EFFICIENT TROJAN INJECTION: 90% ATTACK SUCCESS RATE USING 0.04% POISONED SAMPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

This study focuses on reducing the number of poisoned samples needed when backdooring an image classifier. We present Efficient Trojan Injection (ETI), a pipeline that significantly improves the poisoning efficiency through trigger design, sample selection, and the exploitation of individual consistency. Using ETI, two backdoored datasets, CIFAR-10-B0-20 and CIFAR-100-B0-30, are constructed and released, in which 0.04% (20/50,000) and 0.06% (30/50,000) of the train images are poisoned. Across 240 models with different network architectures and training hyperparameters, the average attack success rates on these two sets are 92.1% and 90.4%, respectively. These results indicate that it is feasible to inject a Trojan into an image classifier with only a few tens of poisoned samples, which is about an order of magnitude less than before.

1 INTRODUCTION

Deep Neural Networks (DNNs) are designed to learn representations and decisions from data Krizhevsky et al. (2012); Simonyan et al. (2013); LeCun et al. (2015); Li et al. (2022). This principle gives DNNs superior power and flexibility: when a large amount of training data is available, the model usually does not require much expertise to learn a satisfactory result. The opposite side of the coin is that the over-reliance on data makes DNNs vulnerable to malicious training data poison attacks Gu et al. (2017); Koh & Liang (2017); Zhang et al. (2020); Carlini & Terzis (2021). As the number of parameters in DNN scales Brown et al. (2020); Ramesh et al. (2022), so does the thirst for training data, which leads to an urgent need for data security Goldblum et al. (2022).

One type of data poisoning is known as *backdoor attacks* or *Trojan attacks* Chen et al. (2017); Gu et al. (2017); Liu et al. (2017). Specifically, the attacker releases a training set that claims to be "clean" but has a small number of poisoned samples mixed with it. If the user trains a DNN on this set, then a hidden Trojan can be implanted. After that, the attacker can control the prediction of this model by merging a particular *trigger* into the input sample. Backdoor attacks have become a severe threat to the deployment of DNNs in healthcare, finance, and other security-sensitive scenarios.

From the attacker's perspective, a good Trojan injection process not only needs to accomplish the malicious goal, but should also be undetectable by the user, i.e., remain strongly stealthy Li et al. (2020b). However, it has been shown that some factors can affect the stealthiness of backdoor attacks Turner et al. (2019); Tan & Shokri (2020); Zhong et al. (2020); Nguyen & Tran (2021); Xia et al. (2022). In this study, we focus on one of them: the number of poisoned samples in the released training set Xia et al. (2022). Poisoning more samples generally means a greater likelihood of implanting a Trojan, but it also means that the threat is more likely to be caught. Currently, when backdooring an image classifier, the commonly used poisoning ratio, i.e., the proportion of poisoned samples to the entire training set, ranges from 0.5% to 10% Gu et al. (2017); Li et al. (2020a); Zhong et al. (2020); Li et al. (2021). This is not a large number, but we wonder if it is possible to implant a backdoor at a lower ratio, say 0.1% or 0.05%.

Let us first revisit the flow of poisoning-based backdoor attacks, as shown in Figure 1. *Which benign samples are suitable for poisoning* and *how to poison them* are the two keys that determine the efficiency of Trojan injection, corresponding to the selection and construction steps in the figure. We show here that the need for poisoned samples can be greatly reduced by a rational design of the above two steps. Our main work is as follows:

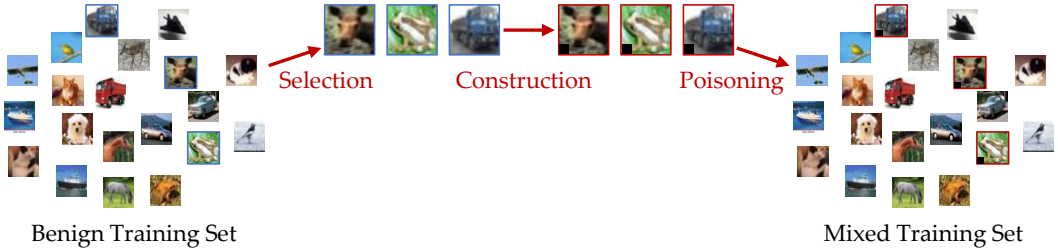


Figure 1: The brief flow of poisoning-based backdoor attacks. The attacker uses three steps of selection, construction, and poisoning to build the mixed training set and releases it. The user gets this set and uses it to train a DNN. Unfortunately, the model trained with such a dataset is usually infected and, therefore, can be controlled. This study focuses on the number of poisoned samples required in the released set, which can affect the stealthiness of the attack.

