Run-Off Election: Improved Provable Defense against Data Poisoning Attacks

Keivan Rezaei [*1] **Kiarash Banihashem** [*1] **Atoosa Chegini** [1] **Soheil Feizi** [1]

Abstract

In data poisoning attacks, an adversary tries to change a model's prediction by adding, modifying, or removing samples in the training data. Recently, *ensemble-based* approaches for obtaining *provable* defenses against data poisoning have been proposed where predictions are done by taking a majority vote across multiple base models. In this work, we show that merely considering the majority vote in ensemble defenses is wasteful as it does not effectively utilize available information in the logits layers of the base models. Instead, we propose *Run-Off Election (ROE)*, a novel aggregation method based on a two-round election across the base models: In the first round, models vote for their preferred class and then a second, *Run-Off* election is held between the top two classes in the first round. Based on this approach, we propose DPA+ROE and FA+ROE defense methods based on Deep Partition Aggregation (DPA) and Finite Aggregation (FA) approaches from prior work. We evaluate our methods on MNIST, CIFAR-10, and GTSRB and obtain improvements in certified accuracy by up to 3%-4%. Also, by applying ROE on a boosted version of DPA [1], we gain improvements around 12%-27% comparing to the current state-of-the-art, establishing **a new state-of-the-art** in (pointwise) certified robustness against data poisoning. In many cases, our approach outperforms the state-of-the-art, even when using 32 times less computational power.

1. Introduction

In recent years, Deep Neural Networks (DNNs) have achieved great success in many research areas, such as computer vision (He et al., 2016) and natural language processing (Chen, 2015) and have become the standard method of choice in many applications. Despite this success, these methods are vulnerable to *poisoning attacks* where the adversary manipulates the training data in order to change the classifications of specific inputs at the test time (Chen et al., 2017; Shafahi et al., 2018; Gao et al., 2021). Since large datasets are obtained using methods such as crawling the web, this issue has become increasingly important as deep models are adopted in safety-critical applications.

While empirical defense methods have been proposed to combat this problem using approaches such as data augmentation and data sanitization (Hodge & Austin, 2004; Paudice et al., 2018; Gong et al., 2020; Borgnia et al., 2021; Ni et al., 2021), the literature around poisoning has followed something of a "cat and mouse" game as in the broader literature on adversarial robustness, where defense methods are quickly broken using adaptive and stronger attack techniques (Carlini et al., 2019). To combat this, several works have focused on obtaining *certifiable defenses* that are *provably robust* against the adversary, regardless of the attack method. These works provide a *certificate* for each sample that is a guaranteed lower bound on the amount of distortion on the training set required to change the model's prediction.

The most scalable provable defenses against data poisoning have been considered the use of *ensemble methods* that are composed of multiple base classifiers (Levine & Feizi, 2020; Chen et al., 2020; Jia et al., 2021; Wang et al., 2022b; Chen et al., 2022). At the test time, the prediction of these models is aggregated by taking a majority vote across them. Depending on the exact method, the certificates may be deterministic or stochastic. For instance, Deep Partition Aggregation (DPA) method of (Levine & Feizi, 2020) trains multiple models on disjoint subsets of the training data. Since each poisoned sample can affect at most one model, this leads to a deterministic certificate based on the gap between the predicted and the runner-up class. This can be wasteful, however as the models predicting other than the top two classes are ignored. While the choice of the partitioning scheme used for training the models has been extensively considered in the literature, for both deterministic (Levine & Feizi, 2020; Wang et al., 2022b) and stochastic (Chen et al., 2020; Jia et al., 2021) partitioning schemes, all of these approaches share the problem as they take a majority vote at test time.

[*]Equal contribution [1]Department of Computer Science, University of Maryland, MD, USA. Correspondence to: Keivan Rezaei <krezaei@umd.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]We thank anonymous reviewer for proposing this method.

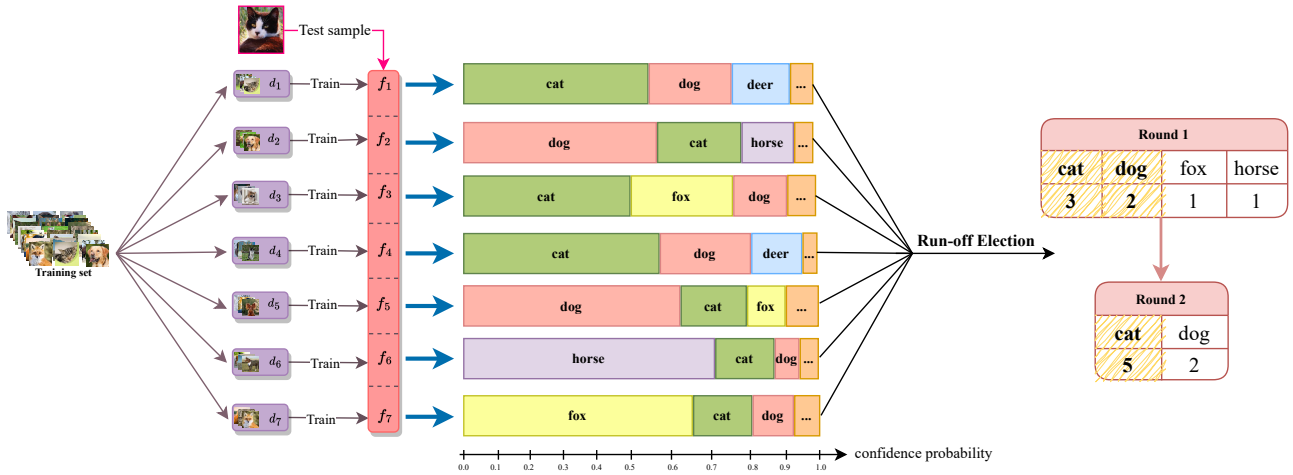

Figure 1. An example illustrating our proposed method **(Run-off Election)**. Training dataset is partitioned into 7 parts ($d_1, d_2, ..., d_7$) and 7 separate classifiers are trained on each part ($f_1, f_2, ..., f_7$). At test time, after giving the input to the classifiers, we obtain the logits-layer information of each of them. For example, the class `dog` has the highest logits-layer score for classifier f_2. In our method (see Section 3), we hold a two-round election. In the first round, each model votes for its top class, and we find the top-two classes with the most votes. In the second run, each model votes for one of these two classes based on the logits-layer information, e.g., f_6 votes for `cat` as it prefers it to `dog`. Existing methods output the most repeated class and can be fooled using a single poisoned sample, e.g., by changing the prediction of f_3 to `dog`. As we prove in Section 3.3 (Theorem 3.4), our method is more robust and the adversary needs at least two poisoned samples to change the model's prediction in this example. This is due to the fact the **gap between the number of votes of the top two classes effectively increases in the second round.**

In this work, we propose a novel aggregation method called Run-Off Election (ROE) that greatly improves on existing approaches using a *two-round election* among the models. In the first round, models vote for their preferred class, and we obtain the top two classes in terms of the number of votes. We then hold a second, *Run-Off* election round where all base models vote for one of these two classes. These votes are obtained using the *logits layer* information of the models, where each model votes for the class with the higher score in this layer. By using all of the base models for prediction, we effectively increase the gap between the predicted class and the runner-up, leading to an improved certificate. An illustrative example of our approach is shown in Figure 1. Our method is general and, in principle, can be applied to any kind of deterministic or stochastic ensemble method. In this paper, we focus on deterministic methods and develop DPA+ROE and FA+ROE defenses based on the Deep Partition Aggregation (DPA) (Levine & Feizi, 2020) and Finite Aggregation (FA) (Wang et al., 2022b) respectively and calculate the prediction certificates.

