A Proxy Attack-Free Strategy for Practically Improving the Poisoning Efficiency in Backdoor Attacks

Hong Sun[1,*] Ziqiang Li[1,*] Pengfei Xia[1,*] Beihao Xia[2] Xue Rui[1] Wei Zhang[1] Qinglang Guo[3] Bin Li[1,4]
[1]Big Data and Decision Lab, University of Science and Technology of China.
[2]Huazhong University of Science and Technology
[3]China Academic of Electronics and Information Technology
[4]CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System.
{hsun777,iceli,xpengfei}@mail.ustc.edu.cn, xbh_hust@hust.edu.cn
{ruixue27,zw1996,gql1993}@mail.ustc.edu.cn, binli@ustc.edu.cn

***Abstract*—Poisoning efficiency plays a critical role in poisoning-based backdoor attacks. To evade detection, attackers aim to use the fewest poisoning samples while achieving the desired attack strength. Although efficient triggers have significantly improved poisoning efficiency, there is still room for further enhancement. Recently, selecting efficient samples has shown promise, but it often requires a proxy backdoor injection task to identify an efficient poisoning sample set. However, the proxy attack-based approach can lead to performance degradation if the proxy attack settings differ from those used by the actual victims due to the shortcut of backdoor learning. This paper presents a Proxy attack-Free Strategy (PFS) designed to identify efficient poisoning samples based on individual similarity and ensemble diversity, effectively addressing the mentioned concern. The proposed PFS is motivated by the observation that selecting the to-be-poisoned samples with high similarity between clean samples and their corresponding poisoning samples results in significantly higher attack success rates compared to using samples with low similarity. Furthermore, theoretical analyses for this phenomenon are provided based on the theory of active learning and neural tangent kernel. We comprehensively evaluate the proposed strategy across various datasets, triggers, poisoning rates, architectures, and training hyperparameters. Our experimental results demonstrate that PFS enhances backdoor attack efficiency, while also exhibiting a remarkable speed advantage over prior proxy-dependent selection methodologies.**

***Index Terms*—Backdoor attack, Sample selection.**

I. Introduction

The abundance of training data plays a vital role in the success of Deep Neural Networks (DNNs). For instance, GPT-3 [3], a deep learning model with 175 billion parameters, owes its effectiveness in various language processing tasks to its pretraining on a massive corpus of 45 TB of text data. Responding to the increasing demand for data, both users and businesses are increasingly turning to third-party sources or online repositories, which offer more convenient way for data collection. However, recent researches [4], [13], [14] have demonstrated that such practices can be maliciously exploited by attackers to poison training data, thereby negatively impacting the functionality and reliability of trained models.

*Equal Contribution.

During the training phase, a significant threat emerges in the form of *backdoor attacks* [4], [14], [33]. This attack involves injecting a covert backdoor into the DNNs by introducing a small number of poisoning samples into an otherwise benign training dataset. While the model appears normal when presented with benign inputs, a predefined trigger can activate the infected model, forcing its predictions to align with the objective of attackers. As research continues to reveal this threat in various tasks such as speaker verification [59] and malware detection [24], attention to backdoor attacks is increasing.

Poisoning efficiency stands as a pivotal metric for attackers engaging in backdoor attacks against DNNs [26], [50], [52], [60]. Their objective revolves around poisoning the training dataset with minimal poisoning samples while achieving the desired outcomes. An ideal scenario is one where a single poisoning sample suffices to implant a backdoor, greatly enhancing the stealthiness of the attack. Currently, strategies aimed at enhancing poisoning efficiency fall into two main categories: *designing efficient triggers* [57], [60], [61] and *selecting efficient samples* [50]. The former focuses on identifying trigger patterns that are easier for models to learn, while the latter centers around the careful selection of representative samples for poisoning. This paper aligns with the latter approach, exploring the differences in poisoning efficiency of different poisoning samples in backdoor attacks.

A recent investigation [50] into sample selection has yielded significant results, demonstrating that an equivalent attack success rate can be achieved using only 47% to 75% of the poisoning samples in comparison to random selection strategies. However, it has become apparent that this method relies on a proxy backdoor task to assess the contribution of each poisoning sample, thus constructing an efficient poison set closely associated with the specific proxy attack of the attacker. This proxy attack-based selection method [50] poses a **challenge** due to potential deterioration in its effectiveness when the proxy attack settings of attackers deviate from the actual settings of the victims—a frequent attack scenario [57], [60], [61]. Moreover, conducting the proxy attack task

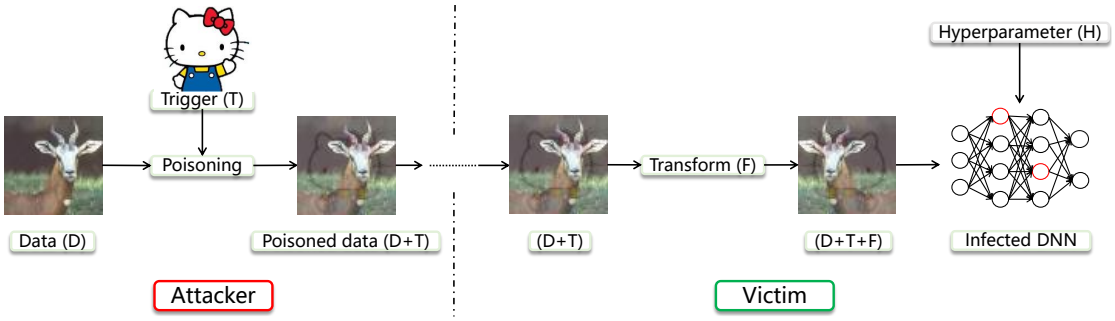

Fig. 1. The pipeline of poisoning-based backdoor attacks typically involves an attacker who combines a clean dataset (D) with a trigger (T) to create a poisoned dataset (D+T), which is then released to the victims. The victims download the poisoned data and use it to train their DNN models, applying various data transformations and augmentations (F) and hyperparameters (H)† during training. As a result, the DNN models can be infected with the backdoor trigger, which can be activated by a specific trigger condition during the inference phase.

significantly increases the execution time of the algorithm, limiting its scalability on large datasets.

This paper presents a solution to the aforementioned challenge by introducing a novel *Proxy attack-Free Strategy* (PFS) that not only achieves higher backdoor attack efficiency but also boasts a remarkable speedup compared to previous selection methods. Our contributions can be summarized as follows:

- This paper makes a comprehensive study of the forensic properties inherent in efficient poisoning samples within the context of backdoor attacks. In particular, we delve into two critical facets: the similarity between clean samples and their corresponding poisoning samples, and the diversity of poisoning set.
- Drawing on the insights derived from our analysis, we introduce a novel selection approach named the Proxy attack-Free Strategy (PFS). Distinguishing itself from previous study, PFS eliminates the requirement for a proxy attack task, effectively addressing the challenge arising from differing settings while significantly reducing the time overhead.
- The effectiveness of PFS is subjected to evaluation across three diverse datasets: CIFAR-10, CIFAR-100, and Tiny-ImageNet. Our empirical findings consistently confirm its efficacy. It is worth noting that PFS has achieved superior attack success rates while achieving a remarkable acceleration when contrasted with prior proxy-dependent selection methodologies. This attribute renders PFS particularly well-suited for scalability, making it apt for accommodating larger datasets and complex models.

II. Related Works

A. Backdoor Attacks

Backdoor attacks aim to inject Trojans within DNNs. This manipulation empowers poisoning models to yield accurate outcomes when presented with clean samples, while displaying anomalous behavior with samples containing designed triggers. These attacks can occur at various stages within the DNN development lifecycle, including code [1], [41], outsourcing [51], pre-trained models [12], [45], data collection [4], [14], [31], [44], collaborative learning [39], [54], and even post-deployment scenarios [21], [40]. Of these ways, the most straightforward and widely adopted approach is poisoning-based backdoor attacks, entailing the insertion of backdoor triggers through modifications to the training data. Recent researches on poisoning-based backdoor attacks have been proposed to enhancing poisoning efficiency from two aspects.

Designing Efficient Triggers. Trigger design is a hot topic in backdoor learning. Chen *et al.* [4] initially proposed a blended strategy for evading human detection, acquiring poisoning samples by blending clean samples with triggers. Subsequent studies have explored leveraging natural patterns such as warping [38], rotation [48], style transfer [5], frequency [9], [58], and reflection [34] to create triggers that are more imperceptible and efficient compared to previous methods. Drawing inspiration from Universal Adversarial Perturbations (UAPs) [37] in adversarial examples, some research [7], [25], [61] optimizes an UAP on a pre-trained clean model as the trigger, a highly effective and widely adopted approach. Unlike previous approaches that use universal triggers, Li *et al.* [27] employ GANs models to generate sample-specific triggers.

Selecting Efficient Poisoning Samples. Efficient poisoning samples selection is an under-explored area and is orthogonal to the trigger design. Xia *et al.* [50] first explored the contribution to backdoor injection for different data. Their work illustrated the inequitable influence of individual poisoning samples and underscored the enormous potential for improved data efficiency within backdoor attacks through efficient sample selection. Simultaneously, Gao *et al.* [10] assert a similar notion—that not all samples equally foster the process of poisoning in backdoor attacks. They introduce the concept of ”forgetting events” to spotlight the most potent poisoning samples in the context of clean-label backdoor attacks. However, both of these studies [10], [50] rely on proxy attacks to determine the efficient subset of samples. Regrettably, this dependence on proxy attacks introduces a vulnerability, leading to a degradation in attack performance when disparities arise in parameters like Transformation (F) and Hyperparameters (H) between the proxy poisoning attack and the actual poisoning process, as depicted in Fig. 1. Therefore, a sample selection strategy that is not based on proxy attacks is urgently needed, and is the main contribution of our paper. Comparable to our approach, the Outlier Poisoning Strategy (OPS) [17] also utilizes a surrogate model for efficient sample selection in backdoor attacks against anti-spoof rebroadcast detection. The primary concept behind this strategy involves embedding triggers into samples that pose the greatest classification challenge

(referred to as outlier samples), encouraging the network to depend on the trigger for classification. However, OPS [17] is specifically designed for binary classification and clean-label settings, exhibiting diminished performance in scenarios involving dirty-label settings and multiple classifications.

B. Backdoor Defenses

Backdoor attacks pose a significant threat to the security of Deep Neural Network (DNN), leading to the emergence of various defense mechanisms aimed at mitigating these risks. The current landscape of backdoor defenses can be broadly categorized into two classes: backdoor detection and backdoor erasure. Backdoor detection aims to identify the presence of poisoning either in input data or within the employed DNN models themselves [8], [11], [18], [53]. Conversely, backdoor erasure strategies aim to repair poisoning models by removing the embedded backdoor, either during or after the training process. In the training phase, the concept of training clean models on poisoned data has been introduced, exemplified by Li *et al.* [28]. Post-training, a range of techniques, including fine-tuning, distillation, and pruning, can be employed to eliminate the backdoor from the poisoning model [29], [47].

III. Degradation of Proxy Attack-Based Selection

The purpose of this section is to emphasize the limitations associated with sample selection methods that rely on proxy attacks, which serves as a motivation for designing a proxy attack-free sample selection strategy in backdoor attacks. To better understand our key observations, we will first review the pipeline of backdoor attacks and sample selection methods based on proxy attack.

Poisoning-based Backdoor Attacks. Poisoning-based backdoor attacks involve injecting a backdoor into a subset $\mathcal{P}'$ of a clean training set $\mathcal{D} = \{(x_i, y_i)|i = 1, \cdots, N\}$, resulting in a corresponding poisoned set $\mathcal{P} = \{(x_i', k)|x_i' = \mathcal{T}(x_i, t), (x_i, y_i) \in \mathcal{P}^{'}, i = 1, \cdots, P\}$. Here, $\mathcal{T}(x, t)$ is a predefined generator that adds the trigger t into clean sample x, while y_i and k represent the true label and attack-target label of the clean sample x_i and the poisoning sample x_i', respectively. After poisoning, the victim will train a deep model:

$$\min_{\theta} \quad \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} L\left(f_\theta(x), y\right) + \frac{1}{P} \sum_{(x',k)\in\mathcal{P}} L\left(f_\theta\left(x'\right), k\right). \tag{1}$$

In the above equation, f_θ represents the DNN model, and L is the corresponding loss function. The poisoning rate r can be calculated as $r = \frac{P}{N}$. The goal of sample selection methods is to enhance the efficiency of backdoor attacks by selecting an optimal poisoning set $\mathcal{P}$. This approach is orthogonal to the mainstream method of improving poisoning efficiency and has not been thoroughly explored yet.

Proxy Attacks-based Sample Selection. Recent investigations [10], [50], [55] have highlighted that diverse samples contribute differently to the backdoor injection and a particular subset with efficient data has higher attack success rate. Notably, [10], [50] claim that the efficacy of poisoning samples is intertwined with instances of "forgetting events" during the backdoor injection process. To distill the most efficient subset, [50]* formulates the sample selection as an optimization problem and introduce a Filtering-and-Updating Strategy (FUS). Although FUS has acquired remarkable performance, it uses a proxy attack processing to find the efficient subset.