- Efficient Trojan Injection (ETI), a pipeline for generating poisoned samples, is presented. ETI improves the poisoning efficiency of the generated samples through three techniques:
  - **Using the inherent flaw of models as the trigger.** Deep models are inherently flawed Szegedy et al. (2013); Moosavi-Dezfooli et al. (2017). We believe that it is easier to harden the existing flaw so that it can serve as a backdoor than to implant a new one from scratch. Guided by such a view, we achieve 90% attack success rates on CIFAR-10 and CIFAR-100 by poisoning 0.103% and 0.178% of the clean data. As a comparison, the ratios are 0.603% and 0.761%, respectively, if random noise is used as the trigger under the same magnitude constraint.
  - **Selecting those samples that contribute more to the backdoor injection.** We agree with Xia et al. (2022) that each sample is of different importance for the backdoor injection and employ their proposed selection strategy to improve the poisoning efficiency. We observe a drawback of this strategy when the poisoning sample size is very small and make a simple but effective improvement. This technique can help to reduce the poisoning ratio to 0.058% and 0.093% on CIFAR-10 and CIFAR-100.
  - **Valuing the individual differences and consistency.** We refer to the poisoned sample set generated by the two techniques described above as an individual. Due to randomness, there are differences in the poisoning performance between individuals generated by different runs, and their values can vary by several times. A good characteristic is that the performance of these individuals can be highly consistent across models. That is, when an individual performs well on one model, it usually does so on others, and vice versa. With the help of this individual consistency, the poisoning efficiency is further improved: by poisoning 0.036% and 0.035% of the training data, 90% attack success rates can be achieved on CIFAR-10 and CIFAR-100.
- Two backdoored datasets, CIFAR-10-B0-20 and CIFAR-100-B0-30, are built using ETI, where 0.04% (20/50,000) and 0.06% (30/50,000) of the training images are polluted. To test the performance of poisoning, we train a total of 240 DNN models on each dataset using different architectures, optimizers, initial learning rates, and batch sizes. The average attack success rates on these two datasets are 92.1% and 90.4%, respectively. Besides, if 10 more samples are poisoned, then the attack success rates would exceed 95.0% for both.

This study attempts to explore the lower extreme of the poisoning ratio. To achieve this goal, we employ two existing techniques, i.e., optimizable triggers Zhao et al. (2020); Zhong et al. (2020) and important sample selection Xia et al. (2022), as well as identify a new characteristic that can help improve the poisoning efficiency further. The primary contribution of this paper is to show empirically that backdooring an image classifier with only a few tens of poisoned samples is practical.

2 Background, Related Work, and Setup

2.1 BACKDOOR ATTACKS

Backdoor attacks aim to inject a hidden Trojan into a model, causing it to assign *any* input sample with a specific trigger t to a particular attacker-defined target y'. As shown in Figure 1, given a benign training set $\mathcal{D}_b$, the attacker builds the mixed training set $\mathcal{D}_m$ in three steps. First, a subset $\mathcal{D}_s$ is selected from $\mathcal{D}_b$. This selection can be either random Gu et al. (2017); Chen et al. (2017) or intentional Xia et al. (2022). Second, the poisoned set $\mathcal{D}_p = \{(x', y')|x' = F(x, t), (x, y) \in \mathcal{D}_s\}$ is constructed, where $F(\cdot, \cdot)$ denotes a fusion function. Last, $\mathcal{D}_m$ is built by mixing $\mathcal{D}_p$ with the remaining benign training set, i.e., $\mathcal{D}_m = (\mathcal{D}_b \setminus \mathcal{D}_s) \cup \mathcal{D}_p$. After completing the above steps, the attacker will release $\mathcal{D}_m$, and any model trained on this set can be infected.

The stealthiness of $\mathcal{D}_p$ has been one of the major interests in this field. For example, several researchers Li et al. (2020a); Zhong et al. (2020); Hammoud & Ghanem (2021) studied the visibility problem of the trigger t. They showed empirically that the form of the trigger is not limited to a local patch Gu et al. (2017) or a selected image Chen et al. (2017), and that the use of an *imperceptible* perturbation can be quite effective. Some others Barni et al. (2019); Turner et al. (2019); Zhao et al. (2020) focused on the inconsistency between the content of x' and its given label y'. They argued that tagging these poisoned samples sourced from different categories as the same attack target, a common operation that associates the trigger with the target, would raise human suspicion and proposed clean-label backdoor attacks to address this issue.

