqrt{2\log|\mathcal{C}|}}{n}\text{(Massart's Lemma)} \\
\leq&\epsilon + \frac{\sqrt{2}s_D}{\sqrt{n}}\sqrt{\log\mathcal{N}(\mathcal{F}, \|\cdot\|_{1,D}, \epsilon)}
\end{aligned}$$

Since this bound stands for any $\epsilon$, Thus we have

$$\hat{R}(\mathcal{F}|D) \leq \inf_{\epsilon \in [0, s_D/2]} \left\{ \epsilon + \frac{\sqrt{2} s_D}{\sqrt{n}} \sqrt{\log \mathcal{N}(\mathcal{F}, \|\cdot\|_{1,D}, \epsilon)} \right\} \tag{23}$$

$\square$

**Theorem 1.** *For distilled model $f^k$ and class number $C$, we have*

$$\begin{aligned} &\hat{R}(\mathcal{L}^k|D_s) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i \hat{R}(\mathcal{O}_i^k|D_s) \qquad (24) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i B(\mathcal{O}_i^k|D_s) \qquad (25) \end{aligned}$$

*Proof.* From Lemma 1 and Lemma 2, we know the Lipschitz constant of $l_\gamma \circ s$ is $\gamma$. Using the $\mathcal{L}_\infty$-contraction property of Rademacher complexity [10], we can expand the complexity of loss function to the complexity of logits values in Eq. (24). Eq. (25) is from Lemma 3. $\square$

**Theorem 2.** *For cloned model $f^c$, let $\omega_c = \sup_{H^c \in \mathcal{H}^c} \|H^c\|_{1,\infty}$, $\beta_c = \sup_{H^c \in \mathcal{H}^c, H^* \in \mathcal{H}^*} \|H^c - H^*\|_{1,\infty}$ and class number $C$. We have*

$$\begin{aligned} \hat{R}(\mathcal{L}^c|D_s) \leq \tilde{O}(\gamma\sqrt{C}) \min \big[ \\ \max_i B(\mathcal{O}_i^c|D_s), \max_j \omega_c B(\mathcal{I}_j^c|D_s) + \beta_c B(\mathcal{J}_j^*|D_s) \big] \end{aligned}$$

*Proof.*

$$\begin{aligned} &\hat{R}(\mathcal{L}^c|D_s) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i \hat{R}(\mathcal{O}_i^c|D_s) \qquad (26) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i \hat{R}\big(\{[H^c(g^c(x) - g^*(x)) \\ &\qquad + (H^c - H^*)g^*(x)]_i\}\big) \qquad (27) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \{\sup_{H^c} \|H^c\|_{1,\infty} \max_i \hat{R}(\mathcal{I}_i^c|D_s) \\ &\qquad + \sup_{H^c, H^*} \|H^c - H^*\|_{1,\infty} \max_i \hat{R}(\mathcal{J}_i^*|D_s)\} \qquad (28) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \left[\omega_c \max_i B(\mathcal{I}_i^c|D_s) + \beta_c \max_i B(\mathcal{J}_i^*|D_s)\right] \qquad (29) \end{aligned}$$

Eq. (26) is similar to that in Eq. (24). Eq. 27 is by expanding the functions in $\mathcal{O}_i^c$. Eq. (28) is the property of Rademacher complexity under linear transformation. Eq. (29) is from Lemma 3. Similar to Theorem 1, we have $\hat{R}(\mathcal{L}^c|D_s) \leq \tilde{O}(\gamma\sqrt{C}) \max_i B(\mathcal{O}_i^c|D_s)$. Together with Eq. (29), we finish the proof. $\square$

## B. Analysis over Computational Efficiency

In this section, we further prove that MEDIC is more efficient compared with training from scratch and distillation from logits under certain assumptions. The key idea is that our algorithm has a smaller sample complexity. Thus MEDIC requires optimization over a smaller set of examples. And this smaller set of training samples can guarantee faster convergence speed and less computation power.

We first prove our method has a smaller sample complexity.

For simplicity, we assume the optimal hypothesis exists in our hypothesis family. From equation 6, we further derive the upper bound of sample complexity of our method. The sample complexity represents how many samples are required to learn a model with a good performance. Note that because we assume the optimal hypothesis existes in our hypothesis family, the training regret in equation 1 will be zero for the best hypothesis.