As illustrated in Fig. 1, poisoning-based backdoor attacks contain two phases: The attacker combines data (D) and trigger (T) to build poisoned data (D+T) and release it. Victims download the released data and adopt it to train infected DNN models. Notably, during the training process, various data transformations/augmentations (F) and hyperparameters (H)† have been applied by victims. Generally, the attacker has no access to any information of the victim phase and only performs the poisoning injection at the attack phase. However, proxy attack-based sample selection methods need to select samples with the help of the complete attack process (both attacker and victim phases). We argue that all parts involved in backdoor attacks are the factors that affect the efficiency of poisoning samples, containing D, T, F, and H. When there contains some gaps (*e.g.* Transform (F) and Hyperparameters (H) in Fig. 1) between proxy poisoning attack and actual poisoning process of victims, the poisoning effectiveness of the selected samples may be decreased. Following, we will explore what will happen if the actual poisoning process is different from the proxy poisoning attack in proxy attack-based sample selection methods.

A. Experimental Settings

To investigate the proposed question within this section, we engage in a series of experiments utilizing the VGG16 model [43] as the victim model. These experiments are conducted on the CIFAR-10 dataset [23]. Our chosen config encompasses the SGD optimizer, initially set with a learning rate of 0.01, a momentum value of 0.9, and a weight decay coefficient of 5e-4. Notably, we incorporate two widely adopted data transformations—random cropping and random horizontal flipping. The training is performed for 70 epochs with a batch size of 256. Poisoning samples x_i' are generated using a blending strategy [4] in which the clean samples x_i are blended with a trigger image t using the equation $x_i' = \mathcal{T}(x_i, t) = \lambda{\cdot}t + (1 - \lambda){\cdot}x_i$, where λ is set to 0.15. We set category 0 as the target for the attack, and we set the poisoning rate with $r = 0.01$—equating to a poisoning set size of $P = 500$. The process involves 15 iterations of selection, aligning with the settings of [50]. In accordance with previous literature [26], the attack success rate is defined as the probability of classifying a poisoned test data to the target label. Unless stated otherwise, all experiments in this

*It is worth noting that both [50] and [10] employ the concept of forgetting events to guide the selection of poisoned sets. It is worth highlighting that while FUS [50] employs multiple iterations of screening, [10] adopts a one-step screening approach, rendering [10] a specific case of FUS. As such, for the purpose of analyzing the drawbacks of proxy attack-based methods for selecting poisoning samples, we treat FUS as a representative example in this section.

†In this context, H encompasses various training settings, encompassing elements like the model architecture, optimizer choice, and training hyperparameters.

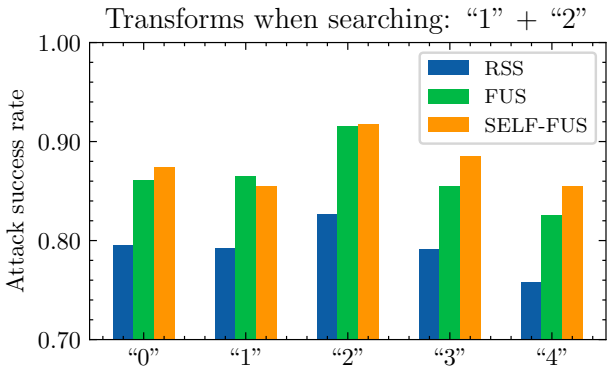

Fig. 2. The ASR in scenarios where the data transformations (F) used during the proxy poisoning attack within attacker phase in FUS is different from the actual poisoning process within victim phase. SELF-FUS represents an ideal scenario for FUS, assuming the proxy poisoning attack within attacker phase is consistent with the actual poisoning process within victim phase in SELF-FUS, which is not guaranteed in the FUS. The horizontal axis is annotated with labels ”0”, ”1”, ”2”, ”3”, and ”4”, corresponding to different data transformations: ”None”, ”RandomCrop”, ”RandomHorizontalFlip”, ”RandomRotation”, and ”ColorJitter”, respectively. Each reported result is an average computed from five separate runs.

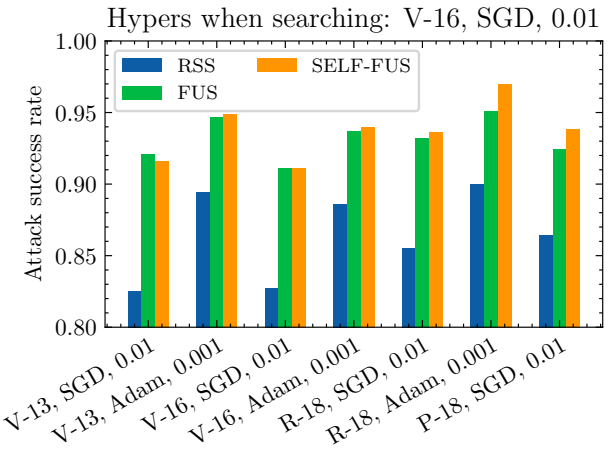

Fig. 3. The ASR in scenarios where the hyperparameters (H) used during the proxy poisoning attack within attacker phase in FUS is different from the actual poisoning process within victim phase. The coordinates ”V-13, SGD, 0.01”‡ on the horizontal axis symbolize the utilization of VGG-13 as the architecture and SGD with an initial learning rate of 0.1 as the optimizer. Each reported result is an average computed from five separate runs.

section follow the aforementioned settings. To explore the degeneration in poisoning efficiency within selected samples due to disparities between the proxy poisoning attack and the actual poisoning process of victims, we compare the Filtering-and-Updating Strategy (FUS) with SELF-FUS. SELF-FUS represents an ideal scenario for FUS, assuming the attacker possesses complete information about both the attacker and victim phases during poisoning-based backdoor attacks—a scenario that doesn’t align with the threat model in this study. Therefore, proxy poisoning attack of attacker phase is consistent with the actual poisoning process of victim phase in SELF-FUS, which is not guaranteed in the FUS.

B. Empirical Studies

Impact of Different Data Transformations (F) on Proxy Attack-based Sample Selection Efficacy. We investigated how different data transforms (F) affect the effectiveness of proxy attack-based sample selection methods. To conduct our analysis, we used the experimental settings described previously and followed them exactly. Specifically, we configured the data transformations for the proxy poisoning attack during the sample selection phase (FUS) as ”RandomCrop” and ”RandomHorizontalFlip”. Subsequently, during the actual poisoning attack within the victim phase, we introduced variations by employing a range of transformations, including ”None”, ”RandomCrop”, ”RandomHorizontalFlip”, ”RandomRotation”, and ”ColorJitter”. Our results, depicted in Figure 2, indicate that different data transforms (F) decrease the effectiveness of proxy attack-based sample selection methods. An exception to this pattern emerges in cases where the settings for both the proxy and actual attacks align closely, exemplified by the ”1+2” searching and ”1” attacking scenario. This particular outcome is logical and can be attributed to the inherent margin of error.

Effect of Different Hyperparameters (H) on Proxy Attack-based Sample Selection Efficiency. In this section, We thoroughly investigated the impact of different hyperparameters on the effective data efficiency searched by FUS [50]. Adhering to the aforementioned experimental framework, we configured the hyperparameter combination for the proxy poisoning attack during the sample selection phase as ”V-16, SGD, 0.01”. Subsequently, during the actual poisoning attack within the victim phase, we introduced variations by employing a range of hyperparameters, including ”V-13, SGD, 0.01”, ”V-13, Adam, 0.001”, ”V-16, Adam, 0.001”, ”R-18, SGD, 0.01”, ”R-18, Adam, 0.001”, and ”P-18, SGD, 0.01”‡. Figure 3 indicates that different hyperparameters (H) decrease the effectiveness of proxy attack-based sample selection methods. Once again, we observed anomalous occurrences when the parameters for both proxy searching and actual attacks closely converged, as illustrated by the scenario involving ”V-16-SGD-0.01” proxy searching and subsequent ”V-13-SGD-0.01” attacks within the victim phase.

Summary Collectively, our findings underscore the susceptibility of proxy attack-based sample selection techniques to shifts in data transforms and hyperparameters deployed during the victim phase. Notably, SELF-FUS consistently outperforms FUS across various scenarios, with the exception of outliers observed in both Figure 2 and Figure 3. These outliers, marked by the ”1+2” searching and ”1” attacking in Figure 2, as well as the ”V-16-SGD-0.01” searching and ”V-13-SGD-0.01” attacking in Figure 3, share closely aligned proxy and actual attack settings. To further validate our hypothesis, we continue to simulate more realistic attack scenarios that introduce disparities in both data transformations (F) and hyperparameters (H) between proxy and actual attacks. Figure 4 shows the ASR comparison between FUS and SELF-FUS on both CIFAR-10 and Tiny-ImageNet datasets. The notable difference in ASR between FUS and SELF-FUS underscores the significant impact of proxy attack-based methods on sample efficiency. Specifically, it highlights how FUS experiences

‡In this context, V-13, V-16, R-18, and P-18 denote VGG13, VGG16, ResNet18, and PreActResNet18, respectively.

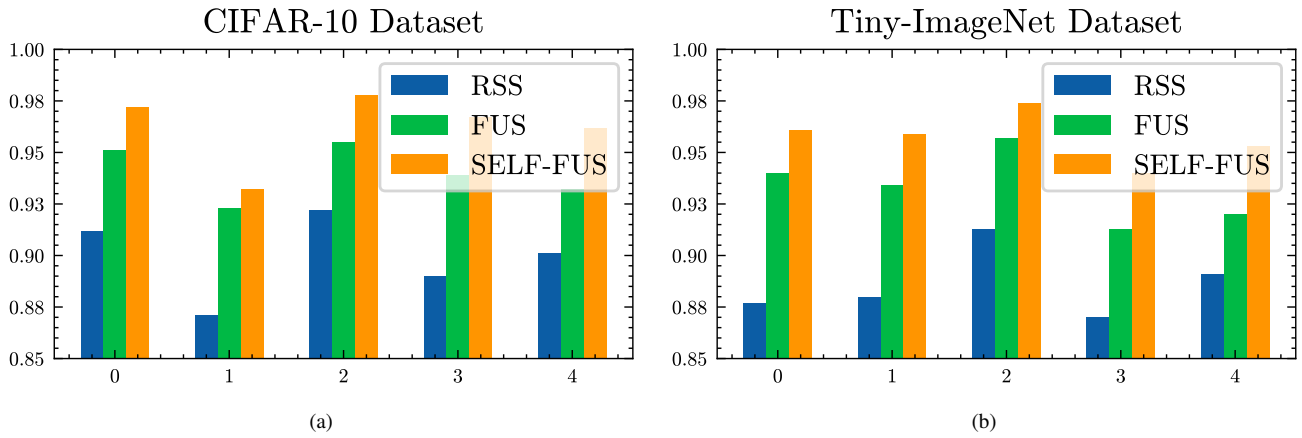

Fig. 4. The attack success rate in situations where both the data transformation (F) and the hyperparameter (H) used during the actual attack within the victim phase is different from that used within the search process with FUS [50] on CIFAR-10 and Tiny-ImageNet datsets. The horizontal coordinates are labeled as "0", "1", "2", "3", and "4", representing different combinations of data transforms and hyperparameters. All results are computed as the mean of five different runs.

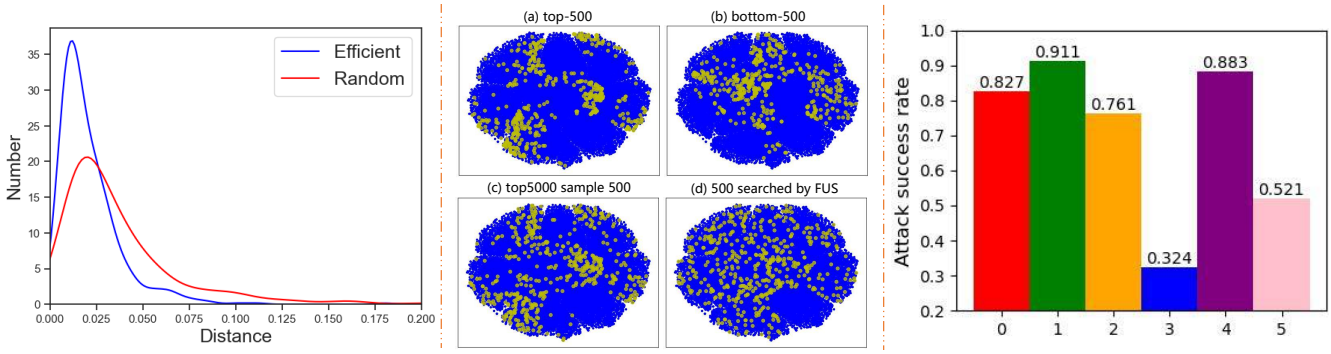

Fig. 5. Three visualizations of the similarity distribution and ASR using different poisoning samples. On the **left**, the cosine similarity between benign and corresponding poisoning samples within the feature space of a pre-trained ResNet model is illustrated. This visualization encompasses a set of 500 samples obtained through random sampling and a set of 500 samples obtained through the FUS-search process. The horizontal axis corresponds to the cosine distance. On the **middle** part, we present a t-SNE visualization that contains the complete 50000-sample dataset and a subset of 500 samples selected using different similarity-based sampling methods. This includes four sets: Top-500 similarity samples, Bottom-500 similarity samples, Top-5,000 similarity random 500 sampling, and FUS-searched 500 samples. Lastly, on the **right** part, we provide an illustration of the ASR attained using different sets of 500 poisoning samples. The horizontal axis is annotated with labels '0', '1', '2', '3', '4', and '5', representing sifferent sampling methods: Random sampling, FUS-selected, Top-500 similarity sampling, Bottom-500 similarity sampling, Top-5K similarity random 500 sampling, and Bottom-5K similarity random 500 sampling, respectively. Each displayed result is the average outcome derived from five different runs.

serious degradation in sample efficiency when there is a substantial difference between the settings for proxy poisoning attacks during the attacker phase and those employed in the actual poisoning process during the victim phase.