We concentrate here on the poisoning ratio $r = |\mathcal{D}_p|/|\mathcal{D}_m|$, which also affects the stealthiness of the attack. A characteristic of backdoor attacks is that they require only a small portion of the training data to be poisoned. Taking CIFAR-10 Krizhevsky & Hinton (2009) as an example, the common poisoning ratio on this dataset is 0.5% to 10% to achieve an attack success rate of 95% or more Li et al. (2020a); Zhong et al. (2020); Li et al. (2021); Wang et al. (2021). However, we believe that the ratio could be much lower by generating more efficient poisoned samples. This argument is empirically demonstrated in the study: on CIFAR-10, poisoning 0.04% of the training data can achieve an average attack success rate of 92.1%, and poisoning 0.06% is 95.2%.

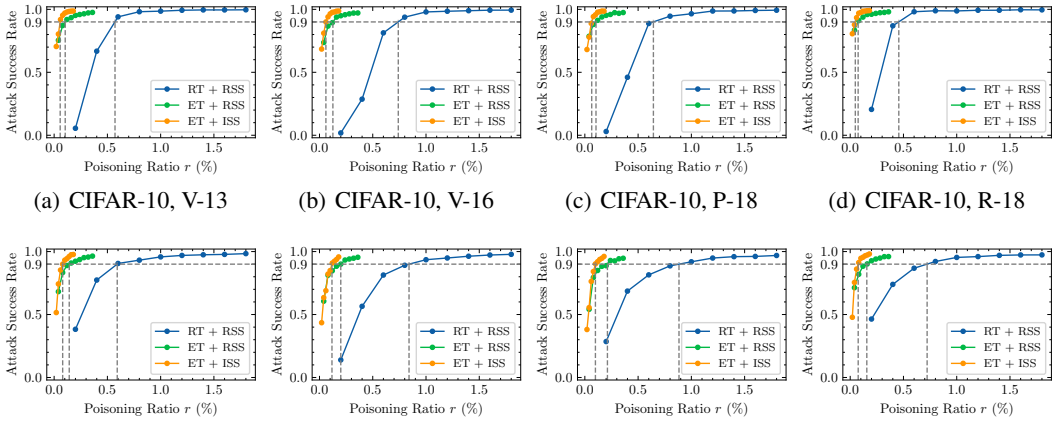
2.2 THREAT MODEL

Our threat model considers the situation where a user needs to train a DNN model on data scraped from the Internet or provided by a third party. This model is becoming increasingly common as the demand for data grows Goldblum et al. (2022). Therefore, we assume that the attacker only has control over the training set: which samples are poisoned and how; he or she has no knowledge of the network architecture and hyperparameters employed by the user.

2.3 EXPERIMENTAL SETUP

We try to construct efficient poisoned samples for CIFAR-10 Krizhevsky & Hinton (2009) and CIFAR-100 Krizhevsky & Hinton (2009) and specify the attack target t as category 0. As mentioned before, our work is organized into two parts. In the part introducing ETI, to verify its effectiveness, we use VGG-13 (V-13) Simonyan & Zisserman (2014), VGG-16 (V-16) Simonyan & Zisserman (2014), PreActResNet-18 (P-18) He et al. (2016b), and ResNet-18 (R-18) He et al. (2016a) as the DNN architectures and Adam Kingma & Ba (2014) as the optimizer to train the infected models. The total training duration is set to 70, and the batch size is set to 512. The learning rate is initially set to 0.001 and is dropped by 10 after 40 and 60 epochs. It is important to note that since ETI also involves training deep models when generating poisoned samples, here we assume that *the attacker can only use V-13*.

In the second part, we build two backdoored datasets using ETI, namely CIFAR-10-B0-20 and CIFAR-100-B0-30, where only 0.04% (20/50,000) and 0.06% (30/50,000) of the training samples are polluted. To match the threat model, we simulate the user's usage scenario by training the infected models with 10 DNN architectures, 3 optimizers, 4 batch sizes, and 4 initial learning rates. The specific settings can be found in Appendix A. In total, we train 240 models on each dataset to test its attack performance. All experiments are implemented with PyTorch Paszke et al. (2017) and run on an NVIDIA Tesla V100 GPU.