Technical challenges of calculating certificates. Compared to the majority vote method used in prior work, calculating certificates in ROE is more challenging since characterizing the adversary's optimal actions is more complex. Letting c^{pred} and c^{sec} denote the predicted class and the runner-up class, the adversary can change the model's prediction in many ways, as we briefly discuss below.

1. It can get the c^{sec} elected in the second round by poisoning models to change their votes from the predicted class to the runner-up class.
2. It can get c^{pred} eliminated in the first round by ensuring that at least two classes receive more votes than c^{pred} in the first round.
3. For some $c \notin \{c^{\text{pred}}, c^{\text{sec}}\}$, it can ensure that c receives more votes than c^{sec} in the first round and receives more votes than c^{pred} in the second round. This leads to the counter-intuitive situation where the adversary *decreases* the votes of the runner-up class c^{sec}.

In order to obtain a unified argument for all of these cases, we introduce the concepts of a *1v1 certificate* and *2v1 certificate*. Intuitively, a 1v1 certificate bounds the number of poisoned samples required for one class to "beat" another class, i.e., receive more votes. This is similar to the prediction certificate of majority vote, with the distinction that we consider any two arbitrary classes. A 2v1 certificate extends this idea and bounds the number of poisoned samples required for *two* classes to beat another class simultaneously.

We will show that as long as the 1v1 and 2v1 certificates can be calculated efficiently, we can use them to calculate a prediction certificate in ROE. Taking a reduction approach is beneficial as it ensures that our method works for any choice of ensembles, as long as 1v1 and 2v1 certificates can

be calculated. Focusing on DPA+ROE and FA+ROE, we show that the 1v1 certificates can be calculated using similar methods as the prediction certificates for the majority vote. Calculating 2v1 certificates are more complex, however as the adversary needs to "spread" its effort between two classes, and it is not straightforward how this should be done. For DPA+ROE, we use a **dynamic programming** based approach to recursively solve this problem. The argument, however, does not extend to FA+ROE since the underlying partitioning scheme is more complex and the adversary is more constrained in its actions. For FA+ROE, we deal with this challenge using a **duality-based approach** that considers the **convex combination of the adversary's constraint set**.

By reasoning on the adversary's behavior as above, we obtain two separate certificates to ensure that the predicted class is unchanged in each of the two rounds. Since the adversary can change our prediction in either of the rounds, we take the minimum of two numbers as our final certificate. We further refer to Section 3.3 for more details on the calculation of the certificate in DPA+ROE and FA+ROE.

Empirical results. We evaluate our model in the context of *deterministic* robustness certificates and observe substantial improvements over the existing state-of-the-art (FA). **FA+ROE can improve certified accuracy by up to** 4.79%**,** 4.73%**, and** 3.54% **respectively on MNIST, CIFAR-10, and GTSRB datasets.** Furthermore, in some cases, **DPA+ROE also outperforms FA** while it significantly uses less computational resources than FA. Note that FA improves over DPA by increasing the number of classifiers, which comes at the cost of a significant increase in training time. Indeed, in some cases on all MNIST, CIFAR-10, and GTSRB datasets, DPA+ROE has improvements over FA while it exploits around **32 times less training cost**. Finally, using a new "boosted DPA" method called DPA* that uses extra training cost to improve the base classifiers of DPA+ROE, we observe improvements of **up to** 3.49%**,** 3.70%**, and** 3.44% **respectively on MNIST, CIFAR-10, and GTSRB datasets**, establishing a new state-of-the-art. [2]

Contributions. In summary, our contributions include:

1. We propose *Run-Off election*, a novel aggregation method for ensemble-based defenses against data poisoning. Our approach is general, provable, and can be applied in combination with different partitioning schemes of the datasets. Using the partitioning schemes in DPA and FA, we propose the DPA+ROE and FA+ROE defense methods.

2. We introduce the notion of 1v1 and 2v1 certificates and show how they can be used to calculate *provable certificates for robustness* for any ensemble method via a reduction. Focusing on DPA+ROE and FA+ROE, we obtain these certificates using careful reasoning on the adversary's optimal action. For each round, we bound the minimum number of poisoned samples the adversary needs. In the first round, we propose a dynamic programming-based approach for characterizing the adversary's action in DPA+ROE and a duality-based approach for bounding the adversary's effect in FA+ROE. In the second round, we carefully bound the minimum number of samples required for electing other classes.

3. We empirically evaluate our method on existing benchmarks. Our experiments show that ROE consistently improves robustness for the DPA, FA, and DPA* methods. In some cases, we improve upon prior work even when using significantly fewer computational resources. Our method establishes the new **state-of-the-art** in certified accuracy against **general data poisoning attacks**.

1.1. Related work

Certified robustness has been widely studied in the literature and prior works have considered various notions of robustness, such as label-flipping (Rosenfeld et al., 2020) and distributional robustness (Lai et al., 2016; Diakonikolas et al., 2016; 2019). Recent works have also studied the poisoning problem theoretically using a PAC learning model (Blum et al., 2021; Gao et al., 2021; Balcan et al., 2022; Hanneke et al., 2022). In this work, we focus on (pointwise) certified robustness against data poisoning and assume a general poisoning model where the adversary may insert, delete, or modify any images.

Most closely related to our work, are the DPA (Levine & Feizi, 2020) and the FA methods (Wang et al., 2022b) that use an ensemble of classifiers to obtain *deterministic* robustness certificates. A similar line of work (Chen et al., 2020; Jia et al., 2021) considers *stochastic* robustness certificates. As mentioned, we improve on these works, establishing a new state-of-the-art for (pointwise) certified robustness. Following prior work, we use *certified fraction*, i.e., the fraction of (test) data points that are certifiable correct, to measure robustness. A similar but slightly different notion is studied by (Chen et al., 2022) who certify the *test accuracy* without certifying any specific data point.