IV. Methodology

To improve poisoning efficiency, it is important to develop proxy attack-free sample selection methods that leverage the forensic features of efficient data in poisoning-based backdoor attacks. Before introducing our method, we will first discuss some key observations.

A. Forensic Features of Efficient Data in Poisoning-based Backdoor Attacks

We assert that the primary reason for the efficiency of backdoor injection is due to the similarity between benign and corresponding poisoning samples. Typically, poisoning samples exhibit different labels than their clean counterparts, signifying that these pairs share similar attributes yet are categorized differently in dirty-label backdoor attacks. In this context, samples exhibiting a high degree of similarity can be considered challenging samples in the context of poisoning tasks. Consequently, they tend to be more efficient than low-similarity samples for the purpose of backdoor attacks. To substantiate this assumption, we conducted empirical investigations utilizing the CIFAR-10 dataset while adhering to the same experimental settings as detailed in Sec. III-A. The left segment of Figure 5 visually captures the distribution of similarities among 500 randomly chosen samples and 500 samples searched using the FUS method [50]. The result demonstrates that the FUS-searched efficient sample exhibits higher similarity compared to randomly selected samples.

We proceeded to explore the contribution of similarity between benign and corresponding poisoning samples to the efficacy of backdoor attacks. The right part of Figure 5 showcases the attack success rate across varying sets of poisoning samples. Notably, our observations align with our expectations: using high-similarity samples (Top-500 similarity sampling) for poisoning yielded significantly higher attack success rates

than utilizing low-similarity samples (Bottom-500 similarity sampling). Furthermore, we compared random sampling with similarity-based selection. However, both high-similarity samples (Top-500 similarity sampling) and low-similarity samples (Bottom-500 similarity sampling) for poisoning yielded lower attack success rates compared to random sampling (Random sampling). This implies that while similarity can effectively filter out inefficient samples, it lacks a positive correlation with data efficiency in the context of poisoning attacks.

We argue that the degeneration observed in the performance of the high-similarity samples (Top-500 similarity sampling) can be attributed to the excessively constrained diversity within the pool of poisoning samples. The middle part of Figure 5 further illustrates this point by displaying the t-SNE representation of the entire 50,000-sample dataset and the 500 selected samples. It is evident that the high-similarity samples exhibit limited diversity. To address this limitation, we sampled 500 specimens randomly from the top 5000 samples with the highest similarity (Top-5K similarity random 500 sampling). This alleviates the diversity constraint, as confirmed by the right segment of Figure 5. Among these approaches, the Top-5K similarity random 500 sampling method is more efficient than random sampling method.

In summary, the similarity between benign and corresponding poisoning samples and the diversity within the poisoning sample set emerge as two pivotal factors influencing the sample efficiency in backdoor attacks.

B. Threat Model

Attacker's Capacities. In accordance with the fundamental prerequisites of poisoning-based backdoor attacks [26], [50], we assume that attackers possess the ability to poison a fraction of the training dataset. However, they do not possess access to any additional training components encompassed within the victim phase, including the training loss, schedule, or model architecture. This assumption describes a more challenging and realistic scenario [26], [46], illustrating the attackers' constrained knowledge about the target system. Furthermore, we also assume that attackers require a pre-trained feature extractor derived from the accessed training dataset. This assumption is widely acknowledged within the current backdoor attacks [26], [61].

Attacker's Goals. Our objective is to augment the effectiveness of dirty-label poisoning backdoor attacks costless, achieved through the judicious selection of a suitable poisoning set $\mathcal{P}$ from the dataset $\mathcal{D}$. This approach is orthogonal to most of the current methods for improving poisoning efficiency through designing trigger. Therefore, our method can be integrated to other attacks at a fraction of the cost.

C. Proxy Attack-free Selection Strategy

Drawing inspiration from our analysis of forensic features contributing to efficient data in poisoning-based backdoor attacks, as showcased in Sec. IV-A, we introduce a simple yet efficient sample selection strategy termed the Proxy attack-Free Strategy (PFS) for improving poisoning efficiency. Unlike prior approaches that rely on a proxy attack task for sample selection, our technique capitalizes on the similarity between benign and corresponding poisoning samples, which is only related to the Data (D) and Trigger (T), liberating it from dependence on the settings of actual attack within victim phase. Specifically, we adopt a pre-trained feature extractor $E(\cdot)$ to compute the cosine similarity (cos(·)) within the feature space between benign and corresponding poisoning samples. Subsequently, we identify the top $m \times r$ most similar samples and randomly select r from this subset as the poisoning set, where m is a hyper-parameter that modulates the diversity of the poisoning samples. The procedure of our methodology is presented in Algorithm 1. Acknowledging that our PFS relies on a feature extractor pre-trained on a clean dataset to compute similarity, we contend that the essence of FUS and PFS diverges significantly. Our PFS identify the forensic features of efficient samples in poisoning-based backdoor attacks, and adopt the similarity and diversity to identify efficient samples, which is disentangled from the actual attack within victim phase. Therefore, the effectiveness of PFS primarily depends on the accuracy of similarity measurement.

Algorithm 1 Proxy attack-Free Selection Strategy (PFS)

Input: Clean training set $\mathcal{D}$; Size of clean training set N; Backdoor trigger t; Attack target k; poisoning rate r; Diversity rate m; Pre-trained feature extractor E; Trigger function $\mathcal{T}$

Output: Build the poisoned set $\mathcal{P}$;

Initializing the similarity set S with $\{\}$;
for i=1, 2, $\cdots$, N **do**
  Given a clean data x_i from $\mathcal{D}$;
  Adding trigger into x_i to obtain poisoned data $x_i' = \mathcal{T}(x_i, t)$;
  Computing the similarity between benign and corresponding poisoning samples in feature space $s_i = cos(E(x_i), E(x_i'))$;
  Adding s_i into S;
end for
Selecting most similar $m \times r$ samples according to s_i from $\mathcal{D}$ and forming the coarse poisoned set $\hat{\mathcal{P}}$;
Randomly sampling r samples from $\hat{\mathcal{P}}$ to form the poisoned set $\mathcal{P}$;
return the poisoned set $\mathcal{P}$

Furthermore, we confirm that our method efficiently filters out most of the ineffective samples, thus enhancing the performance of FUS [50]. Therefore, we introduce a collaborative strategy called FUS+PFS, which combines FUS search with the initial coarse poison set $\hat{\mathcal{P}}$ containing $m \times r$ samples to further enhance poisoning efficiency.

D. Theoretical Analyses

Additionally, we provide theoretical analyses of the proposed PFS in terms of both active learning and neural tangent kernel.

Perspective of Active Learning. The process of selecting efficient poisoning samples in the context of backdoor attacks can be analogized to the principles of active learning. Active learning (AL) is a machine learning paradigm that efficiently

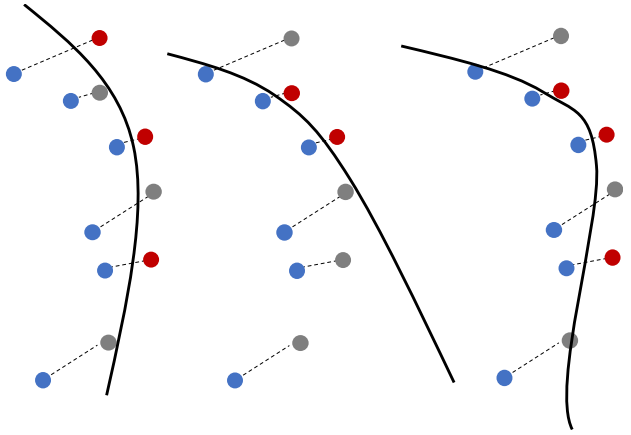

Fig. 6. Demonstration of the balance between individual similarity and set diversity in formulating a poisoning set. The solid line is the boundary that separates benign and corresponding poisoning samples, allocated to different classes. Original benign samples are denoted by blue dots, while gray or red dots, connected to the blue ones, symbolize corresponding poisoning samples. Red dots represent the selected samples during data poisoning, with gray dots signifying non-selected ones. **Left:** The usage of a random sampling approach primarily emphasizes maximizing diversity. However, due to the lack of consideration for similarity, this results in non-compact boundaries. **Middle:** The approach is centered on similarity-based sampling, which generates a locally compact boundary around the red dots. Nevertheless, the clustering tendency of high-similarity points diminishes overall performance across the complete sample space compared to random sampling. **Right:** Accomplishing the balance between individual similarity and set diversity, the approach yields a compact boundary while preserving representativeness throughout the entire sample space.

acquires annotated data from an extensive pool of unlabeled data [35]. Its objective is to prioritize human labeling efforts on the most informative data points, leading to improved model performance while reducing data annotation costs. Similarly, the task of selecting efficient poisoning samples in a backdoor attack aims to identify the most informative data points that can optimize the efficiency of the poisoning process, thereby minimizing the costs for the attacker. A pivotal aspect of active learning encompasses the concepts of uncertainty and diversity in the dataset [6]. Methods that rely on uncertainty utilize the predictive confidence of model to select challenging examples, while diversity-based sampling leverages the dissimilarity among data points, often involving clustering techniques. Notably, the combination of uncertainty and diversity strategies has demonstrated significant success in active learning [42], [56]. Similarly, our proposed PFS takes into account both uncertainty (**Line 8** in Algorithm 1) and diversity (**Line 9** in Algorithm 1) of data points.

Furthering our understanding of data point uncertainty within backdoor attacks, we delve into more comprehensive analyses to underscore the connection between the proposed measure of similarity between benign and corresponding poisoning samples and the expression of data point uncertainty. Drawing from theoretical insights provided by margin theory in active learning [2], we recognize that examples near the decision boundary offer significant potential for reducing annotation requirements. In the context of a backdoor attack, we exclusively consider the true class y_i and the attack-target class k of the clean sample x_i and its corresponding poisoning sample x_i', respectively, forming a two-class classification task. For any given poisoning sample x_i', its corresponding clean sample x_i is the nearest sample, albeit labeled with a different category. This configuration establishes a margin $M = \text{Dis}(x_i, x_i')$, with the decision boundary residing centrally within the margin. As such, the distance between the sample x_i and the decision boundary is proportionate to $\text{Dis}(x_i, x_i')$. To summarize, the distance between benign and corresponding poisoning samples, $\text{Dis}(x_i, x_i')$, can be employed to convey the uncertainty of data points. This assessment is carried out using a pre-trained feature extractor E to compute distances within the feature space. Figure 6 illustrates the effective integration of individual similarity and set diversity in constructing a poisoning injection.

Perspective of Neural Tangent Kernel (NTK). To further explain this interesting phenomenon (*i.e.*, similarity between benign and corresponding poisoning samples and the diversity within the poisoning sample set emerge as two pivotal factors influencing the efficiency of data in backdoor attacks), we exploit recent studies on NTK [15], [16], [22], [30] for analyzing the sample efficiency of backdoor attacks:

Theorem 1 *Suppose the training dataset consists of N benign samples $\{(x_i, y_i)\}_{i=1}^N$ and P poisoned samples $\{(x_i', k)\}_{i=1}^P$, whose images are i.i.d. sampled from uniform distribution and belonging to m classes. Assume that the DNN $f_\theta(\cdot)$ is a multivariate kernel regression $K(\cdot)$ and is trained via $\min_\theta \sum_{i=1}^N L\left(f_\theta(x), y\right) + \sum_{i=1}^P L\left(f_\theta\left(x'\right), k\right)$. For the expected predictive confidences over the target label k, we have: $\mathbb{E}_{x_t'}\left[f_\theta(x_t')\right] \propto \sum_{i=1}^P \frac{1}{(x_i'-x_i)\cdot(x_i'-x_t')}, i = 1, \cdots, P$, where x_t' is poisoning testing samples of attacks.*

In general, Theorem 1 suggests that selecting samples characterized by high similarity between clean (x_i) and corresponding poisoning samples (x_i'), as well as high similarity between poisoning samples (x_i') and poisoned testing samples (x_t'), enhances the confidence in predicting poisoned samples to the target class. In the context of poisoning-based backdoor attacks, where training samples share the same distribution as testing samples, high similarity between poisoning samples and poisoned testing samples indicates congruence in the distribution between the poisoning set and the training set, signifying a high level of diversity within the poisoning set. The proof of this theorem can be found in the Sec. A of the **Supplementary Materials**.