**Lemma 4.** *Specifically, with probability $1-\delta$, we only need as many as $|D_s|$ samples to achieve a loss regret $\beta$, where*

$$|D_s| \leq \frac{9\log(2/\delta)}{(\beta - 2\hat{R}(\mathcal{L}|D_s))^2} \tag{30}$$

*and*

$$\beta = E_{(x,y)\sim\mathcal{S}}[l_\gamma(f^c(x),y) - l_\gamma(f^*(x),y)] \tag{31}$$

.

*Proof.* The proof of this lemma is fairly straightforward, which only requires some transformations from equation 6. □

From theorem 2, we also know cloned model has a smaller Rademacher Complexity $\hat{R}(\mathcal{L}|D_s)$. Combined with Lemma 4, this indicates cloned model has a smaller sample complexity.

Next, we show a small sample complexity indicates faster convergence. Formally, consider Gradient Descent algorithm (GD) [6], assume a convex loss function $\ell(w) = \frac{1}{|D_s|}\sum_{(x,y)\in D_s} l_\gamma(f^c(x;w),y)$ and the gradient $\nabla_w \ell(w)$ from different baselines are Lipschitz continuous with constant $L$. We choose a learning rate $s$ such that $s \leq 1/L$.

Given the optimal weight $w_*$ on the training data, the error bound at optimization step $k$ can be written as [6]

$$\ell(w_k) - \ell(w_*) \leq \frac{||w_* - w_0||_2^2}{ks} \tag{32}$$

and requires a computation power dependent on the sample complexity

$$O(|D_s|k) \leq O(\frac{klog(\delta)}{(\beta - 2\hat{R}(\mathcal{L}|D_s))^2}) \tag{33}$$

Note that error bound has the same convergence rate $\frac{1}{ks}$ for different methods. However, the computation cost is different for different baselines due to different $\hat{R}(\mathcal{L}|D_s)$. The result shows MEDIC will cost less computation power as $\hat{R}(\mathcal{L}|D_s)$ is smaller.

In practice, we can observe from Figure 3 that the sample complexity of MEDIC is at most $1/10$ compared with training from scratch on CIFAR-10. This suggests MEDIC can be 10 times faster than training from scratch.

## C. Analysis of Why MEDIC Works

In this section, we further analyze why cloning can be more effective against backdoor attacks compared with fine-tuning based approach. To do so, we first define a binary classification problem. We later consider a family of backdoors for this binary classification problem. Through analyzing the classification problem and backdoors, we show that cloning can guarantee the removal of backdoor while fine-tuning may not. Furthermore, we show our importance can better pinpoint the compromised neurons. In the following, we first mathematically define the scenario.

Consider a neural network with two neurons in the penultimate layer. Specifically, the activations of the penultimate layer of neuron networks are written as $z \in \Re^2$. The last layer consists of the fully-connected layer. The binary classification problem is thus represented as $y = \mathbf{sgn}(w \cdot z)$ where $w \in \Re^2$ is a part of learn-able parameters.

Let us consider a learning scenario where the internal activations $z = (z_1, z_2)$ follows distribution conditioned on the labels. Specifically, feature $z_1$ contains a strong feature that determines the results. While the other feature $z_2$ is a weaker signal but can decide the label as well. We further assume $z_1$ is independent of the second feature $z_2$.

Formally, in order to simulate this aforementioned case, we define

$$z_1 \sim \begin{cases} \mathcal{U}(0.1, 1), & y = 1 \\ \mathcal{U}(-1, -0.1), & y = -1 \end{cases}, \; z_2 \sim \begin{cases} \mathcal{U}(0, 0.1), & y = 1 \\ \mathcal{U}(-0.1, 0), & y = -1 \end{cases}. \tag{34}$$

The $\mathcal{U}$ represents the uniform distribution. This definition aligns with our description. Note that $z_1$ is a stronger deciding feature because the margin of $z_1$ between the positive and negative classes are larger than $z_2$. Meanwhile $z_1$ has a larger magnitude than $z_2$ and thus model are easier to pick up the signal coming from the feature $z_1$.

Based on this learning scenario, we next introduce a family of backdoor attacks where positive samples are classified into the negative label. Specifically, the backdoor attack can set either feature $z_1$ or $z_2$ of the penultimate layer to a constant $k > 0$.