Our work is closely related to the smoothing technique of (Cohen et al., 2019). that has been extensively studied

[2] We thank an anonymous referee for proposing this method. We note that the numbers reported show the difference in accuracy between DPA*+ROE and DPA* itself. Compared to FA, which was the previous state of the art, the improvements can be larger (as high as 27%). Depending on the setting, either DPA*+ROE, or FA+ROE may produce more robust models. We refer to Section 4 for more details.

both in terms of its applications (Raghunathan et al., 2018; Singla & Feizi, 2019; 2020; Chiang et al., 2020) and known limitations (Yang et al., 2020; Kumar et al., 2020; Blum et al., 2020). The original DPA method is inspired by derandomized smoothing (Levine & Feizi, 2020). Smoothing can also be directly applied to data poisoning attacks (Weber et al., 2020), though this requires strong assumptions on the adversary.

2. Preliminaries

Notation. For a positive integer n, we define $[n] := \{1, \ldots, n\}$. Given two arbitrary sets A and B, we use $A \backslash B$ to denote the set of all elements that are in A but not in B and use $A \times B$ to denote the *Cartesian product*, i.e., $A \times B := \{(a, b) : a \in A, b \in B\}$. We use $d_{\text{sym}}(A, B)$ to denote the size of the *symmetric difference* of A and B, i.e.,

$$d_{\text{sym}}(A, B) := |(A \backslash B) \cup (B \backslash A)|.$$

This can be thought of as a measure of distance between A and B and equals the number of insertions and deletions required for transforming A to B.

We use $\mathcal{X}$ to denote the set of all possible unlabeled samples. This is typically the set of all images, though our approach holds for any general input. Similarly, we use $\mathcal{C}$ to be the set of possible labels. We define a *training set* D as any arbitrary collection of labeled samples and let $\mathcal{D} := \mathcal{X} \times \mathcal{C}$.

We define a *classification algorithm* as any mapping $f : \mathcal{D} \times \mathcal{X} \to \mathcal{C}$, where $f(D, x)$ denotes the prediction of the *classifier* trained on the set D and tested on the sample x. We use the notation $f_D(x) := f(D, x)$ for convenience. We assume that the classifier f_D works by first scoring each class and choosing the class with the maximum score. For neural networks, this corresponds to the logits layer of the model. We use $f_D^{\text{logits}}(x, c) \in \mathbb{R}$ to denote the underlying score of class c for the test sample x and assume that $f_D^{\text{logits}}(x, c) \neq f_D^{\text{logits}}(x, c')$ for all $c \neq c'$.

Threat model. We consider a *general poisoning* model where the adversary can poison the training process by adding, removing, or modifying the training set. Given an upper bound on the adversary's budget, i.e., the maximum amount of alteration it can make in the training set, we aim to certify the prediction of the test samples.

Given a classification algorithm f, a dataset D and a test sample x, we define *a prediction certificate* as any provable lower bound on the number of poisoned samples the algorithm can withstand, without changing its prediction. Formally, $\texttt{Cert}$ is a prediction certificate if

$$f_D(x) = f_{D'}(x) \text{ if } d_{\text{sym}}(D, D') \leq \texttt{Cert}.$$

The minimum number of poisoned samples the adversary needs for changing the model's prediction is $\texttt{Cert} + 1$.

3. Proposed method: Run-Off election

In this section, we present our defense approach. We start by discussing *Run-Off election*, an aggregation method that takes as input a test sample x and ensemble of k models $\{f_i\}_{i=1}^k$, and uses these models to make a prediction. The method makes no assumptions about the ensemble and works for an arbitrary choice of the models f_i. In order to obtain certificates, however, we will need to specify the choice of f_i. In Section 3.2, we consider two choices, DPA+ROE and FA+ROE. In Section 3.3, we show how to obtain certificates for these methods.

3.1. Run-Off Election

As mentioned in the introduction, our method can be seen as a *two-round election*, where each model corresponds to a voter, and each class corresponds to a candidate. Given a test sample x, and an ensemble of k models $\{f_i\}_{i=1}^k$, our election consists of the following two rounds.

- **Round 1.** We first obtain the top two classes as measured by the number of models "voting" for each class. Formally, the setting,

$$N_c^{\text{R1}} := \sum_i \mathbb{1}\left[f_i(x) = c\right], \tag{1}$$

  we calculate the top two classes $c_1^{R1} := \arg\max_c N_c^{\text{R1}}$ and $c_2^{R1} := \arg\max_{c \neq c_1^{R1}} N_c^{\text{R1}}$.

- **Round 2.** We collect the votes of each model in an election between c_1^{R1} and c_2^{R1}. Formally, for $(c, c') \in \{(c_1^{R1}, c_2^{R1}), (c_2^{R1}, c_1^{R1})\}$, we set

$$N_c^{\text{R2}} := \sum_{i=1}^k \mathbb{1}\left[f_i^{\text{logits}}(x, c) > f_i^{\text{logits}}(x, c')\right],$$

  and output $\text{ROE}(D, x) := \arg\max_{c \in \{c_1, c_2\}} N_c^{\text{R2}}$.

We assume that $\arg\max$ breaks ties by favoring the class with the smaller index. The formal pseudocode of ROE is provided in Algorithm 1.

Reliability of the logits layer. Our method implicitly assumes that the information in the logits layer of the models is reliable enough to be useful for prediction purposes. This may seem counter-intuitive because many of the models do not even predict one of the top-two classes. In fact, these are precisely the models that were underutilized by DPA. We note however that we only use the logits layer for a binary classification task in Round 2, which easier than multi-class classification. Furthermore, even though the logits layer information may be less reliable than the choice of the top class, it is still useful because it provides *more* information, making the attack problem more difficult for an adversary.

As we will see in Section 4, the clean accuracy of the model may slightly decrease when using our defense, but its robustness improves significantly for larger attack budgets.

3.2. Choice of Ensemble

We now present two possible choices of the ensembles $\{f_i\}_{i=1}$. We begin by considering a disjoint partitioning scheme based on DPA and then consider a more sophisticated overlapping partitioning scheme based on FA. We denote the methods by DPA+ROE and FA+ROE, respectively.

DPA+ROE. In this method, training data is divided into several partitions and a separate base classifier f_i is trained on each of these partitions. Formally, given a hash function $h : \mathcal{X} \to [k]$, the training set D is divided into k partitions $\{D_i\}_{i=1}^k$, where $D_i := \{x \in D : h(x) = i\}$, and the classifiers $\{f_i\}_{i=1}^k$ are obtained by training a base classifier on these partitions, i.e., $f_i := f_{D_i}$. For instance, when classifying images, f_i can be a standard ResNet model trained on D_i.