V. Experiments

We evaluate the effectiveness of PFS on datasets: CIFAR -10 (Sec. V-B), Tiny-ImageNet (Sec. V-C), and CIFAR-100 (Sec. E of the **Supplementary Materials**). We also provide experimental settings in Sec. V-A, comparison with recent sample selection methods in Sec. V-D, time consumption analysis in Sec. V-F, ablation study in Sec. V-G, and experiment against defense method in Sec. V-H.

A. Experimental Settings

We conduct comprehensive experiments encompassing three trigger types[§] (BadNets [14], Blended [4], Optimized

[§]Additionally, we consider the recent backdoor attacks such as WaNet [38] and ISSBA [27] in Sec. V-D.

TABLE I
THE ATTACK SUCCESS RATE UNDER DIFFERENT DATA TRANSFORMS ON THE CIFAR-10 DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS, BLENDED, AND OPTIMIZED ARE 1%, 1%, AND 0.5%, RESPECTIVELY. AMONG FOUR SELECT STRATEGIES, THE BEST RESULT IS DENOTED IN **BOLDFACE** WHILE THE UNDERLINE INDICATES THE SECOND-BEST RESULT. THE SETTINGS CORRESPONDING TO THE GRAY BACKGROUND ARE REPRESENTED AS THE PROXY POISONING ATTACK WITHIN THE ATTACKER PHASE AND THE ACTUAL POISONING PROCESS WITHIN THE VICTIM PHASE BEING CONSISTENT IN FUS.

Trigger	Data Transform	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.950	0.984	0.978	**0.988**	0.964	0.992	0.988	**0.994**	0.961	0.991	0.985	**0.994**
	RandomCrop	0.906	0.964	0.954	**0.972**	0.918	**0.976**	0.966	**0.976**	0.927	0.977	0.968	**0.981**
	RandomHorizontalFlip	0.932	0.981	0.971	**0.986**	0.964	0.989	0.983	**0.992**	0.954	0.987	0.981	**0.990**
	RandomRotation	0.879	0.952	0.939	**0.953**	0.909	0.972	0.955	**0.976**	0.918	0.974	0.961	**0.977**
	ColorJitter	0.952	0.982	0.975	**0.987**	0.967	0.993	0.988	**0.995**	0.964	0.992	0.985	**0.994**
	RandomCrop+ RandomHorizontalFlip	0.916	0.966	0.956	**0.972**	0.933	0.980	0.970	**0.983**	0.931	0.980	0.970	**0.983**
	Avg	0.923	0.972	0.962	**0.976**	0.943	0.984	0.975	**0.986**	0.943	0.984	0.975	**0.987**
Blended	None	0.795	0.856	0.861	**0.889**	0.814	0.923	0.885	**0.930**	0.867	0.949	0.919	**0.957**
	RandomCrop	0.792	0.830	0.865	**0.874**	0.818	0.888	0.885	**0.915**	0.821	0.903	0.896	**0.921**
	RandomHorizontalFlip	0.827	0.899	0.916	**0.930**	0.867	0.948	0.925	**0.959**	0.864	0.954	0.923	**0.962**
	RandomRotation	0.791	0.821	0.855	**0.864**	0.833	0.897	0.890	**0.926**	0.838	0.918	0.902	**0.933**
	ColorJitter	0.758	0.846	0.826	**0.861**	0.758	0.846	0.826	**0.921**	0.758	0.846	0.826	**0.950**
	RandomCrop+ RandomHorizontalFlip	0.827	0.883	0.911	**0.933**	0.855	0.931	0.932	**0.958**	0.864	0.926	0.924	**0.956**
	Avg	0.798	0.856	0.872	**0.892**	0.824	0.906	0.891	**0.935**	0.835	0.916	0.898	**0.947**
Optimized	None	0.844	0.973	0.910	**0.988**	0.840	0.952	0.897	**0.968**	0.902	0.982	0.940	**0.993**
	RandomCrop	0.952	0.998	0.990	**1.000**	0.966	0.999	0.994	**1.000**	0.970	0.999	0.993	**1.000**
	RandomHorizontalFlip	0.922	0.990	0.975	**0.998**	0.885	0.977	0.945	**0.993**	0.932	0.994	0.981	**0.998**
	RandomRotation	0.844	**0.893**	0.873	0.874	0.828	**0.930**	0.879	0.914	0.863	**0.958**	0.918	0.956
	ColorJitter	0.875	0.962	0.933	**0.991**	0.861	0.940	0.918	**0.977**	0.911	0.985	0.969	**0.995**
	RandomCrop+ RandomHorizontalFlip	0.979	0.999	0.996	**1.000**	0.979	0.999	0.997	**1.000**	0.982	0.999	0.997	**1.000**
	Avg	0.903	0.969	0.946	**0.975**	0.893	0.966	0.938	**0.975**	0.927	0.986	0.966	**0.990**

[61]), six data transformations (None, RandomCrop, RandomHorizontalFlip, RandomRotation, ColorJitter, RandomCrop+RandomHorizontalFlip), and three different DNN architectures (VGG16 [43], ResNet18 [19], PreAct-ResNet18 [20]). Furthermore, we explore the effects of four optimizers: SGD-0.01[¶], SGD-0.02, Adam-0.001, and Adam-0.002. The employed feature extractor $E(\cdot)$ is ResNet18, pre-trained on the specific dataset using the SGD optimizer with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 5e-4, along with two standard data augmentations: random crop and random horizontal flip. Each training iteration spans 70 epochs, conducted with a batch size of 256. In addition, Sec. V-E presents experiments involving various pre-trained feature extractors on the CIFAR-10 dataset. Our findings reveal that even utilizing a commonly used ImageNet pre-trained feature extractor yields notable enhancements compared to the baseline.

We perform a comprehensive comparative analysis, evaluating our method against the widely employed random selection (Random) [4], [14] and FUS [50] approaches. In the FUS search procedure, we utilize the VGG16 architecture, coupled with RandomCrop+RandomHorizontalFlip data transformations and SGD-0.01 as the optimizer to establish the proxy attack. Conversely, for our proposed method, we utilize pre-trained ResNet models for similarity measurement and set the diversity hyperparameter m to 10 across all three datasets. All other hyperparameters remain consistent with those outlined in Sec. III-A. Additionally, Sec. V-D shows the comparison with recent sample selection methods (FUS [50], OPS [17], Gradient [10], RD Score [49]) on the Tiny-ImageNet dataset.

[¶]SGD-0.01 implies the adoption of the SGD optimizer with an initial learning rate of 0.01.

B. Experiments on CIFAR-10 Dataset

Results on different data transforms and architectures. Table I provides a comprehensive overview of the attack success rates for BadNets, Blended, and Optimized triggers, considering various data transforms and architectures. Constructing the proxy attack for FUS, we employ VGG16 with RandomCrop+RandomHorizontalFlip and SGD-0.01 as the optimizer. Our proposed PFS method significantly enhances the poisoning efficacy in backdoor attacks. Across all 54 configurations, the attack success rate of poisoning samples selected through our approach consistently surpasses that of randomly selected samples at identical poisoning rates. On average, our method achieves a improvement of 0.044, 0.074, and 0.066 in terms of attack success rate for BadNets, Blended, and Optimized triggers, respectively. In general, our method outperforms FUS, demonstrating greater ASR in 48 of the 54 scenarios. Notably, FUS searches necessitate significantly more time compared to our method, often by hundreds or even thousands of times. Furthermore, when the model is VGG16 and the trigger is set to Blended, FUS surpasses our PFS. This can be attributed to two factors: (i) Both the proxy attack model employed in FUS and the model utilized in the victim phase are VGG16; (ii) In comparison to BadNets and Optimized triggers, the Blended trigger increases the similarity distance between benign and corresponding poisoning samples, thereby marginally impacting the performance of our proposed PFS. The combination of FUS and PFS, denoted as FUS+PFS, excels in 51 out of the 54 scenarios. This outcome stems from PFS effectively eliminating inefficient samples, consequently decreasing the search space for FUS.

TABLE II
THE ATTACK SUCCESS RATE UNDER DIFFERENT OPTIMIZERS ON THE CIFAR-10 DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS, BLENDED, AND OPTIMIZED ARE 1%, 1%, AND 0.5%, RESPECTIVELY. AMONG FOUR SELECT STRATEGIES, THE BEST RESULT IS DENOTED IN **BOLDFACE** WHILE THE UNDERLINE INDICATES THE SECOND-BEST RESULT. THE SETTINGS CORRESPONDING TO THE GRAY BACKGROUND ARE REPRESENTED AS THE PROXY POISONING ATTACK WITHIN THE ATTACKER PHASE AND THE ACTUAL POISONING PROCESS WITHIN THE VICTIM PHASE BEING CONSISTENT IN FUS.

Trigger	Optimizer	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	SGD-0.01	0.916	0.966	0.956	**0.972**	0.933	0.980	0.970	**0.983**	0.931	0.980	0.970	**0.983**
	SGD-0.02	0.918	0.971	0.962	**0.976**	0.938	0.983	0.975	**0.986**	0.943	0.984	0.976	**0.987**
	Adam-0.0005	0.920	0.934	0.963	**0.971**	0.945	0.960	0.975	**0.985**	0.944	0.960	0.976	**0.984**
	Adam-0.001	0.928	**0.971**	0.967	0.957	0.950	0.985	0.977	**0.986**	0.954	0.982	0.977	**0.985**
	Avg	0.921	0.961	0.962	**0.969**	0.942	0.977	0.974	**0.985**	0.943	0.977	0.975	**0.985**
Blended	SGD-0.01	0.827	0.883	0.911	**0.933**	0.855	0.931	0.932	**0.958**	0.864	0.926	0.924	**0.956**
	SGD-0.02	0.836	0.887	0.934	**0.935**	0.881	0.939	0.950	**0.968**	0.881	0.947	0.948	**0.967**
	Adam-0.0005	0.873	0.890	**0.924**	0.908	0.897	0.952	0.953	**0.974**	0.893	0.952	0.940	**0.966**
	Adam-0.001	0.886	0.910	**0.937**	0.909	0.900	0.953	0.951	**0.970**	0.896	0.949	0.943	**0.962**
	Avg	0.856	0.891	**0.927**	0.921	0.883	0.944	0.947	**0.968**	0.884	0.944	0.939	**0.963**
Optimized	SGD-0.01	0.979	0.999	0.996	**1.000**	0.979	0.999	0.997	**1.000**	0.982	0.999	0.997	**1.000**
	SGD-0.02	0.979	0.999	0.998	**1.000**	0.982	0.999	0.997	**1.000**	0.984	0.999	0.997	**1.000**
	Adam-0.0005	0.975	0.997	0.996	**1.000**	0.986	0.998	0.996	**1.000**	0.982	0.998	0.996	**1.000**
	Adam-0.001	0.980	0.999	0.997	**1.000**	0.985	0.999	0.997	**1.000**	0.982	0.998	0.994	**1.000**
	Avg	0.978	0.999	0.997	**1.000**	0.983	0.999	0.997	**1.000**	0.983	0.997	0.996	**1.000**

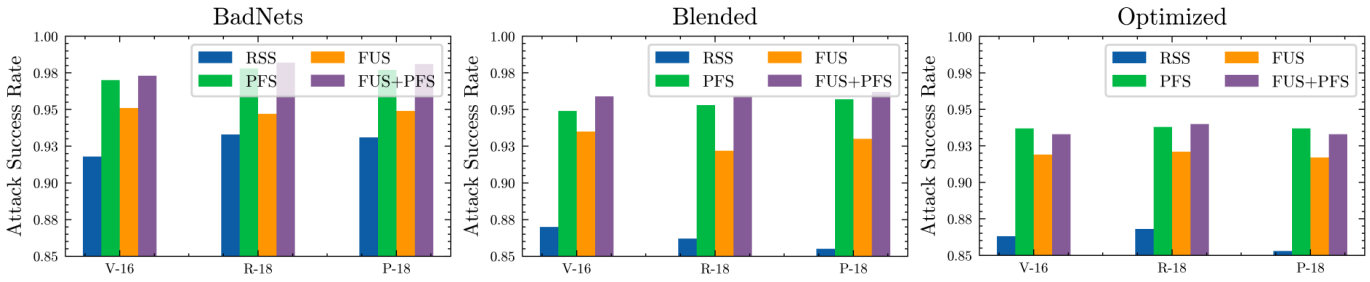

Fig. 7. Average ASR of different data transforms and optimizers on the Tiny-ImageNet dataset. Notably, PFS consistently outperforms FUS across all settings.