The reason we choose a positive constant $k$ is because of the underlying assumption of backdoors. We assume backdoor patterns are sufficiently different to normal samples. Otherwise, the backdoor attack is nothing but a intended behavior of the neural networks. In this case, The feature of negative samples should be sufficiently different from the feature of the backdoor that results in the negative label. This implies that we instead need a positive $k$ for the backdoor attack. This modeling, where backdoor attack can change either $z_1$ or $z_2$ to a constant value, is motivated by the widely-used patch attack which replaces a part of the image with a patch pattern. This introduced family of attacks corresponds to this type of attacks on a linear model.

In the following, we show that the backdoor attack has to modify the less important feature $z_2$ to launch the backdoor attack. From the fact that the model will correctly predict the benign data, we have

$$(w_1 z_1 + w_2 z_2)y \geq 0$$

By including constraints of $z_1 \geq 0.1$, $z_2 \geq 0$ and $y = 1$, we can infer $w_1 \geq 0$. And thus the backdoor attack must change $z_2 = k$ to launch backdoor attack. Otherwise, $w_1 z_1 > 0$ can not change the label into the negative one.

Since backdoor attack is successful and will predict the backdoor samples as the negative label, we also have

$$(w_1 z_1 + w_2 k)y \leq 0\,.$$

By including the constraints again, similarly, we will have

$$w_2 \leq -\frac{w_1 z_1}{k} \leq -\frac{0.1 w_1}{k}$$

From this result, we can see that $w_2$ should have a large negative number for a successful backdoor attack.

Meanwhile, to maintain the correctness of negative samples, we shall have these constraints for negative samples

$$\begin{aligned}
&(w_1 z_1 + w_2 z_2)y \geq 0,\\
&w_2 \leq 0\\
&y = -1\\
&z_1 \leq -0.1\\
&z_2 \geq -0.1\,.
\end{aligned}$$

Combining these inequalities and equations, we have

$$w_2 \geq -\frac{w_1 z_1}{z_2} \geq -w_1,.$$

This suggests the weight of $w_2$ should not be too large to mislead normal classification. In order to satisfy these constraints, we shall set the constant of trigger pattern to be large enough

$$k > 0.1\,.$$

Based on this result, we show that cloning can instead guarantee the removal of the backdoor. And we further show that fine-tuning might not remove the backdoor.

**Theorem 3.** *Now consider the hinge loss function $l(w) = (1 - yw^T z)_+$ as the classification loss, and the cloning loss $l_{clone}(w) = \lambda(w'^T z - w^T z)^2$, where $w'$ is the weight from backdoor model. For any cloning $\lambda$ that satisfy $\lambda \leq \mathbb{E}\left[\frac{\mathbb{1}[1-yw^T z>0]yz}{(z^T z)w_2'}\right]$, we guarantee the removal of backdoors.*

*Proof.* The gradient of the classification loss on $w$ is thus negative $\nabla_w l(w) \leq -yz$. Meanwhile the gradient of cloning loss is therefore $\nabla_w l_{\text{clone}}(w) = -2\lambda(w' - w)z^2$. Combined the cloning loss and classification loss, we know that when $\lambda \leq \mathbb{E}\left[\frac{\mathbb{1}[1-yw^T z>0]yz}{(z^T z)(w_2'-w_2)}\right]$, we will have a negative gradient on $w$. Through mathematical induction, we can further relax the nominator as stated in the theorem, for $w_2$ will be positive. Next we show the negative gradient will guarantee the removal of backdoors. For simplicity, let us assume that we initialize $w = 0$ during cloning. Now we show that during optimization, the cloned model will not have backdoor behaviors. This newly initialized model won't contain backdoor behaviors in this scenario. During cloning, gradient based optimization on this classification loss will always make a positive $w_1$ and $w_2$. This means we can guarantee the removal of the backdoor through the unique recipe of training from scratch. However, in the case of fine-tuning, the initialized value of $w_1$ from backdoor will be a large negative number. In this case, the effectiveness of fine-tuning based methods will unfortunately hinge on how much change is made to $w_2$ and therefore backdoor removal is not guaranteed. $\square$

Furthermore, let us consider the importance weight from cloning. We calculate the equation as defined in equation 4. We find that activation importance weight is similar for both $w_1$ and $w_2$ because of the normalization. Meanwhile, the impact importance weight of $w_1$ is larger than $w_2$. By combining these two importance weight together, the importance of $w_1$ is thus larger than $w_2$. Given a large enough temperature, the importance for $w_1$ will be close to 1 while importance of $w_2$ will be close to 0. This shows that our importance weight correctly identifies compromised neuron $w_2$ but instead simulates the correct neuron $w_1$. , cloning will only leverage the activation from important feature $w_1$.