FA+ROE. In this method, we use two hash functions $h_{\text{spl}} : \mathcal{D} \to [kd]$ and $h_{\text{spr}} : [kd] \to [kd]^d$. We first *split* the datasets into $[kd]$ "buckets" using h_{spl}. We then create kd partitions by *spreading* these buckets, sending each bucket to all of the partitions specified by h_{spr}. Formally, for $i \in [kd]$, we define D_i as $D_i := \{x \in D : i \in h_{\text{spr}}(h_{\text{spl}}(x))\}$. We then train a separate classifier $f_i := f_{D_i}$ for each D_i. A pseudocode of training FA is shown in Algorithm 3.

3.3. Calculating Certificate

Since our aggregation method is more involved than taking a simple majority vote, the adversary can affect the decision-making process in more ways. Calculating the prediction certificate thus requires a more careful argument compared to prior work. In order to present a unified argument, we introduce the concept of *1v1* and *2v1* certificates. A 1v1 certificate bounds the number of poisoned samples required for one class to beat another class while a 2v1 certificate extends this idea and bounds the number of poisoned samples required for *two* classes to beat another class.

We will show that as long as the 1v1 and 2v1 certificates can be calculated efficiently, we can use them to calculate a prediction certificate (Theorem 3.4). The reduction ensures that our approach is general and works for any choice of ensemble, as long as 1v1 and 2v1 certificates can be calculated. We then provide an implementation of how to calculate those certificates for DPA+ROE (Lemmas 3.5 and 3.6) and FA+ROE (Lemmas 3.7 and 3.8).

We begin by defining the notion of the *gap* between two classes.

Definition 3.1. Given an ensemble $\{f_i\}_{i=1}^k$, a sample $x \in \mathcal{X}$, and classes c, c', we define the *gap* between c, c' as

$$\text{gap}(\{f_i\}_{i=1}^k, x, c, c') := N_c - N_{c'} + \mathbb{1}\left[c' > c\right],$$

where $N_c := \sum_i \mathbb{1}\left[f_i(x) = c\right]$ For $\{f_i\}_{i=1}^k$ obtained using the training set D, we use $\text{gap}(D, x, c, c')$ to denote $\text{gap}(\{f_i\}_{i=1}^k, x, c, c')$.

We will omit the dependence of gap on $\{f_i\}_{i=1}^k$ and x when it is clear from context. We say that c *beats* c' if $\text{gap}(c, c') > 0$. If the adversary wants the class c' to beat c, it needs to poison the training set D until $\text{gap}(c, c')$ becomes non-positive. We can therefore use this notion to reason on the optimal behaviour of the adversary.

We define a 1v1 certificate as follows.

Definition 3.2 (1v1 certificate)**.** Given models $\{f_i\}_{i=1}^k$, a test sample x, and two classes $c, c' \in \mathcal{C}$ we say $\texttt{Certv1} \in \mathbb{N}$ is a *1v1 certificate* for c vs c', if for all D' such that $d_{\text{sym}}(D, D') \leq \texttt{Certv1}$, we have $\text{gap}(D', x, c, c') > 0$.

We note that if $\text{gap}(D, x, c, c') \leq 0$, then $\texttt{Certv1}$ is -1.

We similarly define a 2v1 certificate as follows.

Definition 3.3 (2v1 certificate)**.** Given models $\{f_i\}_{i=1}^k$, a test sample x, and three classes $c, c_1, c_2 \in \mathcal{C}$ where $c_1 \neq c_2$ we say $\texttt{Certv2} \in \mathbb{N}$ is a *2v1 certificate* for c vs c_1, c_2 if for all D' such that $d_{\text{sym}}(D, D') \leq \texttt{Certv2}$, we have $\text{gap}(D', x, c, c_1) > 0$ or $\text{gap}(D', x, c, c_2) > 0$.

Assuming these certificates can be calculated efficiently, we can obtain a prediction certificate, as we outline below. Let c^{pred} denote the predicted and c^{sec} the runner-up class. The adversary can change the model's prediction in one of the following two ways.

1. It can eliminate c^{pred} in Round 1. This means it needs to choose two classes $c_1, c_2 \in \mathcal{C} \setminus \{c^{\text{pred}}\}$ and ensure that c_1 and c_2 both have more votes than c^{pred} in Round 1. By definition, this requires more than $\texttt{Certv2}(c^{\text{pred}}, c_1, c_2)$. Since the adversary can choose c_1, c_2, we can lower bound the number of poisoned samples the method can provably tolerate with

$$\texttt{Cert}^{\text{R1}} := \min_{c_1, c_2 \in \mathcal{C} \setminus \{c^{\text{pred}}\}} \texttt{Certv2}(c^{\text{pred}}, c_1, c_2).$$

2. It can eliminate c^{pred} in Round 2. Letting c denote the class that is ultimately chosen, this requires that c makes it to Round 2 and beats c^{pred} in Round 2. For c to make it to Round 2, the adversary needs to ensure that it beats either c^{pred} or c^{sec} in Round 1. Given the previous case, we can assume that c^{pred} makes it to Round 2, which means c needs to beat c^{sec}. We can ensure that this will not happen as long as the number of poisoned samples is at most

$$\texttt{Cert}^{\text{R2}}_{c,1} := \texttt{Certv1}(\{f_i\}_{i=1}^k, c^{\text{sec}}, c).$$

Note that this also includes the special case $c = c^{\text{sec}}$ as $\texttt{Certv1}(c^{\text{sec}}, c^{\text{sec}}) = -1$.

As for c beating c^{pred} in Round 2, let $g_i^c : \mathcal{X} \to \{c, c^{\text{pred}}\}$ denote the binary c^{pred} vs c classifier obtained from f_i. Formally, we set $g_i^c(x)$ to c if $f_i^{\text{logits}}(x, c) > f_i^{\text{logits}}(x, c^{\text{pred}})$ and set it to c^{pred} otherwise. We can lower bound the number of poisoned samples our method can tolerate without changing its prediction with

$$\texttt{Cert}^{\text{R2}}_{c,2} := \texttt{Certv1}(\{g_i^c\}_{i=1}^k, c^{\text{pred}}, c).$$

Overall, since the adversary can choose the class c, we obtain the bound

$$\texttt{Cert}^{\text{R2}} := \min_{c \neq c^{\text{pred}}} \max\{\texttt{Cert}^{\text{R2}}_{c,1}, \texttt{Cert}^{\text{R2}}_{c,2}\}.$$

Given the above analysis, we obtain the following theorem, a formal proof of which is provided in Appendix E.2.

Theorem 3.4 (ROE prediction certificate)**.** *Let c^{pred} denote the prediction of Algorithm 1 after training on a dataset D. For any training set D', if $d_{sym}(D, D') \leq \min\left\{\texttt{Cert}^{\text{R1}}, \texttt{Cert}^{\text{R2}}\right\}$, then Algorithm 1 would still predict c^{pred} when trained on the dataset D'.*

We now show how we can obtain 1v1 and 2v1 certificates for DPA+ROE and FA+ROE.