Results on different optimizer and architectures. Table II presents the ASR for the BadNets, Blended, and Optimized triggers, considering different optimizers and architectures. We utilize VGG16 with RandomCrop+RandomHorizontalFlip data transformations and SGD-0.01 as the optimizer for constructing the proxy attack in FUS. The outcomes underscore the effectiveness of our PFS method. Firstly, across all 36 configurations, our PFS consistently achieves higher attack success rates than random selection, with average improvements of 0.0437, 0.052 and 0.017 for the BadNets, Blended, and Optimized triggers, respectively. Secondly, our PFS outperforms FUS in 25 out of the 36 cases. Similar to previous experiments, when the model is VGG16 and the trigger is set to Blended, FUS outperforms our PFS. Notably, the combination of our method with FUS, referred to as FUS+PFS, achieves the best performance in 33 out of the 36 settings.

The **Supplementary Materials** contain additional experiments aimed at further evaluating the effectiveness and robustness of our proposed strategy. Specifically, in Sec. B and Sec. C, we provide the attack success rate (ASR) and benign accuracy (BA) under poisoning rates of 1.5% and 2% on the CIFAR-10 dataset, respectively. The outcomes illustrate the robustness of our method across different poisoning rates while maintaining benign accuracy. Moreover, in Sec. D, we present the ASR for an additional target class (class 3) on the CIFAR-10 dataset. Overall, our results consistently affirm the effectiveness of the proposed pfs and support our hypothesis that the efficiency of backdoor attacks depends on the interaction between individual similarity and set diversity.

C. Experiments on Tiny-ImageNet Dataset

Results on different data transforms and architectures. Table III shows that PFS outperforms the random method in terms of attack success rates of poisoning samples on Tiny-ImageNet, with average boosts of 0.054, 0.094, and 0.072 on BadNets, Blended, and Optimized triggers, respectively. In comparison to FUS, PFS outperforms FUS in 45 out of 54 cases. In contrast to experiments on the CIFAR-10 dataset, our proposed PFS method consistently achieves higher average ASR than FUS across different data transformations on the Tiny-ImageNet dataset. When the two methods are combined, the best results are achieved in 34 out of 54 settings, indicating the superiority of our method. These findings provide strong evidence supporting the effectiveness of our proposed method on the scenarios involving extensive datasets.

Results on different data transforms, optimizer, and architectures. To further illustrate the effectiveness of our proposed PFS, Figure 7 visually shows the average ASR across various data transforms and optimizers on the Tiny-ImageNet dataset. The findings indicate that all advanced

TABLE III
THE ATTACK SUCCESS RATE ON TINY-IMAGENET DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS, BLENDED, AND OPTIMIZED ARE 1%, 1%, AND 0.5%, RESPECTIVELY. AMONG FOUR SELECT STRATEGIES, THE BEST RESULT IS DENOTED IN **BOLDFACE** WHILE THE UNDERLINE INDICATES THE SECOND-BEST RESULT. THE SETTINGS CORRESPONDING TO THE GRAY BACKGROUND ARE REPRESENTED AS THE PROXY POISONING ATTACK WITHIN THE ATTACKER PHASE AND THE ACTUAL POISONING PROCESS WITHIN THE VICTIM PHASE BEING CONSISTENT IN FUS.

Trigger	Data Transform	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.952	**0.990**	0.983	0.989	0.960	**0.992**	0.985	0.991	0.956	**0.991**	0.981	0.989
	RandomCrop	0.893	**0.959**	0.950	0.958	0.917	0.970	0.963	**0.973**	0.904	**0.973**	0.958	0.971
	RandomHorizontalFlip	0.952	**0.989**	0.984	**0.989**	0.963	**0.991**	0.981	0.989	0.947	0.983	0.974	**0.985**
	RandomRotation	0.881	0.932	0.937	**0.944**	0.912	0.959	0.964	**0.966**	0.910	0.957	0.963	**0.970**
	ColorJitter	0.956	**0.990**	0.985	**0.990**	0.966	**0.993**	0.986	0.992	0.956	**0.990**	0.983	0.988
	RandomCrop+ RandomHorizontalFlip	0.900	0.943	**0.955**	0.950	0.917	**0.977**	0.967	0.975	0.924	0.974	0.965	**0.978**
	Avg	0.922	0.967	0.966	**0.970**	0.939	0.980	0.974	**0.981**	0.933	0.978	0.971	**0.980**
Blended	None	0.889	0.964	0.947	**0.975**	0.876	0.965	0.930	**0.973**	0.873	0.969	0.941	**0.978**
	RandomCrop	0.836	0.928	0.920	**0.942**	0.835	0.931	0.914	**0.936**	0.822	0.927	0.906	**0.932**
	RandomHorizontalFlip	0.905	0.966	0.964	**0.980**	0.888	0.964	0.950	**0.977**	0.886	0.965	0.954	**0.981**
	RandomRotation	0.865	0.931	0.946	**0.953**	0.872	0.945	0.942	**0.966**	0.870	0.950	0.941	**0.966**
	ColorJitter	0.883	0.958	0.945	**0.964**	0.871	0.956	0.917	**0.962**	0.872	0.967	0.931	**0.971**
	RandomCrop+ RandomHorizontalFlip	0.850	0.931	0.940	**0.958**	0.840	0.929	0.919	**0.942**	0.828	0.931	0.912	**0.940**
	Avg	0.871	0.946	0.944	**0.962**	0.864	0.948	0.929	**0.959**	0.859	0.952	0.931	**0.961**
Optimized	None	0.907	0.981	0.956	**0.984**	0.875	**0.968**	0.934	**0.968**	0.862	**0.970**	0.935	0.967
	RandomCrop	0.897	0.976	0.959	**0.980**	0.886	0.968	0.950	**0.972**	0.869	**0.958**	0.945	0.951
	RandomHorizontalFlip	0.903	0.979	0.956	**0.985**	0.892	**0.976**	0.950	**0.976**	0.877	**0.975**	0.948	0.974
	RandomRotation	0.899	**0.977**	0.957	0.972	0.897	**0.978**	0.956	0.972	0.888	**0.973**	0.948	0.956
	ColorJitter	0.695	0.714	**0.744**	0.707	0.769	0.773	**0.803**	0.775	0.772	0.779	**0.816**	0.781
	RandomCrop+ RandomHorizontalFlip	0.892	0.973	0.960	**0.974**	0.890	0.964	0.952	**0.968**	0.870	**0.963**	0.949	0.957
	Avg	0.866	0.933	0.922	**0.934**	0.868	0.938	0.924	**0.939**	0.856	**0.936**	0.924	0.931

TABLE IV
THE COMPARISON WITH RECENT SAMPLE SELECTION METHODS ON THE TINY-IMAGENET DATASET. ALL RESULTS ARE COMPUTED THE MEAN BY DIFFERENT DATA TRANSFORMS. THE POISONING RATE OF DIFFERENT POISONED ATTACKS IS SET TO 1%.

Sample Selection	Attacks		
	Blended	WaNet	ISSBA
RSS	0.864	0.813	0.764
FUS [50]	0.929	0.877	0.838
OPS [17]	0.882	0.843	0.792
Gradient [10]	0.901	0.851	0.808
RD Score [49]	0.913	0.847	0.806
PFS (Ours)	**0.948**	**0.893**	**0.861**

TABLE V
RUNNING TIME (SEC) OF DIFFERENT SELECTION METHODS ON THE CIFAR-10 AND TINY-IMAGENET DATASET WITH AN NVIDIA A100 GPU.

Dataset	PFS (w/o pre-training)	PFS (w pre-training)	FUS
CIFAR-10	17	562	9,150
Tiny-ImageNet	19	6784	106284

sample selection strategies (FUS, PFS, and FUS+PFS) achieve notably superior results compared to the random selection strategy (RSS). Furthermore, our proposed PFS consistently achieves better ASR compared to FUS across all settings.

D. Comparison with Recent Sample Selection Strategies

Table IV presents a comparative analysis between our PFS and recent sample selection strategies (FUS [50], OPS [17], Gradient [10], and RD Score [49]) on the Tiny-ImageNet dataset. Various recent triggers (Blended, WaNet, and ISSBA) are considered in this experiment. The results demonstrate that all sample selection strategies exhibit superior ASR compared to the random selection strategy (RSS), with our proposed PFS consistently demonstrating the highest sample efficiency across all settings when compared to other recent sample selection strategies.

E. Experiments on Different Pre-trained Feature Extractor

Although both FUS and PFS rely on a proxy task, there are essential differences between proxy tasks. FUS relies on a proxy attack task closely resembling the actual attack. Nevertheless, due to the inherent shortcuts of backdoor learning, overfitting to the data transformation and hyperparameter of proxy attack significantly impact actual injection efficiency. In contrast, our PFS selects efficient samples considering individual similarity and set diversity. As a result, PFS operates independently of the specific attack deployed during the victim phase, with minimal impact from data transformation and hyperparameters within the pre-trained feature extractor on sample efficiency during this phase. This independence is evidenced in our experiments with various pre-trained extractors, as illustrated in Figure 8. Remarkably, even a common extractor pre-trained on ImageNet results in considerable improvements compared to the baseline.

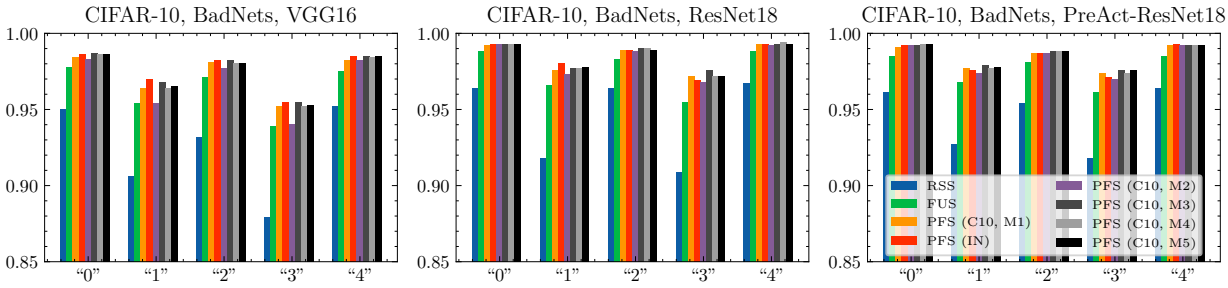

Fig. 8. The ASR among different data transforms ("0", "1", "2", "3", and "4" are the same as that on Figure 2) on the CIFAR-10 dataset. we use different pre-trained feature extractor to construct our PFS. C10: CIFAR-10. IN: ImageNet. M1: R18-SGD-RandFlip+RandCrop. M2: R18-Adam-RandRotation. M3: R18-SGD-RandRotation. M4: V16-SGD-RandRotation. M5: V16-SGD-None.

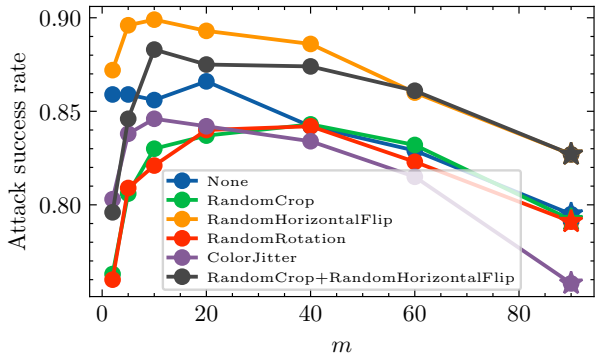

Fig. 9. Ablation study of diversity rate m on the CIFAR-10 dataset, where $\star$ indicate the results of Random Sampling.

F. Analysis of Time Consumption

Table V presents a comparison of the running times of our PFS with those of FUS. PFS demonstrates remarkable speed, being $538\times$ faster than FUS on CIFAR-10 dataset. Even with pre-training time included, our method is still approximately $16\times$ faster on CIFAR-10 dataset. Importantly, our method utilizes the **same** pre-trained model for different triggers, while FUS needs to conduct a complete search from the beginning for each trigger. This advantage makes our method a practical choice for situations where efficiency is crucial.

G. Ablation Study

We conduct an ablation study on the hyperparameter m, which determines the diversity of the poisoning sample set. The results are shown in Fig. 9, and the attack success rates demonstrate similar trends under different data transforms. Initially, as diversity gradually increases, the attack success rates also increase. However, too large of a value for m can reduce the role of similarity and weaken the attack. These findings highlight the importance of balancing similarity and diversity for efficient poisoning samples, supporting our claim in Sec. IV-A. In our PFS, we set $m = 10$ in all experimental settings.

H. Experiment with Backdoor Defense.

We evaluate the robustness of PFS against backdoor defense using the pruning-based defense method [32] and the tune-based defense method FST [36]. Pruning [32] is a widely used defense mechanism. It achieves this by pruning neurons that remain inactive in response to benign inputs, thereby neutralizing the effectiveness of the backdoor behavior. On the other hand, FST [36] is a tuning-based backdoor purification method. It operates by inducing feature shifts through deliberate adjustments to the classifier weights, effectively diverging them from the initially compromised weights. As depicted in Figure 10, PFS demonstrates superior robustness against Pruning [32] compared to RSS and FUS. Similarly, Figure 11 illustrates that our proposed PFS effectively against the tune-based defense method FST [36] compared to RSS and FUS. It is noteworthy that while FST [36] demonstrates superior defense capabilities compared to Pruning [32], it comes at the expense of a substantial reduction in benign accuracy.