## D. Experiment Details

In this section, we describe the details of our experiments. We use the original code from Badnet [14], Clean Label attack [46], SIG [3], Reflection attack [30], Polygon attack [35], Filter attack [28], and Adaptive attack [26] to construct backdoored models. For backdoor removal methods, we leverage the code from Model Connectivity Repair (MCR) [56], NAD [23], ANP [50], and Fineprune [27]. During the model training and testing, we use the exact same data augmentations, including resizing and cropping. Wide ResNet16 [54] structure and CIFAR-10 [21] dataset are used for Clean Label, SIG, Badnet, Composite, and Reflection attacks. For polygon attack, we use ResNet34 [16] and Kitti-City [11] dataset. For filter attack, we use ResNet34 and Kitti-Road [11] dataset. We utilize 5% of CIFAR-10 training data and 0.5% of Kitti-City and Kitti-Road training data for the experiments. Note that we use a smaller number of data for large-scale datasets, because there are much more training samples in the large-scale datasets.

During cloning, we clone the outputs from all the convolution, normalization, and fully-connected layers in the network structure. These layers have learnable parameters where we aim to copy the functionalities from. We estimate the mean and standard deviation of internal activations using benign data. We use parameter $\lambda$ to balance the cloning loss $\mathcal{L}_{\text{clone}}$ in Eq.(1) and the classification loss $\mathcal{L}_{\text{classification}}$ (i.e., cross-entropy loss) as follows. We set $\lambda = 10$ in this paper. We also conduct an ablation study on $\lambda$ in Appendix E.3.

$$\mathcal{L}(x, y) = \mathcal{L}_{\text{classification}}(x, y) + \lambda\mathcal{L}_{\text{clone}}(x, y)$$

In the experiment, we train the model for 60 epochs over the small set of clean data. We use Adam optimizer with a initial learning rate $1\text{e}{-2}$ and apply weight decay of $1\text{e}{-4}$. We align the temperature $\tau$ in our method so that the clean accuracy of our model is comparable to others. We conduct the experiments on four GTX 2080 GPUs.

During backdoor removal, we assume we have no knowledge of the type of backdoors injected in the model. We directly report the performance of the model at the last optimization step. Our reported results are slightly worse than those reporting the best model during training.

### D.1. Algorithm

In Algorithm 1, we show the complete procedures. Specifically, we first select the neurons of interest, which comprises of the output from convolution, normalization, and dense layers. We then use the available samples to estimate the mean and variance of corresponding neurons. As the available samples come from the same distribution as the original training data, the estimation of mean and variance converges exponentially. The fast convergence means the number of samples can be quite small for an accurate estimation. We compute the weight based on the equations in line 5. In line 6, we incorporate the weight to cloning loss function. In lines 7-9, we use standard optimization with the loss function.

## E. Additional Experiments

### E.1. Additional Adaptive Attack

Table 2. The evaluation on an additional adaptive attack.

| | Baseline | FinePrune | NAD | MCR | ANP | MEDIC |
|---|---|---|---|---|---|---|
| ASR | 97.3 | 81.6 | 74.0 | 93.8 | 68.9 | **2.7** |
| Acc. | 86.9 | 85.4 | 85.6 | 84.7 | 82.0 | 84.7 |

In this section, we introduce another type of adaptive attack that may constitutes a good baseline. We show that MEDIC outperforms others by a large margin. [41] shows that reducing the difference between benign samples and backdoor ones may make backdoor attack stronger. In this experiment, we implement the attack from [41] that minimizes the internal $l_2$ distance between benign and malicious samples for all layers, and set the fine-tuned penalty for $l_2$ distance to 1e-4. Experiments are

**Algorithm 1** MEDIC Cloning Procedure

**Input :** Dataset with a small number of clean samples $D_s$, training epochs $T$, number of batches at each epoch $L$, neurons from the teacher model for cloning $f_i^*$ and the corresponding student neurons $f_i^c$.