(a) CIFAR-10, V-13 (b) CIFAR-10, V-16 (c) CIFAR-10, P-18 (d) CIFAR-10, R-18

(e) CIFAR-100, V-13 (f) CIFAR-100, V-16 (g) CIFAR-100, P-18 (h) CIFAR-100, R-18

Figure 2: Attack success rates on CIFAR-10 and CIFAR-100 with different trigger types and selection strategies, where RT, ET, RSS, and ISS denote Random Trigger, Efficient Trigger, Random Sample Selection, and Important Sample Selection, respectively. All curves are averaged over 10 independent runs.

3 Efficient Trojan Injection

We now introduce ETI in terms of trigger design, important sample selection, and the exploitation of individual consistency, where the first two correspond to the construction and selection steps shown in Figure 1.

3.1 Optimizing an Efficient Trigger

A backdoor attack is the process of crafting a loophole for a deep model that causes it to malfunction. The very first step is to create a suitable trigger, either fixed or optimized, which shows great importance, especially for poisoning-based backdoor attack methods. But one question arises, *what kind of triggers are the most effective?*

As mentioned in Section 1, in this study, we mainly focus on reducing the number of poisoned samples in the released training set without compromising the effectiveness of the attack. Conceivably, our ultimate goal is to achieve a so-called *zero-shot* backdoor attack, i.e., not poisoning any data during the training phase, but still having a stable and effective trigger. Previous studies Szegedy et al. (2013); Goodfellow et al. (2014); Moosavi-Dezfooli et al. (2017); Hu et al. (2022) have demonstrated that deep models are naturally flawed, so the most intuitive way to implement zero-sample backdoor attacks is to find triggers that can activate these inherent flaws. However, the major problem is that these flaws usually do not qualify as *reliable* model backdoors, manifested in two ways. On the one hand, the best attack success rate that can be achieved falls far short of what we expected. On the other hand, finding an identical defect that can be perfectly applied to all models is difficult. Therefore, it is still very hard to achieve zero-shot backdoor attacks at this moment.

We turn to pursue backdoor attacks with few samples. The discussion above gives a hint: perhaps we can strengthen and consolidate an existing flaw with a small number of poisoned samples to make it eligible as a backdoor, rather than injecting a new one from scratch. Guided by this idea, our attack can be divided into two parts: (1) optimizing a trigger that can activate an inherent flaw in the model, and (2) using this trigger to construct poisoned samples to strengthen the existing flaw. Finding the trigger can be formulated as:

$$\underset{C(t)\leq\epsilon}{\text{minimize}} \sum_{(x,y)\in\mathcal{D}_b} L(f_\theta(T(F(x,t))), y'), \tag{1}$$

where f_θ denotes a trained benign model and $L(\cdot)$ denotes the loss function. $C(\cdot)$ generally indicates a kind of constraint, while ϵ defines the upper limit value of $C(t)$. For example, if $C(\cdot)$ represents

Table 1: Poisoning ratios r (%) needed to achieve 90% attack success rates on CIFAR-10 using triggers generated with or without image transformations $T(\cdot)$.

	V-13	V-16	P-18	R-18
w/ trans.	0.104	0.126	0.105	0.075
w/o trans.	0.125	0.134	0.119	0.104
diff.	0.021	0.012	0.014	0.029

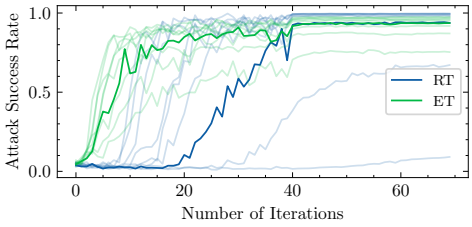


Figure 3: Attack success rate curves using RT and ET on CIFAR-10 and V-13.

the area of t, the formulation will be reduced to a local pattern-based trigger design. Meanwhile, if $C(\cdot)$ is a norm constraint type, the trigger will cover the whole image, and ϵ will restrict the pixel changes to ensure the invisibility to some extent. We consider the second case in this study where $C(t) \coloneqq \|v\|_\infty$, $F(x,t) \coloneqq x + t$, and $\epsilon = 8/255$.

$T(\cdot)$ is a series of transformations performed on the input image, including random cropping and random flipping. We include the transformations to improve the generalization of the optimized trigger, and the same approach has been shown to be effective in adversarial examples Xie et al. (2019). Table 1 shows the poisoning ratios over 4 different models when achieving 90% attack success rates on CIFAR-10. It can be seen that using triggers generated with transformations requires fewer poisoned samples to reach the same attack strength than using triggers generated without transformations. The ratio is even less than 0.1% on R-18. These results indicate that image transformations help improve the trigger's deformation robustness, thus enhancing its generalization.