VI. Conclusion and Future Work

Conclusion. This paper contributes empirical insights into two interesting phenomena associated with data efficiency in backdoor attacks. Firstly, we highlight the performance degeneration experienced by sample selection methods relying on proxy attacks when discrepancies arise between proxy attacks within attacker phase and actual poisoning processes within the victim phase. Furthermore, we argue that the significance of both the similarity between benign and corresponding poisoning samples and the diversity within the poisoning sample set as essential factors for efficient data in backdoor attacks. Leveraging these observations, we introduce a simple yet efficient proxy attack-free sample selection strategy that is driven by the similarity between benign and corresponding poisoning samples. Our method significantly improves poisoning efficiency across benchmark datasets with minimal additional cost.

Future Work. While the proposed PFS has demonstrated efficient data selection in backdoor attacks, we note that similarity between benign and corresponding poisoned samples does not necessarily guarantee efficiency, as similarity only filters out non-effective samples. In the future, it would be desirable to identify an unified indicator that positively correlates with data efficiency in backdoor attacks. Moreover, our proposed PFS is customized specifically for dirty-label backdoor attacks. In future research, we intend to explore efficient sample selection strategies designed specifically for clean-label backdoor attacks.

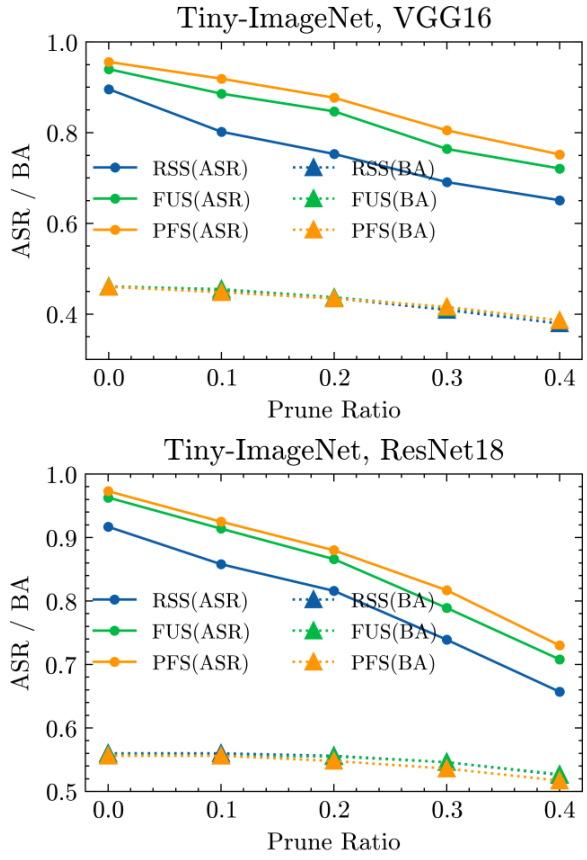

Fig. 10. Defense results of pruning [32] on the Tiny-ImageNet dataset, where the number of clean samples owned by the defender is 500, and the poisoning ratio is 1%.

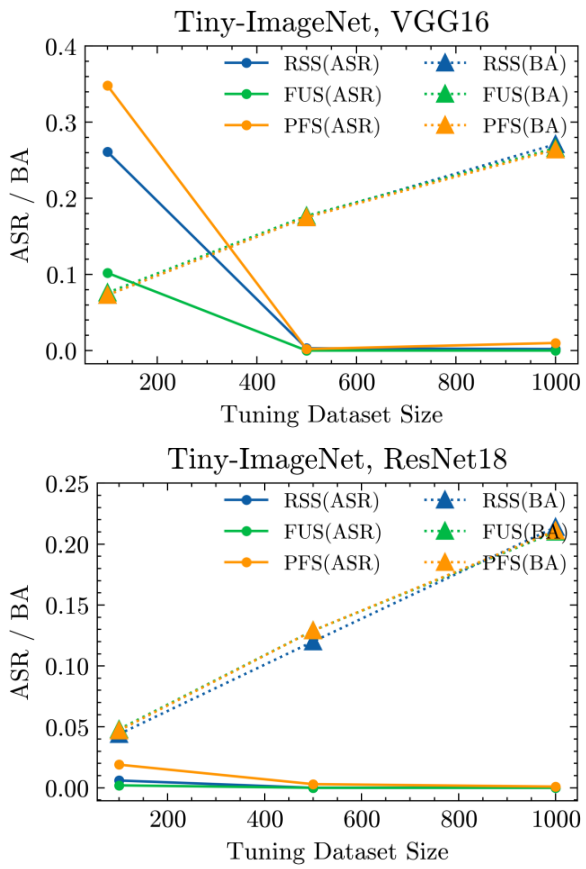

Fig. 11. Defense results of FST [36] on the Tiny-ImageNet dataset, where the number of clean samples owned by the defender are 100, 500, and 1000. The poisoning ratio is set to 1%.

ACKNOWLEDGMENTS

The work was supported in part by the National Natural Science Foundation of China under Grands U19B2044 and 61836011.

REFERENCES

[1] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1505–1521, 2021. 2

[2] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *International Conference on Computational Learning Theory*, pages 35–50. Springer, 2007. 7

[3] 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. 1

[4] 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. 1, 2, 3, 7, 8

[5] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1148–1156, 2021. 2

[6] Sanjoy Dasgupta. Two faces of active learning. *Theoretical computer science*, 412(19):1767–1781, 2011. 7

[7] Khoa Doan, Yingjie Lao, and Ping Li. Backdoor attack with imperceptible input and latent modification. *Advances in Neural Information Processing Systems*, 34:18944–18957, 2021. 2

[8] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16482–16491, 2021. 3

[9] Yu Feng, Benteng Ma, Jing Zhang, Shanshan Zhao, Yong Xia, and Dacheng Tao. Fiba: Frequency-injection based backdoor attack in medical image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20876–20885, 2022. 2

[10] Yinghua Gao, Yiming Li, Linghui Zhu, Dongxian Wu, Yong Jiang, and Shu-Tao Xia. Not all samples are born equal: Towards effective clean-label backdoor attacks. *Pattern Recognition*, 139:109512, 2023. 2, 3, 8, 10

[11] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 113–125, 2019. 3

[12] Yunjie Ge, Qian Wang, Baolin Zheng, Xinlu Zhuang, Qi Li, Chao Shen, and Cong Wang. Anti-distillation backdoor attacks: Backdoors can really survive in knowledge distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 826–834, 2021. 2

[13] 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. 1

[14] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 1, 2, 7, 8, 14

[15] Junfeng Guo, Ang Li, and Cong Liu. Aeva: Black-box backdoor detection using adversarial extreme value analysis. *arXiv preprint arXiv:2110.14880*, 2021. 7, 14

[16] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint*

arXiv:2302.03251, 2023. 7, 14
[17] Wei Guo, Benedetta Tondi, and Mauro Barni. A temporal chrominance trigger for clean-label backdoor attack against anti-spoof rebroadcast detection. *IEEE Transactions on Dependable and Secure Computing*, 2023. 2, 3, 8, 10
[18] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: defending against backdoor attacks using robust statistics. *arXiv preprint arXiv:2104.11315*, 2021. 3
[19] 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*, pages 770–778, 2016. 8
[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 8
[21] Sanghyun Hong, Pietro Frigo, Yiğitcan Kaya, Cristiano Giuffrida, and Tudor Dumitraș. Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 497–514, 2019. 2
[22] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018. 7, 14
[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 3
[24] Chaoran Li, Xiao Chen, Derui Wang, Sheng Wen, Muhammad Ejaz Ahmed, Seyit Camtepe, and Yang Xiang. Backdoor attack on machine learning based android malware detectors. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3357–3370, 2021. 1
[25] 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, 2020. 2
[26] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1, 3, 6
[27] 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*, pages 16463–16472, 2021. 2, 7, 14
[28] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021. 3
[29] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021. 3
[30] Yiming Li, Mingyan Zhu, Junfeng Guo, Tao Wei, Shu-Tao Xia, and Zhan Qin. Towards sample-specific backdoor attack with clean labels via attribute trigger. *arXiv preprint arXiv:2312.04584*, 2023. 7, 14
[31] Cong Liao, Haoti Zhong, Anna Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307*, 2018. 2
[32] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pages 273–294. Springer, 2018. 11, 12
[33] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017. 1
[34] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer, 2020. 2
[35] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021. 7
[36] Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. *arXiv preprint arXiv:2310.01875*, 2023. 11, 12
[37] 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*, pages 1765–1773, 2017. 2
[38] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021. 2, 7
[39] Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, and Ahmad-Reza Sadeghi. Poisoning attacks on federated learning-based iot intrusion detection system. In *Proc. Workshop Decentralized IoT Syst. Secur.(DISS)*, pages 1–7, 2020. 2
[40] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13198–13207, 2020. 2
[41] Goutham Ramakrishnan and Aws Albarghouthi. Backdoors in neural models of source code. *arXiv preprint arXiv:2006.06841*, 2020. 2
[42] Dongyu Ru, Jiangtao Feng, Lin Qiu, Hao Zhou, Mingxuan Wang, Weinan Zhang, Yong Yu, and Lei Li. Active sentence learning by adversarial uncertainty sampling in discrete space. *arXiv preprint arXiv:2004.08046*, 2020. 7
[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 8
[44] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 2
[45] Shuo Wang, Surya Nepal, Carsten Rudolph, Marthie Grobler, Shangyu Chen, and Tianle Chen. Backdoor attacks against transfer learning with pre-trained deep learning models. *IEEE Transactions on Services Computing*, 2020. 2
[46] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Mingli Zhu, Ruotong Wang, Li Liu, and Chao Shen. Backdoorbench: A comprehensive benchmark and analysis of backdoor learning. *arXiv preprint arXiv:2401.15002*, 2024. 6
[47] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021. 3
[48] Tong Wu, Tianhao Wang, Vikash Sehwag, Saeed Mahloujifar, and Prateek Mittal. Just rotate it: Deploying backdoor attacks via rotation transformation. *arXiv preprint arXiv:2207.10825*, 2022. 2
[49] Yutong Wu, Xingshuo Han, Han Qiu, and Tianwei Zhang. Computation and data efficient backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4805–4814, 2023. 8, 10
[50] 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*, pages 3992–3998, 2022. 1, 2, 3, 4, 5, 6, 8, 10
[51] Pengfei Xia, Hongjing Niu, Ziqiang Li, and Bin Li. Enhancing backdoor attacks with multi-level mmd regularization. *IEEE Transactions on Dependable and Secure Computing*, 2022. 2
[52] Pengfei Xia, Yueqi Zeng, Ziqiang Li, Wei Zhang, and Bin Li. Efficient trojan injection: 90% attack success rate using 0.04% poisoned samples, 2023. 1
[53] Zhen Xiang, David J Miller, and George Kesidis. Post-training detection of backdoor attacks for two-class and multi-attack scenarios. *arXiv preprint arXiv:2201.08474*, 2022. 3
[54] Runhua Xu, James BD Joshi, and Chao Li. Cryptonn: Training neural networks over encrypted data. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1199–1209. IEEE, 2019. 2
[55] Yu Yang, Tian Yu Liu, and Baharan Mirzasoleiman. Not all poisons are created equal: Robust training against data poisoning. In *International Conference on Machine Learning*, pages 25154–25165. PMLR, 2022. 3
[56] Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. Cold-start active learning through self-supervised language modeling. *arXiv preprint arXiv:2010.09535*, 2020. 7
[57] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. *arXiv preprint arXiv:2204.05255*, 2022. 1
[58] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16473–16481, 2021. 2
[59] Tongqing Zhai, Yiming Li, Ziqi Zhang, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Backdoor attack against speaker verification. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2560–2564. IEEE, 2021. 1
[60] 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*, pages 14443–14452, 2020. 1
[61] 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*, pages 97–108, 2020. 1, 2, 6, 8

APPENDIX

This supplementary material is organized as follows. We first discuss the attack success rate and benign accuracy under poisoning rates with 1.5% and 2% on the CIFAR-10 dataset in Sec. B and Sec. C, respectively. Additionally, we present the results of attack success rate under additional target classes on the CIFAR-10 dataset in Sec. D. Besides, Sec. V-C presents the experimental results on the Tiny-ImageNet dataset.