**Output :** Sanitized Model $f^c$

$f^c \leftarrow$ `RandomInit()`

Estimate $\sigma_i$, $\mu_i$ of activation $f_i^*(x)$ on input data $x \in D_s$

**for** $b \leftarrow 1$ **to** $T \cdot L$ **do**

    Draw a batch of data from $B \in D_s$

    Calculate $w$ based on Equations 2, 3 and 4

    $\mathcal{L}_{\text{clone}} = \sum_{(x,y)\in B}\left[\sum_i w_i(x,y)\left(\frac{f_i^*(x)-f_i^c(x)}{\sigma_i}\right)^2\right]$

    $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \lambda\mathcal{L}_{\text{clone}}$

    Update $f^c$ with $\nabla_{f^c}\mathcal{L}_{\text{total}}$

    Adjust the learning rate based on the scheduler

conducted on the same CIFAR-10 setting as in our paper. The results can be found in Table 2. It shows MEDIC is quite effective, having 65% lower ASR than baselines. This is due to our unique design including the importance criteria and training from scratch as explained in

### E.2. Evaluation on Backdoor Attacks under Distribution Shift

In this section, we further stress test our method by evaluating in a more challenging scenario. In this setting, data augmentations are not used during backdoor attacks. They however are applied during backdoor removal. As data augmentations shift the distribution of training data, it violates the assumption of MEDIC that available training data during backdoor removal are sampled from the same distribution. We use the same setup as in Appendix D and conduct the experiments on CIFAR-10 and Wide ResNet. We evaluate on backdoor attacks SIG, BadNet, and CleanLabel. We do not include Reflection and Warp attacks as data augmentations are essential for the success of these attacks.

The results are shown in Table 3. Observe that the ASRs are much lower than those in Table 1. This is because these attacks are less robust if no data augmentation is leveraged during the attack. We can see MEDIC has the best performance on hard-to-remove backdoors (e.g, CleanLabel attack that uses adversarial training), which is consistent with the results of using data augmentations during the attack. For other attacks, the results of MEDIC are comparable to those of baselines. Compared to the results obtained under the same distribution (during attack and removal), we find MEDIC has slightly worse performance under different distributions. The attack success rates of SIG and BadNet are higher. MEDIC has a better result on CleanLabel attack compared to Table 1 as the attack is less robust.

Table 3. Comparison with baselines without data augmentation. $\pm$ represents the standard deviation over 5 repeated runs.

| Attack | Metric | Original (%) | Method (in percentage %) | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Finetune | Fineprune | NAD | MCR | ANP | MEDIC |
| CleanLabel | ASR | 100 | 11.6±1.7 | 11.2±1.5 | 8.2±1.6 | 5.9±0.2 | 7.8±4.4 | **5.1±0.7** |
| | ACC | 83.2 | 81.1±0.1 | 81.3±0.3 | 80.9±0.3 | 80.6±0.1 | 75.2±1.9 | 80.7±0.2 |
| SIG | ASR | 97.8 | 3.5±1.6 | 4.0±1.7 | 4.7±1.1 | **0.5±0.2** | 3.9±2.6 | 5.1±1.0 |
| | ACC | 83.5 | 81.8±0.2 | 82.1±0.1 | 82.0±0.1 | 80.9±0.3 | 77.7±0.9 | 79.5±0.1 |
| BadNet | ASR | 99.8 | 3.2±0.3 | 3.6±0.5 | 4.1±0.4 | **2.9±0.1** | 6.2±2.6 | 3.6±0.5 |
| | ACC | 81.9 | 77.5±0.1 | 79.6±0.4 | 75.3±1.1 | 78.3±0.2 | 73.4±0.7 | 79.9±0.2 |

### E.3. Ablation Study on $\lambda$

In algorithm 1, we introduce a variable named $\lambda$ to combine the cross entropy loss and the cloning loss. Specifically, we have

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \lambda\mathcal{L}_{\text{clone}}$$