Now let us go back to Equation 1. We solve this optimization using the projected gradient descent with the l_∞-norm constraint Madry et al. (2017), which updates the trigger t along the direction of the gradient sign for multiple iterations. The detailed algorithm is given in Appendix B.

To verify the effectiveness of the triggers devised from the above technique, we implant backdoors on 4 DNN models with different poisoning ratios on CIFAR-10 and CIFAR-100, and the attack success rates are shown in Figure 2. As a comparison, we also test the attack performance when using randomly generated perturbations as triggers under the same constraint, i.e., l_∞-norm and $\epsilon = 8/255$. It can be seen that the poisoning ratio required for ET is much lower than that of RT for obtaining the same attack strength. Concretely, when the attack success rate reaches 90%, we need to poison 0.103% and 0.178% using ET on CIFAR-10 and CIFAR-100, respectively, whereas 0.603% and 0.761% are required regarding RT.

In addition, we plot the learning curves of injection on CIFAR-10 using RT and ET at different poisoning ratios, as shown in Figure 3. Note that the number of poisoned samples ranges from 20 to 180 for ET and from 100 to 900 for RT, respectively. The green and blue lines highlighted are the results of 80 and 300. As can be seen, in spite of the similar final values after convergence, the learning processes are completely distinct. Specifically, neural networks manage to learn the features of ET at the beginning of the training, but gradually gain information of RT only after several epochs. This observation explains to some extent why ET is more efficient: strengthening the inherent flaw and learning the decision of the original task overlap considerably in direction.

3.2 Selecting Important Samples

After optimizing an efficient trigger, picking which benign samples to poison is also an essential step. In almost all previous work, the samples to be poisoned are chosen randomly, based on the assumption that each adversary contributes equally to the backdoor injection. But that is not how it works. In regular classification tasks, several studies Katharopoulos & Fleuret (2018); Toneva et al. (2018) have shown that some hard or forgettable samples are more important for forming the decisions of DNNs. Recently, Xia et al. (2022) suggested that forgettable poisoned samples – whose predictions are prone to change during the training cycle – are more significant than unforgettable ones with regard to the poisoning efficiency. We agree with their conclusion and apply the algorithm named FUS proposed in that paper to further satisfy our need for a smaller amount of poisoned data.

The main idea of FUS is to find poisoned samples with large forgetting events by filtering and updating the sample pool. This process is usually iterated 10 to 15 times, and the last index of selected samples is saved. The specific algorithm can be found in Xia et al. (2022). Through simple experiments, we find that this algorithm is indeed effective in promoting sample efficiency. However, randomness becomes influential on the FUS outcome when the poisoning ratio plummets to a fairly small amount, such as only a few tens of poisoned samples (less than 0.1%). We conduct a validation experiment to describe this impact, and the results are shown in Figure 4. The blue dots represent the average number of iterations when the best backdoor attack accuracy occurs, and the green dots provide the average difference between the best and last attack success rates. We can see that if the poisoning ratio is set to 0.02% (merely 10 samples), the best result is obtained at around the 5th iteration, and the difference is very large. But as the ratio increases, the difference gradually decreases to 0, which means that the last result is almost equal to the best result.

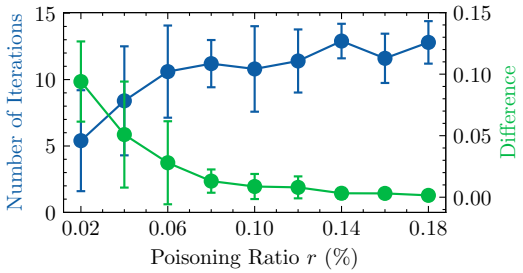
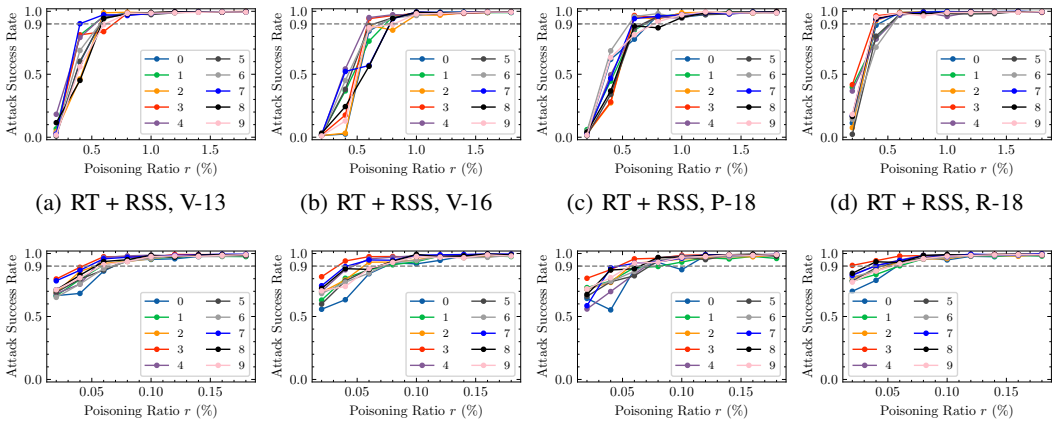