A. The Proof of Theorem 1

Theorem 2 *Suppose the training dataset consists of N benign samples $\{(x_i, y_i)\}_{i=1}^{N}$ and P poisoned samples $\{(x_i', k)\}_{i=1}^{P}$, whose images are i.i.d. sampled from uniform distribution and belonging to m classes. Assume that the DNN $f_\theta(\cdot)$ is a multivariate kernel regression $K(\cdot)$ and is trained via $\min_{\theta} \sum_{i=1}^{N} L\left(f_\theta(x), y\right) + \sum_{i=1}^{P} L\left(f_\theta\left(x'\right), k\right)$. For the expected predictive confidences over the target label k, we have: $\mathbb{E}_{x_t'}\left[f_\theta(x_t')\right] \propto \sum_{i=1}^{P} \frac{1}{(x_i'-x_i)\cdot(x_i'-x_t')}, i = 1, \cdots, P$, where x_t' is poisoned testing samples of attacks.*

Proof 1 (Proof of Theorem 1:) *We treat our model as a m-way kernel least square classifier and use a cross-entropy loss for training the kernel. The output of $f_\theta(\cdot)$ is a m-dimensional vector. Let us assume $\phi_t(\cdot) \in \mathbb{R}$ be expected predictive confidences corresponding to the target class k. Following previous works [15], [16], we know the kernel regression solution for neural tangent kernel (NTK) [22] is:*

$$\phi_t(\cdot) = \frac{\sum_{i=1}^{N} K\left(\cdot, \boldsymbol{x_i}\right) \cdot y_i + \sum_{i=1}^{P} K\left(\cdot, \boldsymbol{x_i'}\right) \cdot k}{\sum_{i=1}^{N} K\left(\cdot, \boldsymbol{x_i}\right) + \sum_{i=1}^{P} K\left(\cdot, \boldsymbol{x_i'}\right)}, \tag{2}$$

where $K\left(\cdot,\cdot\right)$ is the RBF kernel and $K\left(\boldsymbol{x}, \boldsymbol{x_i}\right) = e^{-2\gamma||\boldsymbol{x}-\boldsymbol{x_i}||^2} (\gamma > 0)$. Since the training samples are evenly distributed, there are $\frac{N}{m}$ benign samples belonging to k. Without loss of generality, we assume the target label $k = 1$ while others are 0. Then the regression solution can be re-formulated to:

$$\phi_t(\cdot) = \frac{\sum_{i=1}^{N/k} K\left(\cdot, \boldsymbol{x_i}\right) + \sum_{i=1}^{P} K\left(\cdot, \boldsymbol{x_i'}\right)}{\sum_{i=1}^{N} K\left(\cdot, \boldsymbol{x_i}\right) + \sum_{i=1}^{P} K\left(\cdot, \boldsymbol{x_i'}\right)}, \tag{3}$$

Therefore, following [30], we have:

$$\mathbb{E}_{\boldsymbol{x_i'}}\left[f_\theta(\boldsymbol{x_i'})\right] \triangleq \phi_t(\boldsymbol{x_t'}) = \frac{\sum_{i=1}^{N/k} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i}\right) + \sum_{i=1}^{P} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)}{\sum_{i=1}^{N} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i}\right) + \sum_{i=1}^{P} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)} \approx \frac{\sum_{i=1}^{P} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)}{\sum_{i=1}^{N} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i}\right) + \sum_{i=1}^{P} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)}. \tag{4}$$

Similar to [16], we also remove the term $\sum_{i=1}^{N/k} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i}\right)$ because $\boldsymbol{x_t'}$ typically don't belong to the target k and $\sum_{i=1}^{N/k} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i}\right) \ll \sum_{i=1}^{P} K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)$, otherwise the attacker has no incentive to craft poisoned samples.

When P close to N, which implies that the poisoning rate close to 50%, the attacker can achieve the optimal attack efficacy [14], [27]. Given $N = P$, we have:

$$\begin{aligned}
\phi_t(\boldsymbol{x_t'}) &= \sum_{i=1}^{P} \frac{K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)}{K\left(\boldsymbol{x_t'}, \boldsymbol{x_i}\right) + K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)} = \sum_{i=1}^{P} \frac{1}{1 + \frac{K(\boldsymbol{x_t'}, \boldsymbol{x_i})}{K\left(\boldsymbol{x_t'}, \boldsymbol{x_i'}\right)}} = \sum_{i=1}^{P} \frac{1}{1 + \frac{e^{-2\gamma||\boldsymbol{x_t'}-\boldsymbol{x_i}||^2}}{e^{-2\gamma||\boldsymbol{x_t'}-\boldsymbol{x_i'}||^2}}} \\
&= \sum_{i=1}^{P} \frac{1}{1 + e^{2\gamma||\boldsymbol{x_t'}-\boldsymbol{x_i'}||^2 - 2\gamma||\boldsymbol{x_t'}-\boldsymbol{x_i}||^2}} = \sum_{i=1}^{P} \frac{1}{1 + e^{2\gamma(\boldsymbol{x_i'}-\boldsymbol{x_i})\cdot(\boldsymbol{x_i'}+\boldsymbol{x_i}-2\boldsymbol{x_t'})}} \\
&\approx \sum_{i=1}^{P} \frac{1}{1 + e^{4\gamma(\boldsymbol{x_i'}-\boldsymbol{x_i})\cdot(\boldsymbol{x_i'}-\boldsymbol{x_t'})}}
\end{aligned} \tag{5}$$

we here let $\boldsymbol{x_i} = \boldsymbol{x_i'}$ in $\boldsymbol{x_i'} + \boldsymbol{x_i} - 2\boldsymbol{x_t'}$, because $\boldsymbol{x_i'}$ typically be similar as $\boldsymbol{x_i}$ and $\boldsymbol{x_i'} - \boldsymbol{x_i} \ll \boldsymbol{x_i'} - \boldsymbol{x_t'}$. Therefore, we have:

$$\mathbb{E}_{x_t'}\left[f_\theta(x_t')\right] \propto \sum_{i=1}^{P} \frac{1}{(\boldsymbol{x_i'} - \boldsymbol{x_i}) \cdot (\boldsymbol{x_i'} - \boldsymbol{x_t'})}. \tag{6}$$

B. Attack Success Rate Under Additional poisoning rates on the CIFAR-10 Dataset

We evaluate the attack success rate (ASR) under different poisoning rates on the CIFAR-10 dataset. Tables VI and VII present the ASR with trigger BadNets and Blended at poisoning rates of 1.5% and 2%, respectively. Results under poisoning rates of 1.5% and 2% are similar to those at 1%.

(i) Experiments under poisoning rates of 1.5%: Our PFS approach outperforms random selection in all cases, with average boosts of 0.030 and 0.055 on BadNets and Blended triggers, respectively. The PFS method also achieves better attack results than FUS in most cases (27/36). The combination of these two methods achieves superior results in all 36 settings.

TABLE VI
THE ATTACK SUCCESS RATE (ASR) ON THE CIFAR-10 DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS AND BLENDED ARE 1.5%, RESPECTIVELY. AMONG FOUR SELECT STRATEGIES, THE BEST RESULT IS DENOTED IN **BOLDFACE** WHILE THE UNDERLINE INDICATES THE SECOND-BEST RESULT. THE SETTINGS CORRESPONDING TO THE GRAY BACKGROUND ARE REPRESENTED AS THE PROXY POISONING ATTACK WITHIN THE ATTACKER PHASE AND THE ACTUAL POISONING PROCESS WITHIN THE VICTIM PHASE BEING CONSISTENT IN FUS.

Trigger	Data Transform	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.961	0.986	0.982	**0.992**	0.976	0.993	0.990	**0.997**	0.973	0.992	0.989	**0.997**
	RandomCrop	0.924	0.970	0.966	**0.983**	0.944	0.980	0.974	**0.988**	0.946	0.979	0.977	**0.989**
	RandomHorizontalFlip	0.952	0.983	0.979	**0.991**	0.973	0.975	0.986	**0.995**	0.967	0.988	0.984	**0.994**
	RandomRotation	0.915	0.968	0.958	**0.977**	0.929	0.979	0.971	**0.988**	0.941	0.979	0.973	**0.988**
	ColorJitter	0.958	0.986	0.981	**0.992**	0.977	0.994	0.990	**0.997**	0.972	0.993	0.988	**0.996**
	RandomCrop+ RandomHorizontalFlip	0.932	0.970	0.965	**0.981**	0.949	0.980	0.978	**0.989**	0.952	0.981	0.978	**0.989**
	Avg	0.940	0.977	0.972	**0.986**	0.958	0.984	0.982	**0.992**	0.959	0.985	0.982	**0.992**
Blended	None	0.866	0.921	0.918	**0.953**	0.883	0.954	0.939	**0.971**	0.909	0.970	0.954	**0.982**
	RandomCrop	0.855	0.900	0.933	**0.942**	0.879	0.935	0.943	**0.963**	0.882	0.948	0.938	**0.967**
	RandomHorizontalFlip	0.889	0.937	0.952	**0.970**	0.906	0.966	0.960	**0.983**	0.927	0.975	0.961	**0.987**
	RandomRotation	0.859	0.909	0.929	**0.937**	0.894	0.938	0.947	**0.967**	0.901	0.955	0.953	**0.972**
	ColorJitter	0.851	0.915	0.912	**0.938**	0.872	0.950	0.929	**0.969**	0.913	0.966	0.945	**0.976**
	RandomCrop+ RandomHorizontalFlip	0.885	0.931	0.952	**0.965**	0.905	0.958	0.964	**0.982**	0.919	0.967	0.967	**0.984**
	Avg	0.868	0.919	0.933	**0.951**	0.890	0.950	0.947	**0.973**	0.909	0.964	0.953	**0.978**

TABLE VII
THE ATTACK SUCCESS RATE (ASR) ON THE CIFAR-10 DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS AND BLENDED ARE 2%, RESPECTIVELY. AMONG FOUR SELECT STRATEGIES, THE BEST RESULT IS DENOTED IN **BOLDFACE** WHILE THE UNDERLINE INDICATES THE SECOND-BEST RESULT. THE SETTINGS CORRESPONDING TO THE GRAY BACKGROUND ARE REPRESENTED AS THE PROXY POISONING ATTACK WITHIN THE ATTACKER PHASE AND THE ACTUAL POISONING PROCESS WITHIN THE VICTIM PHASE BEING CONSISTENT IN FUS.

Trigger	Data Transform	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.971	0.987	0.986	**0.993**	0.982	0.994	0.992	**0.998**	0.977	0.993	0.991	**0.997**
	RandomCrop	0.940	0.973	0.970	**0.986**	0.954	0.980	0.979	**0.991**	0.946	0.982	0.979	**0.992**
	RandomHorizontalFlip	0.961	0.984	0.981	**0.992**	0.976	0.991	0.988	**0.995**	0.975	0.989	0.986	**0.995**
	RandomRotation	0.931	0.969	0.966	**0.986**	0.948	0.977	0.977	**0.992**	0.946	0.981	0.979	**0.991**
	ColorJitter	0.966	0.986	0.984	**0.993**	0.982	0.993	0.992	**0.998**	0.981	0.992	0.990	**0.997**
	RandomCrop+ RandomHorizontalFlip	0.938	0.974	0.972	**0.986**	0.957	0.982	0.980	**0.992**	0.959	0.984	0.981	**0.991**
	Avg	0.951	0.979	0.977	**0.989**	0.967	0.986	0.985	**0.994**	0.964	0.987	0.984	**0.994**
Blended	None	0.914	0.943	**0.952**	**0.952**	0.916	0.969	0.955	**0.981**	0.937	0.978	0.967	**0.988**
	RandomCrop	0.896	0.937	0.958	**0.966**	0.912	0.961	0.961	**0.978**	0.913	0.960	0.965	**0.985**
	RandomHorizontalFlip	0.923	0.955	0.972	**0.983**	0.931	0.976	0.973	**0.990**	0.947	0.982	0.976	**0.991**
	RandomRotation	0.900	0.936	0.952	**0.963**	0.915	0.958	0.958	**0.973**	0.922	0.965	0.965	**0.982**
	ColorJitter	0.904	0.940	0.951	**0.963**	0.918	0.965	0.949	**0.979**	0.938	0.973	0.964	**0.986**
	RandomCrop+ RandomHorizontalFlip	0.922	0.955	0.973	**0.986**	0.927	0.968	0.977	**0.991**	0.933	0.973	0.979	**0.992**
	Avg	0.910	0.944	0.960	**0.969**	0.920	0.966	0.962	**0.969**	0.932	0.972	0.969	**0.987**

(ii) Experiments under poisoning rates of 2%: Our PFS approach outperforms random selection in all cases, with average boosts of 0.023 and 0.04 on BadNets and Blended triggers, respectively. The PFS method also achieves better attack results than FUS in most cases (26/36). The combination of these two methods achieves superior results in all 36 settings.

C. Benign Accuracy Under Additional poisoning rates on the CIFAR-10 Dataset

We also present the benign accuracy (BA) (*i.e.*, the probability of classifying a benign test data to the correct label) under other poisoning rates on the CIFAR-10 dataset. Table VIII and Table IX illustrate the benign accuracy with trigger BadNets and Blended under poisoning rates of 1.5% and 2%, respectively. The results show the similar BA with four strategies, which demonstrates that our proposed strategy has no negative impact on benign accuracy.

D. Attack Success Rate Under Additional Target Classes on the CIFAR-10 Dataset

This section presents the attack success rate (ASR) under additional target classes on the CIFAR-10 dataset. Table X illustrates the attack success rate with trigger BadNets and Blended under target category 3. Compared with random selection, our PFS

TABLE VIII
THE BENIGN ACCURACY (BA) ON THE CIFAR-10 DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS AND BLENDED ARE 1.5%, RESPECTIVELY.