Certificate for DPA+ROE. We start with calculating $\texttt{Certv1}(c, c')$. Since each poisoned sample can affect at most one model, the optimal action for the adversary is to "flip" a vote from c to c'. The adversary, therefore, requires more than half of the gap between the votes of c and c'. Formally, we use the following lemma, the proof of which is provided in Appendix E.3.1.

Lemma 3.5 (DPA+ROE 1v1 certificate)**.** *Define* $\texttt{Certv1}$ *as* $\texttt{Certv1}(c, c') := \left\lceil \frac{\max\left(0, \text{gap}(c,c')\right)}{2} \right\rceil - 1$. *Then* $\texttt{Certv1}$ *is a 1v1 certificate for DPA+ROE.*

Calculating $\texttt{Certv2}(c, c_1, c_2)$ is more complex as the adversary's optimal action choice is less clear. Intuitively, while the adversary should always change votes from c to either c_1 or c_2, it is not clear which class it should choose. To solve this issue, we use dynamic programming. Defining $\text{gap}_i := \max\{0, \text{gap}(c, c_i)\}$ for $i \in \{1, 2\}$, we calculate $\texttt{Certv2}$ as a function of $\text{gap}_1, \text{gap}_2$. As long as $\text{gap}_1, \text{gap}_2 > 2$, an optimal adversary should first choose a poison to reduce one of the gaps and then continue the poisoning process. This leads to a recursive formulation which we solve efficiently using dynamic programming. Formally, we fill a matrix dp of size $[k+2]^2$ where if $\min(i, j) \geq 2$ we set

$$\text{dp}[i, j] = 1 + \min\{\text{dp}[i-1, j-2], \text{dp}[i-2, j-1]\},$$

and if $\min(i, j) \leq 1$, we set $\text{dp}[i, j] := \left\lceil \frac{\max(i,j)}{2} \right\rceil - 1$. We obtain the following lemma, the proof of which is in Appendix E.3.2.

Lemma 3.6 (DPA+ROE 2v1 certificate)**.** *Define*

$$\texttt{Certv2}(c, c_1, c_2) := \text{dp}[\text{gap}_1, \text{gap}_2],$$

where $\text{gap}_i := \max\{0, \text{gap}(c, c_i)\}$. *Then* $\texttt{Certv2}$ *is a 2v1 certificate for DPA+ROE.*

Certificate for FA+ROE. We start with the 1v1 certificate. Consider a poisoned sample of the adversary and assume it falls in some bucket i. By definition of buckets, this can only affect the models f_j satisfying $j \in h_{\text{spr}}(i)$. If model j votes for c, this can reduce the gap by at most two, and if the model votes for some class $\tilde{c} \notin \{c, c'\}$, it can reduce the gap by 1. This allows us to bound the effect of poisoning each bucket on the gap. As we will see, the effect of poisoning multiple buckets is, at most, the sum of the effects of each bucket. Formally, we obtain the following lemma, the proof of which is in Appendix E.4.1.

Lemma 3.7 (FA+ROE 1v1 certificate)**.** *Given two classes c, c', define the* poisoning power *of each bucket $b \in [kd]$ as*

$$\text{pw}_b := \sum_{i \in h_{spr}(b)} 2\mathbb{1}\left[f_i(x) = c\right] + \mathbb{1}\left[f_i(x) \notin \{c, c'\}\right].$$

Let $\texttt{Certv1}(c, c')$ be the largest number such that the sum of the $\texttt{Certv1}$ largest values in $(\text{pw}_b)_{b \in [kd]}$ is less than $\text{gap}(c, c')$. Then $\texttt{Certv1}$ is a 1v1 certificate.

Formal pseudocode for obtaining $\texttt{Certv1}$ is provided in Algorithm 4.

In order to calculate $\texttt{Certv2}(c, c_1, c_2)$, we first observe that the adversary needs more than $\max_i(\texttt{Certv1}(c, c_i))$ poisoned samples since both c_1 and c_2 need to beat c. This is not necessarily enough, however, as making c_1 and c_2 beat c *simultaneously* may be more difficult than making each beat c_i individually. In order to obtain a stronger bound, we use an approach inspired by duality and consider the conical combination of the constraints. Defining $\text{gap}_i := \max\{0, \text{gap}(c, c_i)\}$, we observe that if gap_1 and gap_2 both become non-positive, then so does every combination $\lambda\text{gap}_1 + \lambda'\text{gap}_1$ for $\lambda, \lambda' \geq 0$. As a special case, this includes $\text{gap}^+ := \text{gap}_1 + \text{gap}_2$. We can bound the number of poisoned samples for making gap^+ non-positive using a similar argument as the 1v1 certificate. Each bucket b can only affect models j such that $j \in h_{\text{spr}}(b)$. If j votes for c_1 or c_2, the gap cannot be reduced. If j votes for c, the gap can be reduced by at most 3 and if j votes for some $\tilde{c} \notin \{c, c_1, c_2\}$, the gap can be reduced by at most 1. We Define the *total poisoning power* of each bucket as

$$\text{pw}_b^+ := \sum_{i \in h_{\text{spr}}(b)} 3\mathbb{1}\left[f_i(x) = c\right] + \mathbb{1}\left[f_i(x) \notin (c, c_1, c_2)\right],$$

where we hide the dependence on c, c_1, c_2 for brevity. We obtain the following lemma, the proof of which is in Appendix E.4.2.

Lemma 3.8 (FA+ROE 2v1 certificate)**.** *For any $c, c_1, c_2 \in \mathcal{C}$, let $\texttt{Certv2}^+$ denote the largest number such that the sum of the $\texttt{Certv2}^+$ largest values in $(\mathrm{pw}_b^+)_{b\in[kd]}$ is less than gap^+. For $i \in \{1, 2\}$, define $\texttt{Certv2}^{(i)} := \texttt{Certv1}(c, c_i)$. Finally, define $\texttt{Certv2}$ as*

$$\texttt{Certv2} := \max\{\texttt{Certv2}^{(1)}, \texttt{Certv2}^{(2)}, \texttt{Certv2}^+\}.$$

Then $\texttt{Certv2}$ is a 2v1 certificate for FA+ROE.

4. Evaluation

In this section, we empirically analyze our method and demonstrate that it reaches state-of-the-art results in certified robustness. In some cases, this comes with considerably less computation than the baseline.

4.1. Experimental Setting

We consider the same setup as prior work (Levine & Feizi, 2020; Wang et al., 2022b) and evaluate our method on MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky et al., 2009) and GTSRB (Stallkamp et al., 2012) datasets. We similarly use Network-In-Network (Lin et al., 2013) architecture, to be trained with the set of hyperparameters from (Gidaris et al., 2018). Wang et al. (2022a) observe that the accuracy of ensemble methods can be further improved by having better base classifiers, i.e., base classifiers that have better classification accuracy. They improve over the original DPA by training base classifiers on the augmented version of datasets. As we want to have a fair comparison to the FA baseline, we train classifiers of both DPA and FA as Wang et al. (2022b).