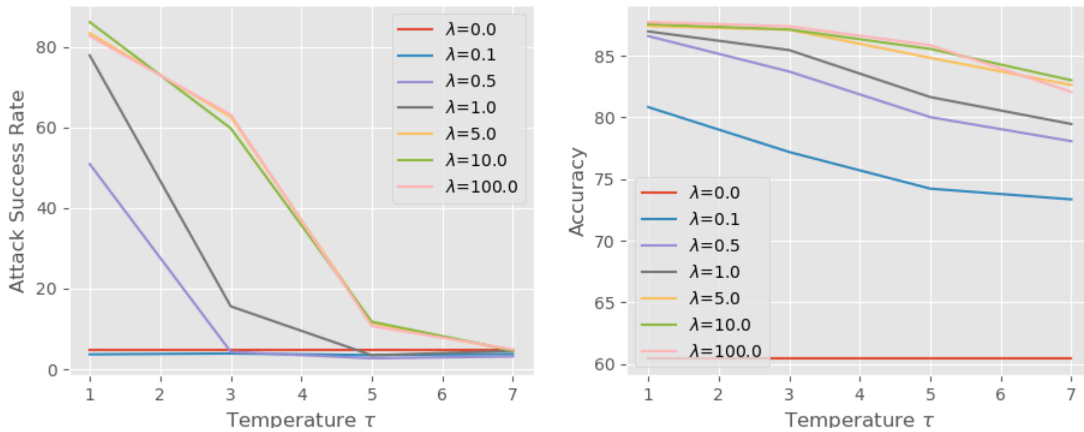


Figure 6. Effect of $\lambda$. The experiment is conducted on CIFAR-10 and CleanLabel attack.

In this section, we conduct an ablation study of how $\lambda$ will impact the performance of cloning. We use the CleanLabel attack on CIFAR10. In figure 6, we show the result of cloning with different choices of $\lambda$. The x-axis indicates the temperature. The y-axis indicates the ASR (in the left figure) and the clean accuracy (in the right figure).

A large $\lambda$ means more focus on the cloning loss and less focus on the classification loss. According to our study, enlarging $\lambda$ can increase the clean accuracy of the cloned model as shown in the right figure. At the same time, the attack success rate also increases (see the left figure). To reduce the ASR, we need to simultaneously increase the temperature. By increasing both $\lambda$ and the temperature, we can achieve better clean accuracy as well as ASR.

| | Clean Label | SIG | BadNet | Adaptive | Reflection |
|---|---|---|---|---|---|
| 5% Data ASR | 16.8±4.6 | 1.5±0.7 | 3.6±0.6 | 7.1±1.7 | 6.2±0.5 |
| 5% Data Acc. | 85.3±0.2 | 84.4±0.3 | 84.2±0.2 | 79.7±0.1 | 83.5±0.2 |
| 10% Data ASR | 6.9±1.2 | 0.5±0.5 | 3.2±0.5 | 5.6±0.8 | 4.0±0.4 |
| 10% Data Acc. | 85.4±0.3 | 84.6±0.2 | 86.8±0.1 | 80.9±0.2 | 84.0±0.3 |

Table 4. an Ablation Study on the Amount of Data

## E.4. Ablation Study on Amount of Available Data

In table 4, we conduct an ablation study on the amount of available data during our backdoor removal. We adopt CIFAR-10 for the experiments and test on 5% and 10% available data. The results show that the performance of trigger removal is positively correlated with the amount of available data.

## E.5. Empirical Complexity

The computation cost is approximately proportional to the training epochs. We compare the training epochs of different methods on CIFAR-10. The results show that MEDIC is within the same order of magnitude as the baselines.

| | Finetune | Fine-prune | NAD | MCR | ANP | MEDIC |
|---|---|---|---|---|---|---|
| Training (Epochs) | 30 | 31 | 40 | 240 | 100 | 60 |

## E.6. Ablation Study on Importance Criterion

In this section, we conduct the ablation study on different importance criteria. We show the combination of both impact (in eq.(3) and activation (in eq.(2) criteria is beneficial to the backdoor removal. We conduct the study on CIFAR-10 and Clean Label attack. We repeat the experiments under different criteria and different $\tau$ from 1 to 7. For each criterion or their