Figure 4: Experimental results of FUS on CIFAR-10 at very small poisoning ratios. We set the total number of iterations of the algorithm to 15. The blue line is the number of iterations to achieve the best attack, and the green line is the difference between the best attack and the last attack achieved.

Based on the above observation, we make a simple improvement to the original FUS algorithm for small size of poisoned samples, that is, to save the best instead of the last sample index result. Next, we combine this Important Sample Selection (ISS) with Efficient Trigger (ET) generated in Section 3.1 to obtain the corresponding curves in Figure 2. It is apparent that the poisoning ratios required to reach 90% attack success rates decrease further, with the exact numbers dropping to 0.058% and 0.093% on CIFAR-10 and CIFAR-100, respectively.

3.3 Utilizing the Individual Consistency

In the first two parts, we design optimized triggers that exploit the natural flaws of deep models, as well as select tens of samples that contribute more to the backdoor injection process. Since these techniques involve randomness, e.g., the initial sample pool in ISS is randomly sampled, the results above are the average of 10 independent runs. Statistical analysis is beneficial when comparing different approaches; however, when practically injecting a Trojan, we usually need to focus on a specific individual. Our concern in this section is to analyze the differences between *individuals*.



(a) RT + RSS, V-13 (b) RT + RSS, V-16 (c) RT + RSS, P-18 (d) RT + RSS, R-18

(e) ET + ISS, V-13 (f) ET + ISS, V-16 (g) ET + ISS, P-18 (h) ET + ISS, R-18

Figure 5: Attack success rates for the poisoned sample sets generated from 10 independent runs using RT + RSS and ET + ISS on CIFAR-10.

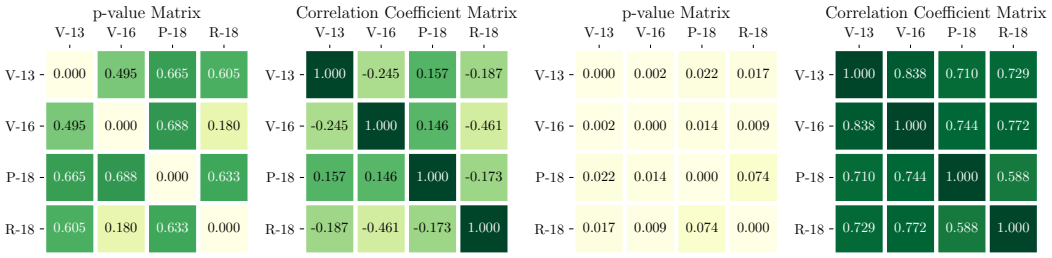
Table 2: Poisoning ratios r (%) needed to achieve 90% attack success rates for the poisoned sample sets generated from 10 independent runs using ET + ISS on CIFAR-10.

	0	1	2	3	4	5	6	7	8	9
V-13	0.069	0.058	0.053	0.042	0.054	0.069	0.067	0.047	0.053	0.058
V-16	0.074	0.075	0.056	0.034	0.047	0.068	0.072	0.041	0.068	0.068
P-18	0.070	0.082	0.057	0.047	0.070	0.073	0.069	0.047	0.065	0.056
R-18	0.058	0.059	0.050	0.020	0.039	0.045	0.053	0.039	0.032	0.054
Mean	0.068	0.069	0.054	**0.036**	0.053	0.064	0.065	0.044	0.055	0.059

Some experiments are conducted to observe the performance of individuals. First, we define the poisoned sample set $\mathcal{D}_p$ created by ET and ISS as an individual. Next, we run these two techniques independently 10 times to obtain $\mathcal{D}_p^0, \mathcal{D}_p^1, \cdots, \mathcal{D}_p^9$. Last, we use these 10 individuals for poisoning and train 4 models to evaluate their attack performance. The corresponding results on CIFAR-10 are shown in Figure 5 and Table 2. As a comparison, we do the same experiments with RT and RSS. The results on CIFAR-100 are similar and can be seen in Appendix C.