As in prior work (Wang et al., 2022b), we consider *certified fraction* (CF) as our performance metric. Given a budget B for the adversary, the certified fraction of the dataset denotes the fraction of test samples that are *provably* classified correctly as long as the dataset is not altered by more than B points. We note that some works (e.g., Levine & Feizi (2020)) use the term *certified accuracy* for this metric.

As baselines, we use the Deep Partition Aggregation (DPA) method of Levine & Feizi (2020) and the Finite Aggregation (FA) method of Wang et al. (2022b). As discussed in Wang et al. (2022b), FA is effectively a generalization of DPA that uses overlapping partitions. Compared to DPA, FA takes an additional parameter d and uses d times as many base models. When using $d = 1$, FA coincides with DPA. While larger values of d increase the robustness of the model, this comes at the cost of increased computation; the training cost for FA is d times the training cost for DPA.

Boosted DPA. Given the increased computational cost of FA, in order to obtain a fair comparison, we also consider a "boosted" variant of DPA which we denote by DPA*.[3] For each partition D_i of the dataset, we train d models $\{f_{i,j}\}_{j=1}^d$ on the partition using different seed values. At test time, we average the logits layer of these models. In other words, each model f_i in DPA*is itself an ensemble of d "submodels". The approach effectively makes the predictions more robust to the noisiness of the training process. The certificate for the model is calculated using the same technique as the certificate for DPA as outlined in Section 3.3. We note that the case $d = 1$ corresponds to DPA.

4.2. Results

Our results are shown in Tables 1, 2, 3, 4, 5 and Figures 2, 3. We do not report error bars as the variation caused by training noise is negligible (See Appendix D). As seen in Table 1, when we apply our aggregation method to FA, it can remarkably improve the certified accuracy of the original Finite Aggregation (we compare these two methods with the same d). Improvements can be up to 3% or 4%. More interestingly, by applying our technique to DPA*, we observe significant improvement which can be up tp 27% in some cases. This implies that DPA*+ROE is the new **state-of-the-art** in provable defense.

Overall, the results show that no matter choice of base classifiers is, **applying ROE improves the certified robustness**. The results of applying ROE to DPA can be seen in Table 3. We obtain improvements in certified accuracy by up to $4.73\%, 3.14\%$, and 3.18% on MNIST, CIFAR-10, and GTSRB, respectively. Improvements of using ROE on DPA* is reported in Table 2. DPA*+ROE improves robustness of DPA* on MNIST, CIFAR-10, and GTSRB by up to $3.49\%, 3.70\%$, and 3.44%, respectively.

Perhaps more impressively, as seen in Table 4, DPA+ROE also competes with, and for larger values of B outperforms FA while it significantly needs less training cost as its training cost is equivalent to that of DPA. For example, on CIFAR-10 dataset when $k = 250$, by using a single NVIDIA GeForce RTX 2080 Ti GPU, the total training time of classifiers used in DPA+ROE is around 3 hours while it takes around 47.3 hours to train classifiers needed in FA with $d = 16$. Roughly speaking, the training of FA with parameter d, takes d time more than that of DPA, or DPA+ROE.

Although DPA+ROE uses less training time, it obtains a higher certified accuracy for larger values of B, e.g., on the standard CIFAR-10 dataset when $k = 50$, it obtains a higher certified fraction than FA with $d = 32$ when $B \geq 15$ **even though it uses 32 times less computation in training**. A similar comparison of DPA+ROE and DPA* in

[3] We thank an anonymous referee of the paper for proposing this method.

Table 1. Certified fraction of DPA*+ROE, FA+ROE, and original FA with various values of hyperparameter d with respect to different attack sizes B. Improvements of DPA*+ROE and FA+ROE over the original FA (with same d) are highlighted in blue if they are positive and red otherwise.

dataset	k	method	d	certified fraction				
MNIST	1200			$B \leq 100$	$B \leq 200$	$B \leq 300$	$B \leq 400$	$B \leq 500$
		FA	16	92.75%	87.89%	78.91%	62.42%	31.97%
		FA+ROE		92.80%(+0.05%)	88.09%(+0.2%)	80.26%(+1.35%)	65.31%(+2.89%)	36.76%(+4.79%)
		DPA*+ROE		92.70%(-0.05%)	88.41%(+0.52%)	81.75%(+2.84%)	69.67%(+7.25%)	44.81%(+12.84%)
CIFAR-10	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 18$	$B \leq 20$
		FA	16	60.55%	48.85%	34.61%	25.46%	19.90%
		FA+ROE		61.71%(+1.16%)	51.18%(+2.33%)	37.12%(+2.51%)	28.49%(+3.03%)	22.08%(+2.18%)
		DPA*+ROE		61.87%(+1.32%)	52.71%(+3.86%)	41.51%(+6.9%)	33.42%(+7.96%)	27.47%(+7.57%)
		FA	32	61.31%	50.31%	36.03%	26.55%	19.93%
		FA+ROE		62.56%(+1.25%)	52.55%(+2.24%)	38.83%(+2.8%)	29.05%(+2.5%)	21.97%(+2.04%)
		DPA*+ROE		65.99%(+4.68%)	61.51%(+11.2%)	56.13%(+20.1%)	51.83%(+25.28%)	47.61%(+27.68%)
	250			$B \leq 10$	$B \leq 20$	$B \leq 40$	$B \leq 50$	$B \leq 60$
		FA	8	45.38%	36.05%	20.08%	14.39%	9.70%
		FA+ROE		46.80%(+1.42%)	38.56%(+2.51%)	23.61%(+3.53%)	17.86%(+3.47%)	13.06%(+3.36%)
		DPA*+ROE		47.14%(+1.76%)	39.32%(+3.27%)	25.41%(+5.33%)	19.68%(+5.29%)	15.02%(+5.32%)
		FA	16	46.52%	37.56%	21.99%	15.79%	11.09%
		FA+ROE		48.33%(+1.81%)	40.71%(+3.15%)	26.38%(+4.39%)	20.52%(+4.73%)	14.64%(+3.55%)
		DPA*+ROE		46.88%(+0.36%)	39.50%(+1.94%)	25.49%(+3.5%)	19.83%(+4.04%)	15.04%(+3.95%)
GTSRB	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 20$	$B \leq 22$
		FA	16	82.71%	74.66%	63.77%	47.52%	35.54%
		FA+ROE		82.59%(–0.12%)	75.55%(+0.89%)	65.47%(+1.7%)	50.33%(+2.81%)	38.89%(+3.35%)
		DPA*+ROE		83.67%(+0.96%)	77.84%(+3.18%)	70.63%(+6.86%)	57.97%(+10.45%)	49.33%(+13.79%)
		FA	32	83.52%	76.26%	66.32%	49.68%	38.31%
		FA+ROE		83.61%(+0.1%)	77.07%(+0.81%)	67.83%(+1.51%)	51.81%(+2.14%)	41.61%(+3.29%)
		DPA*+ROE		83.67%(+0.15%)	77.93%(+1.66%)	70.67%(+4.35%)	58.38%(+8.71%)	49.61%(+11.3%)
	100			$B \leq 5$	$B \leq 15$	$B \leq 20$	$B \leq 25$	$B \leq 30$
		FA	16	48.19%	33.95%	25.96%	18.92%	13.82%
		FA+ROE		48.00%(–0.19%)	35.76%(+1.81%)	28.92%(+2.95%)	22.30%(+3.38%)	16.32%(+2.49%)
		DPA*+ROE		47.50%(–0.69%)	36.48%(+2.53%)	30.60%(+4.64%)	24.63%(+5.71%)	19.26%(+5.44%)
		FA	32	48.39%	34.96%	27.05%	19.83%	14.47%
		FA+ROE		48.15%(–0.25%)	36.81%(+1.84%)	29.85%(+2.8%)	23.37%(+3.54%)	17.41%(+2.95%)
		DPA*+ROE		47.66%(–0.74%)	36.66%(+1.69%)	30.70%(+3.66%)	24.71%(+4.88%)	19.33%(+4.87%)