Trigger	Data Transform	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.868	0.868	0.865	0.866	0.837	0.837	0.838	0.835	0.850	0.849	0.848	0.847
	RandomCrop	0.903	0.905	0.905	0.903	0.914	0.912	0.914	0.911	0.914	0.914	0.914	0.914
	RandomHorizontalFlip	0.892	0.892	0.893	0.894	0.878	0.879	0.879	0.878	0.886	0.888	0.886	0.883
	RandomRotation	0.876	0.873	0.876	0.874	0.883	0.881	0.880	0.880	0.889	0.888	0.889	0.889
	ColorJitter	0.860	0.860	0.860	0.859	0.833	0.829	0.833	0.830	0.843	0.844	0.841	0.843
	RandomCrop+ RandomHorizontalFlip	0.920	0.919	0.920	0.921	0.927	0.926	0.928	0.925	0.927	0.928	0.929	0.927
	Avg	0.887	0.886	0.887	0.886	0.879	0.877	0.879	0.877	0.885	0.885	0.885	0.884
Blended	None	0.865	0.866	0.865	0.866	0.836	0.832	0.832	0.833	0.846	0.846	0.849	0.847
	RandomCrop	0.904	0.906	0.904	0.904	0.912	0.913	0.912	0.913	0.913	0.914	0.915	0.914
	RandomHorizontalFlip	0.892	0.892	0.889	0.891	0.876	0.878	0.875	0.875	0.882	0.884	0.885	0.887
	RandomRotation	0.873	0.875	0.875	0.873	0.880	0.880	0.881	0.880	0.887	0.888	0.889	0.889
	ColorJitter	0.858	0.859	0.860	0.858	0.829	0.832	0.824	0.828	0.839	0.841	0.843	0.842
	RandomCrop+ RandomHorizontalFlip	0.922	0.920	0.919	0.920	0.928	0.927	0.928	0.928	0.928	0.927	0.929	0.928
	Avg	0.886	0.886	0.885	0.885	0.877	0.877	0.875	0.876	0.883	0.883	0.885	0.885

TABLE IX
THE BENIGN ACCURACY (BA) ON THE CIFAR-10 DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS AND BLENDED ARE 2%, RESPECTIVELY.

Trigger	Data Transform	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.867	0.865	0.866	0.863	0.837	0.835	0.835	0.833	0.851	0.846	0.848	0.847
	RandomCrop	0.904	0.905	0.905	0.905	0.913	0.914	0.912	0.912	0.916	0.915	0.913	0.913
	RandomHorizontalFlip	0.891	0.891	0.891	0.891	0.878	0.878	0.878	0.876	0.886	0.885	0.886	0.885
	RandomRotation	0.873	0.876	0.874	0.872	0.882	0.882	0.881	0.880	0.888	0.889	0.889	0.890
	ColorJitter	0.862	0.857	0.857	0.859	0.830	0.827	0.829	0.831	0.844	0.843	0.843	0.841
	RandomCrop+ RandomHorizontalFlip	0.920	0.920	0.919	0.919	0.928	0.926	0.926	0.925	0.928	0.930	0.928	0.927
	Avg	0.886	0.886	0.885	0.885	0.878	0.877	0.877	0.876	0.886	0.885	0.885	0.884
Blended	None	0.865	0.866	0.865	0.866	0.836	0.832	0.832	0.833	0.846	0.846	0.849	0.847
	RandomCrop	0.904	0.906	0.904	0.904	0.912	0.913	0.912	0.913	0.913	0.914	0.915	0.914
	RandomHorizontalFlip	0.892	0.892	0.889	0.891	0.876	0.878	0.875	0.875	0.882	0.884	0.885	0.887
	RandomRotation	0.873	0.875	0.875	0.873	0.880	0.880	0.881	0.880	0.887	0.888	0.889	0.889
	ColorJitter	0.858	0.859	0.860	0.858	0.829	0.832	0.824	0.828	0.839	0.841	0.843	0.842
	RandomCrop+ RandomHorizontalFlip	0.922	0.920	0.919	0.920	0.928	0.927	0.928	0.928	0.928	0.927	0.929	0.928
	Avg	0.886	0.886	0.885	0.885	0.877	0.877	0.875	0.876	0.883	0.883	0.885	0.885

approach outperforms in all cases, with average boosts of 0.045 and 0.076 on BadNets and Blended triggers. The PFS method also achieves better attack results in most cases compared to FUS (22/24). The combination of these two achieves superior results in all 24 settings. These outcomes consistently confirm the conclusions we reached in the paper.

E. Experiments on CIFAR-100 Dataset

The results for CIFAR-100 in Table XI are similar to those for CIFAR-10. Our PFS approach outperforms random selection in all cases, with average boosts of 0.047, 0.068, and 0.037 for BadNets, Blended, and Optimized triggers, respectively. Compared to FUS, PFS achieves better attack results in 49 out of 54 settings.

F. Experiments on Different Pre-trained Feature Extractor

To compute the cosine similarity in the proposed PFS, a pre-trained feature extractor E is required. This section examines the impact of various pre-trained feature extractors E on the CIFAR-10 dataset. As demonstrated in Fig. 12, even using a common feature extractor pre-trained on the ImageNet dataset yields significant improvements compared to the baseline.

TABLE X

THE ATTACK SUCCESS RATE (ASR) ON THE CIFAR-10 DATASET. IN THIS EXPERIMENT, ATTACK TARGET K IS SET TO CATEGORY 3. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS AND BLENDED ARE 1%, RESPECTIVELY. AMONG FOUR SELECT STRATEGIES, THE BEST RESULT IS DENOTED IN **BOLDFACE** WHILE THE UNDERLINE INDICATES THE SECOND-BEST RESULT. THE SETTINGS CORRESPONDING TO THE GRAY BACKGROUND ARE REPRESENTED AS THE PROXY POISONING ATTACK WITHIN THE ATTACKER PHASE AND THE ACTUAL POISONING PROCESS WITHIN THE VICTIM PHASE BEING CONSISTENT IN FUS.

Trigger	Data Transform	Model							
		VGG16				ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.950	0.984	0.977	**0.986**	0.959	0.991	0.985	**0.994**
	RandomCrop	0.909	0.960	0.954	**0.968**	0.925	0.976	0.966	**0.979**
	RandomHorizontalFlip	0.934	0.980	0.972	**0.983**	0.959	0.987	0.981	**0.990**
	RandomRotation	0.890	**0.951**	0.942	**0.951**	0.912	0.968	0.955	**0.972**
	ColorJitter	0.945	0.983	0.975	**0.986**	0.966	0.991	0.985	**0.993**
	RandomCrop+ RandomHorizontalFlip	0.905	0.964	0.957	**0.968**	0.925	0.978	0.968	**0.980**
	Avg	0.922	0.970	0.963	**0.974**	0.941	0.982	0.973	**0.985**
Blended	None	0.774	0.867	0.851	**0.899**	0.806	0.915	0.850	**0.935**
	RandomCrop	0.780	0.813	0.822	**0.869**	0.791	0.902	0.827	**0.918**
	RandomHorizontalFlip	0.836	0.899	0.887	**0.934**	0.845	0.949	0.888	**0.962**
	RandomRotation	0.793	0.832	0.813	**0.847**	0.804	0.898	0.840	**0.903**
	ColorJitter	0.778	0.836	0.805	**0.874**	0.814	0.907	0.854	**0.927**
	RandomCrop+ RandomHorizontalFlip	0.815	0.843	0.871	**0.884**	0.838	0.927	0.893	**0.951**
	Avg	0.796	0.848	0.842	**0.885**	0.816	0.916	0.859	**0.933**

TABLE XI

THE ATTACK SUCCESS RATE ON CIFAR-100 DATASET. ALL RESULTS ARE AVERAGED OVER 5 DIFFERENT RUNS. THE POISONING RATES OF EXPERIMENTS WITH TRIGGER BADNETS, BLENDED, AND OPTIMIZED ARE 1.5%, 1.5%, AND 0.75%, RESPECTIVELY. AMONG FOUR SELECT STRATEGIES, THE BEST RESULT IS DENOTED IN **BOLDFACE** WHILE THE UNDERLINE INDICATES THE SECOND-BEST RESULT. THE SETTINGS CORRESPONDING TO THE GRAY BACKGROUND ARE REPRESENTED AS THE PROXY POISONING ATTACK WITHIN THE ATTACKER PHASE AND THE ACTUAL POISONING PROCESS WITHIN THE VICTIM PHASE BEING CONSISTENT IN FUS.

Trigger	Data Transform	Model											
		VGG16				ResNet18				PreAct-ResNet18			
		Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS	Random	PFS	FUS	FUS+PFS
BadNets	None	0.933	0.970	0.965	**0.971**	0.950	0.978	0.973	**0.979**	0.949	**0.977**	0.970	**0.977**
	RandomCrop	0.882	**0.947**	0.930	0.945	0.894	0.951	0.940	**0.952**	0.888	**0.951**	0.940	**0.951**
	RandomHorizontalFlip	0.920	0.964	0.958	**0.965**	0.936	0.970	0.965	**0.971**	0.929	**0.968**	0.962	**0.968**
	RandomRotation	0.868	0.923	0.912	**0.924**	0.866	**0.939**	0.925	0.938	0.875	0.935	0.926	**0.938**
	ColorJitter	0.934	0.970	0.965	**0.971**	0.952	0.978	0.973	**0.979**	0.946	0.977	0.970	**0.978**
	RandomCrop+ RandomHorizontalFlip	0.879	**0.939**	0.929	**0.939**	0.888	0.947	0.942	**0.950**	0.894	0.949	0.941	**0.951**
	Avg	0.903	0.952	0.943	**0.953**	0.914	0.961	0.953	**0.962**	0.914	0.960	0.952	**0.961**
Blended	None	0.866	0.923	0.920	**0.941**	0.884	0.963	0.946	**0.974**	0.894	0.975	0.950	**0.982**
	RandomCrop	0.863	0.874	**0.891**	0.880	0.826	0.916	0.906	**0.938**	0.829	0.931	0.912	**0.942**
	RandomHorizontalFlip	0.891	0.945	0.935	**0.960**	0.902	0.968	0.951	**0.977**	0.912	0.974	0.958	**0.982**
	RandomRotation	0.818	0.862	**0.901**	0.865	0.841	0.916	0.916	**0.936**	0.848	0.926	0.921	**0.944**
	ColorJitter	0.863	0.920	0.917	**0.931**	0.904	0.972	0.956	**0.977**	0.913	0.979	0.959	**0.983**
	RandomCrop+ RandomHorizontalFlip	0.811	0.865	**0.905**	0.872	0.826	0.918	0.920	**0.942**	0.850	0.931	0.924	**0.947**
	Avg	0.852	0.898	**0.912**	0.908	0.864	0.942	0.933	**0.957**	0.874	0.953	0.937	**0.963**
Optimized	None	0.979	**0.999**	0.995	**0.999**	0.884	0.972	0.931	**0.982**	0.895	0.983	0.962	**0.987**
	RandomCrop	0.989	0.999	0.997	**1.000**	0.983	0.999	0.996	**1.000**	0.981	0.999	0.995	**1.000**
	RandomHorizontalFlip	0.985	0.999	0.997	**1.000**	0.938	**0.998**	0.980	0.983	0.938	0.997	0.984	**0.998**
	RandomRotation	0.810	0.831	**0.867**	0.785	0.827	**0.909**	0.882	0.882	0.835	**0.908**	0.883	0.900
	ColorJitter	0.984	0.997	0.993	**1.000**	0.967	**0.989**	0.979	0.985	0.956	0.993	0.990	**0.994**
	RandomCrop+ RandomHorizontalFlip	0.991	0.999	0.997	**1.000**	0.987	0.999	0.998	**1.000**	0.982	0.998	0.994	**0.999**
	Avg	0.956	0.971	**0.974**	0.964	0.931	**0.978**	0.961	0.972	0.931	**0.980**	0.968	**0.980**

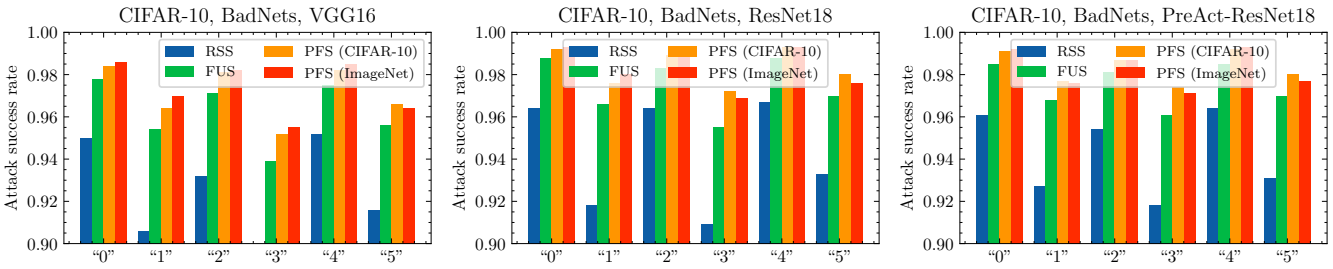

Fig. 12. Experiments on Different Pre-trained Models. The attack success rate under different pre-trained models on the CIFAR-10 dataset. All results are computed the mean by 5 different run. Experiment settings are the same as that on the Table 1 of the original paper.