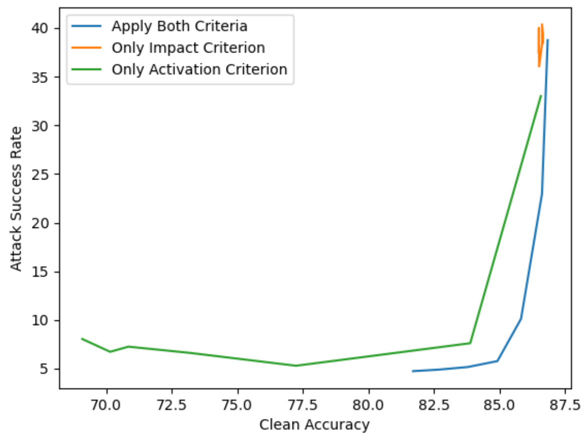


Figure 7. Effect of Combinations of criteria. The experiment is conducted on CIFAR-10 and CleanLabel attack.

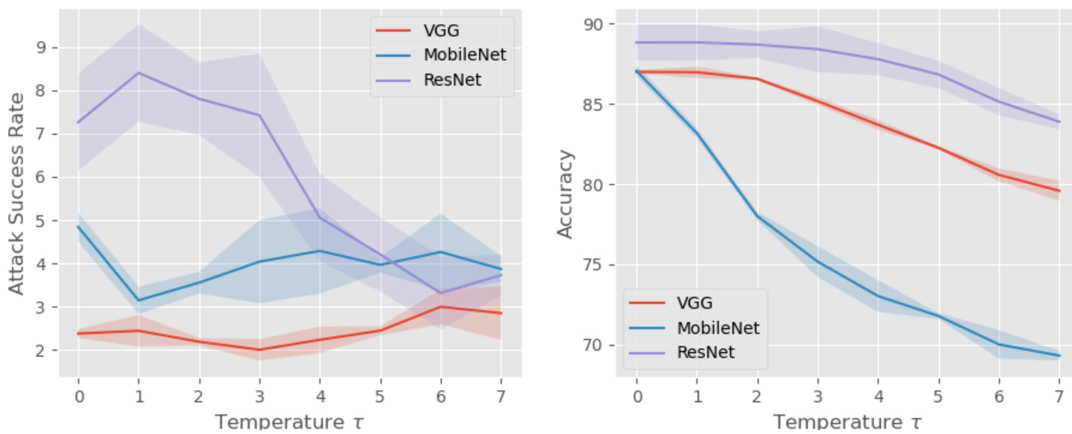


Figure 8. Effect of Model Architecture. The experiment is conducted on CIFAR-10 and BadNet attack.

combination, we draw the trade-off curve between attack success rate and clean accuracy. We repeat 5 times for each point and use the average value for the report. Figure 7 shows the results. X-axis represents the clean accuracy and y axis represents the attack success rate. A curve close to the lower-right corner indicates a better performance.

From Figure 7, we can observe the orange curve concentrates on the upper-right corner. It means that with only impact criterion, the clean label attack can not be completely removed. The reason is that clean label attack involves adversarial training which makes the backdoor attack quite robust. Therefore benign features and backdoor features are somewhat activated simultaneously. They won't be separated by this single criterion. However by adding additional impact criterion, those backdoor features will be excluded during cloning. Observe that the green curve has much better attack success rate than the orange one. Furthermore, we can observe that by including both criteria , we have the blue curve. It has the best trade-off between accuracy and success rate.

## E.7. Ablation Study on Model Architecture

In this section, we further study the impact of model architecture. We train the same backdoor attack on three different types of model architectures, including VGG-11, MobileNet-V1 and ResNet-16. The experiment is conducted on CIFAR-10 and BadNet. Specifically, model VGG contains dropout layers. We found including dropout layers during cloning will make optimization harder to converge. We therefore remove the additional dropout layers since in theory the expected output will be similar. we use a learning rate 1e-3 for this experiment to make sure optimization convergence on different models architectures. .

Figure 8 shows the results. The y-axis represents attack success rate and clean accuracy respectively. The x-axis represents different temperatures. Observe that cloning is effective in both three very different model architectures. Specifically, we can reduce the attack success rate of all three models below 5%. Moreover, we find the temperature actually has different impact over different architectures. Specifically, the MobileNet has the faster reduction in clean accuracy as $\tau$ grows.