Table 5 shows that in some cases, DPA+ROE can outperform DPA*as well while it uses significantly less training cost.

The increased accuracy of the models depends on the setting; one may choose DPA+ROE, FA+ROE or DPA*+ROE as each have their advantages. DPA+ROE has the advantage that it uses less computational resources while DPA*+ROE and FA+ROE obtain better accuracy. We note that the benefit of DPA*+ROE and FA+ROE compared to DPA+ROE seem to come from two different sources; DPA*+ROE uses increased computation to make each model more robust while FA+ROE uses a more complex partitioning scheme. As such, it can be expected that either of the methods may be preferable depending on the setting and this is observed in our experiments, though generally DPA*+ROE seems to have higher accuracy. Visual comparison of different methods can be seen in Figures 2 and 3. While these distinctions are interesting, they are orthogonal to the focus of our paper; we show that using ROE improves robustness in all of these settings without increasing the training cost.

Effect of the budget B. Our results show that ROE methods are especially useful for larger values of the adversary's budget B. Intuitively, ROE is utilizing base classifiers that were previously discarded. As such, for a fixed budget B, *the ratio* of the poisoned samples to the utilized models is considerably smaller for our method, which allows us to obtain improved results. We note that this is in strong contrast to FA, where for larger values of B, the accuracy gains compared to DPA diminish and eventually cease to exist. Indeed, as seen in Figure 2, FA can actually be *worse* than DPA for large budgets, while our method remains strongly favorable, as we achieve 5% higher certified fraction on the standard CIFAR-10 dataset.

While our aggregation method performs well when the adversary's budget is high, we see a slightly lower certified fraction in Figures 2 and 3 when B is relatively small. In these cases, the certified fraction is close to clean accuracy, i.e., accuracy of the model when training data is not poisoned. Intuitively, the drop is because of the (slightly) lower reliability of the logits-layer information as discussed in

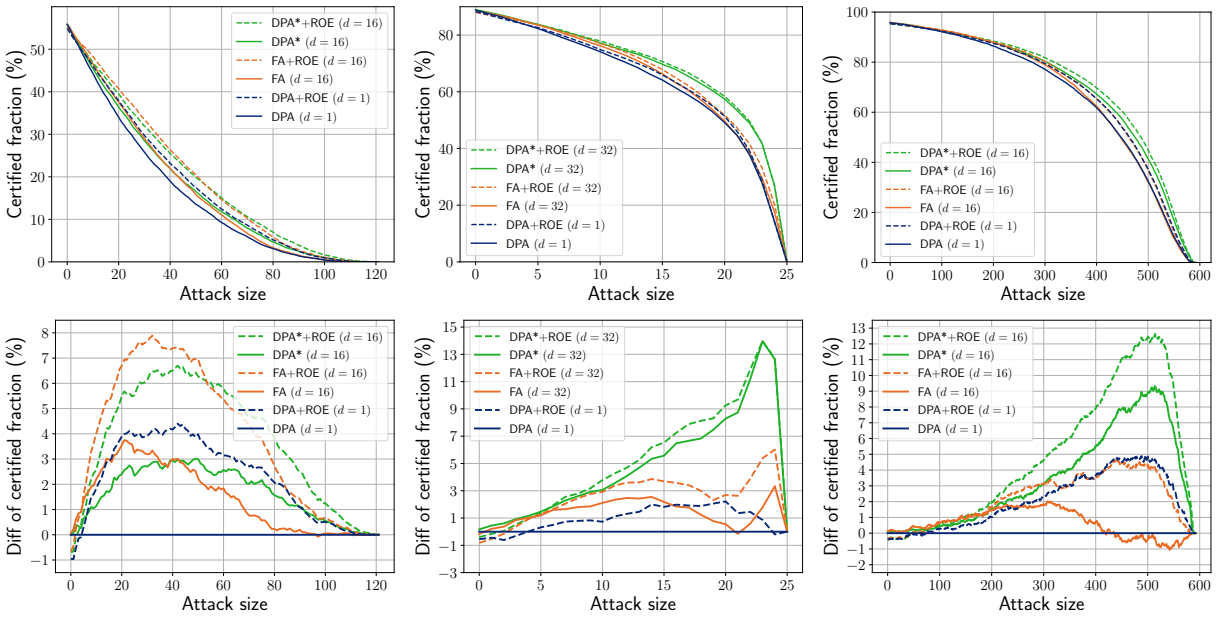

Figure 2. **First row**: The curves of certified fraction of different methods on different datasets. **Second row**: The improvements of certified fraction over DPA. Plots in the first columns refers to CIFAR-10 ($k = 250$), plots in the second column refers to GRSTB ($k = 50$), and plots in the last column corresponds to MNIST ($k = 1200$). Note that training cost of different methods scales up with d, i.e., **training of FA, FA+ROE, DPA*, and DPA*+ROE with parameter** d takes roughly d **times more than that of DPA or DPA+ROE**. When the adversary's budget is large, **DPA+ROE outperforms FA** while it significantly exploits less training cost. In some case, DPA+ROE can outperform DPA* as well.

Section 3.1. ROE methods have slightly lower clean accuracy because they involve all models in prediction, even models which are not accurate for a sample test. On the other hand, when the model's prediction is correct, involving more models makes the adversary's task harder.