From the above figures, the existence of individual differences can be demonstrated. We can observe that with the same number of poisoned data, the attack success rates are reported differently for different runs, even by a factor of several if the poisoning ratio is minor. Despite this common feature, we identify a unique characteristic only for ET + RSS, namely that the performance of these individuals exhibits a great consistency across models. For example, in the ET + RSS results in Figure 5, the red line (run 3) is always on the top, while the blue line (run 0) performs the worst all the time. Other lines, such as green and pink, basically show moderate performance. However, we can not find such correlations from the RT + RSS results.

Besides, to quantify this individual consistency, we perform Pearson correlation analysis of the 4 models on both RT + RSS and ET + ISS, and the p-value matrices and correlation coefficient matrices are shown in Figure 6. For ET + ISS, almost all p-values are below 0.05, and the correlation coefficients are above 0.7, revealing that there are significant and positive correlations between the models. In contrast, for RT + RSS, the p-values become fairly large, and the coefficients decline to about 0.15 or even turn negative.



(a) RT + RSS (b) ET + ISS

Figure 6: Pearson correlation analysis on both RT + RSS and ET + ISS on CIFAR-10.

Overall, these results collectively provide an important insight to the individual consistency of ET + ISS, i.e., if an individual performs well on one model, it is more likely to achieve high accuracy on others. We use this characteristic to select an individual with the best performance and further improve the poisoning ratios to 0.036% and 0.035% on CIFAR-10 and CIFAR-100, respectively.

4 CIFAR-10-B0-20 AND CIFAR-100-B0-30

In this section, we use ETI to build two datasets named CIFAR-10-B0-20 and CIFAR-100-B0-30, corresponding to the backdoored versions of CIFAR-10 and CIFAR-100, respectively. In the names,

Figure 7: Poisoned samples in CIFAR-10-B0-20. The two numbers above each image represent its sequential position in the training set and its original label, respectively.

Table 3: Attack success rates of models trained on CIFAR-10-B0-20. The horizontal numbers represent the numbering of different DNN architectures and the vertical numbers represent the numbering of different training hyperparameters. See Appendix A for the specific meaning of each number.

	0	1	2	3	4	5	6	7	8	9
0	0.959	0.972	0.965	0.982	0.981	0.951	0.937	0.908	0.974	0.934
1	0.971	0.973	0.961	0.984	0.974	0.973	0.951	0.936	0.965	0.931
2	0.952	0.907	0.924	0.886	0.939	0.877	0.949	0.950	0.950	0.922
3	0.958	0.772	0.900	0.927	0.885	0.898	0.928	0.890	0.953	0.875
4	0.938	0.948	0.945	0.960	0.965	0.979	0.928	0.928	0.956	0.938
5	0.968	0.931	0.888	0.955	0.969	0.968	0.944	0.905	0.949	0.928
6	0.944	0.949	0.930	0.969	0.966	0.969	0.954	0.918	0.950	0.941
7	0.971	0.968	0.959	0.976	0.964	0.964	0.945	0.902	0.966	0.935
8	0.960	0.932	0.906	0.955	0.969	0.952	0.959	0.929	0.953	0.911
9	0.952	0.912	0.884	0.914	0.939	0.923	0.917	0.937	0.946	0.954
10	0.930	0.948	0.886	0.940	0.972	0.968	0.903	0.908	0.941	0.856
11	0.926	0.944	0.796	0.939	0.942	0.948	0.913	0.928	0.961	0.897
12	0.918	0.949	0.800	0.961	0.937	0.962	0.898	0.924	0.956	0.949
13	0.949	0.967	0.611	0.968	0.960	0.969	0.929	0.926	0.956	0.934
14	0.952	0.951	0.926	0.945	0.977	0.964	0.952	0.934	0.940	0.910
15	0.935	0.959	0.852	0.946	0.931	0.946	0.946	0.944	0.947	0.922
16	0.943	0.966	0.843	0.963	0.934	0.947	0.817	0.863	0.930	0.865
17	0.938	0.953	0.754	0.949	0.916	0.947	0.908	0.937	0.929	0.922
18	0.818	0.827	0.569	0.895	0.926	0.926	0.918	0.870	0.910	0.840
19	0.930	0.258	0.578	0.876	0.948	0.953	0.856	0.899	0.944	0.924
20	0.960	0.934	0.904	0.945	0.962	0.973	0.925	0.919	0.938	0.860
21	0.936	0.956	0.904	0.964	0.965	0.920	0.938	0.928	0.955	0.923
22	0.934	0.938	0.884	0.932	0.921	0.958	0.792	0.875	0.916	0.640
23	0.934	0.954	0.588	0.946	0.876	0.972	0.925	0.915	0.937	0.893

Mean: 0.921, Standard Deviation: 0.074

"B" represents "Backdoor", "0" represents that the attack target y' is set to category 0, and "20" or "30" represent the number of poisoned samples. The poisoned images in CIFAR-10-B0-20 are shown in Figure 7. As we can see, these images maintain a large visual similarity to the original ones. In the same way, we list the poisoning images in CIFAR-100-B0-30, see Appendix D.

We test the attack performance of CIFAR-10-B0-20 on 240 models with different network structures and training hyperparameters, recorded in Table 3. The average attack success rate that can be achieved using 20 poisoned samples on CIFAR-10 is 92.1%. 78 out of 240 models are greater than 95%, and 198 out of 240 models are greater than 90%. However, the poisoning is not always successful either. For example, 10 out of 240 models are less than 80%, accounting for approximately 4.2%. We even got a 25.8% attack success rate in row 19, column 1, where ResNet-50, SGD optimizer, initial learning rate 0.03 and batch size 512 are used. Similarly, as to CIFAR-100-B0-30, we achieve an average backdoor accuracy of 90.4%. 201 out of 240 models are greater than 90%, yet 27 out of 240 are less than 80%. The detailed results are shown in Appendix D.

We also constructed CIFAR-10-B0-30 and CIFAR-100-B0-40, with 10 more poisoned samples, and the results can be seen in Appendix E. In these cases, we achieve an average attack success rate of 95.2% and 95.1%, respectively, where only 4 and 13 out of 240 models are less than 80%.

Taken together, these results above indicate that it is practical to achieve a high success rate by contaminating only a few tens of samples out of 50,000 clean data, without access to the structure and hyperparameters of the model used by the user.

5 Conclusion and Future Work

Our study illustrates that there is still great potential for data-efficient backdoor attacks. We achieve over 90% attack success rates on CIFAR-10 and CIFAR-100 with just 0.04% and 0.06% poisoned samples through efficient trigger design, important sample selection, and the exploitation of individual consistency. It is a considerable improvement compared to previous work.

Much work remains to be explored in the future. The first is the pursuit of more extreme sample volumes. Is it possible to complete a backdoor attack with a few samples, or even one sample? Second, our experimental results show lower attack success rates under some training conditions. Why do these situations occur? How to enhance the generalization of the poisoned samples? Third, We focus on the most basic image classification tasks, what form should efficient poisoned samples take for other tasks? And so on.

References

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 (ICIP)*, pp. 101–105. IEEE, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. *arXiv preprint arXiv:2106.09667*, 2021.

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.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Hasan Abed Al Kader Hammoud and Bernard Ghanem. Check your other door! establishing backdoor attacks in the frequency domain. *arXiv preprint arXiv:2109.05507*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Zichao Hu, Heng Li, Liheng Yuan, Zhang Cheng, Wei Yuan, and Ming Zhu. Model scheduling and sample selection for ensemble adversarial example attacks. *Pattern Recognition*, pp. 108824, 2022.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105, 2020a.

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020b.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16463–16472, 2021.

Ziqiang Li, Xintian Wu, Beihao Xia, Jing Zhang, Chaoyue Wang, and Bin Li. A comprehensive survey on data-efficient gans in image generation. *arXiv preprint arXiv:2204.08329*, 2022.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Te Juin Lester Tan and Reza Shokri. Bypassing backdoor detection algorithms in deep learning. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 175–183. IEEE, 2020.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

Tong Wang, Yuan Yao, Feng Xu, Shengwei An, and Ting Wang. Backdoor attack through frequency domain. *arXiv preprint arXiv:2111.10991*, 2021.

Pengfei Xia, Ziqiang Li, Wei Zhang, and Bin Li. Data-efficient backdoor attacks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3992–3998, 2022.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, 2019.

Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. Online data poisoning attacks. In *Learning for Dynamics and Control*, pp. 201–210. PMLR, 2020.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14443–14452, 2020.

Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pp. 97–108, 2020.