5. Conclusion

In this paper, we introduced Run-Off Election (ROE), a new aggregation method for ensemble-based defenses against data poisoning. We proposed a novel two-stage election across the base models of the ensemble that utilizes all of the models in order to increase the prediction gap between the top and runner-up classes. We developed a unified framework for calculating certificates for our method and proposed three new defense methods – DPA+ROE, FA+ROE, and DPA*+ROE – based on prior work and a new DPA* method proposed by an anonymous reviewer. We evaluated our methods on standard benchmarks and observed improved poisoning certificates while simultaneously reducing the training cost. Our method established a new state-of-the-art in provable defense against general data poisoning in several datasets.

For future work, it would be interesting to extend our methodology to other ensemble methods including the ones producing stochastic certificates. As discussed in Section 3, in principle, ROE can be applied on top of any ensemble method, though it is not immediately clear how one can obtain prediction certificates for such hybrid models. We hope that our unified approach to calculating certificates in Section 3.3, can be helpful for other ensemble methods as well.

Another interesting direction is to explore more complex aggregation mechanisms, such as an N-round election for $N > 2$. Two potential challenges for designing such methods are certificate calculation and potential decreased accuracy due to unreliable logits-layer information. These challenges also appear in our work; our method for calculating certificates is more involved than prior work and in some settings our models have lower clean accuracy. We hope that the techniques proposed in our paper can help address these challenges for more complex mechanisms as well.

6. Acknowledgements

The authors thank Wenxiao Wang for helpful discussions throughout the project. This project was supported in part by NSF CAREER AWARD 1942230, a grant from NIST 60NANB20D134, HR001119S0026 (GARD), ONR YIP award N00014-22-1-2271, Army Grant No. W911NF2120076 and the NSF award CCF2212458.

References

Balcan, M.-F., Blum, A., Hanneke, S., and Sharma, D. Robustly-reliable learners under poisoning attacks. *arXiv preprint arXiv:2203.04160*, 2022.

Blum, A., Dick, T., Manoj, N., and Zhang, H. Random smoothing might be unable to certify l_∞ robustness for high-dimensional images. *J. Mach. Learn. Res.*, 21:211–1, 2020.

Blum, A., Hanneke, S., Qian, J., and Shao, H. Robust learning under clean-label attack. In *Conference on Learning Theory*, pp. 591–634. PMLR, 2021.

Borgnia, E., Cherepanova, V., Fowl, L., Ghiasi, A., Geiping, J., Goldblum, M., Goldstein, T., and Gupta, A. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3855–3859. IEEE, 2021.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Chen, R., Li, J., Wu, C., Sheng, B., and Li, P. A framework of randomized selection based certified defenses against data poisoning attacks. *arXiv preprint arXiv:2009.08739*, 2020.

Chen, R., Li, Z., Li, J., Yan, J., and Wu, C. On collective robustness of bagging against data poisoning. In *International Conference on Machine Learning*, pp. 3299–3319. PMLR, 2022.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Chen, Y. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo, 2015.

Chiang, P.-y., Ni, R., Abdelkader, A., Zhu, C., Studer, C., and Goldstein, T. Certified defenses for adversarial patches. *arXiv preprint arXiv:2003.06693*, 2020.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J. Z., Moitra, A., and Stewart, A. Robust estimators in high dimensions without the computational intractability. *CoRR*, abs/1604.06443, 2016. URL http://arxiv.org/abs/1604.06443.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Steinhardt, J., and Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pp. 1596–1606. PMLR, 2019.

Gao, J., Karbasi, A., and Mahmoody, M. Learning and certification under instance-targeted poisoning. In *Uncertainty in Artificial Intelligence*, pp. 2135–2145. PMLR, 2021.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Gong, C., Ren, T., Ye, M., and Liu, Q. Maxup: A simple way to improve generalization of neural network training. *arXiv preprint arXiv:2002.09024*, 2020.

Hanneke, S., Karbasi, A., Mahmoody, M., Mehalel, I., and Moran, S. On optimal learning under targeted data poisoning. *arXiv preprint arXiv:2210.02713*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hodge, V. and Austin, J. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

Jia, J., Cao, X., and Gong, N. Z. Intrinsic certified robustness of bagging against data poisoning attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7961–7969, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kumar, A., Levine, A., Goldstein, T., and Feizi, S. Curse of dimensionality on randomized smoothing for certifiable robustness. In *International Conference on Machine Learning*, pp. 5458–5467. PMLR, 2020.

Lai, K. A., Rao, A. B., and Vempala, S. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 665–674. IEEE, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Levine, A. and Feizi, S. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*, 2020.

Lin, M., Chen, Q., and Yan, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Ni, R., Goldblum, M., Sharaf, A., Kong, K., and Goldstein, T. Data augmentation for meta-learning. In *International Conference on Machine Learning*, pp. 8152–8161. PMLR, 2021.

Paudice, A., Muñoz-González, L., and Lupu, E. C. Label sanitization against label flipping poisoning attacks. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 5–15. Springer, 2018.

Raghunathan, A., Steinhardt, J., and Liang, P. S. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems*, 31, 2018.

Rosenfeld, E., Winston, E., Ravikumar, P., and Kolter, Z. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pp. 8230–8241. PMLR, 2020.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Singla, S. and Feizi, S. Robustness certificates against adversarial examples for relu networks. *arXiv preprint arXiv:1902.01235*, 2019.

Singla, S. and Feizi, S. Second-order provable defenses against adversarial attacks. In *International conference on machine learning*, pp. 8981–8991. PMLR, 2020.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32: 323–332, 2012.

Wang, W., Levine, A., and Feizi, S. Lethal dose conjecture on data poisoning. *arXiv preprint arXiv:2208.03309*, 2022a.

Wang, W., Levine, A. J., and Feizi, S. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *International Conference on Machine Learning*, pp. 22769–22783. PMLR, 2022b.

Weber, M., Xu, X., Karlaš, B., Zhang, C., and Li, B. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*, 2020.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020.

Table 2. Certified fraction of DPA*+ROE, and DPA* with various values for hyperparameter d, with respect to different attack sizes B. Improvements over the DPA* baseline are highlighted in blue if they are positive and red otherwise.

dataset	k	method	d	certified fraction				
MNIST	1200			$B \leq 100$	$B \leq 200$	$B \leq 300$	$B \leq 400$	$B \leq 500$
		DPA*	16	92.55%	87.99%	80.23%	67.25%	41.32%
		DPA*+ROE		92.70%(+0.15%)	88.41%(+0.42%)	81.75%(+1.52%)	69.67%(+2.42%)	44.81%(+3.49%)
CIFAR-10	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 18$	$B \leq 20$
		DPA*	16	61.00%	50.94%	39.29%	31.41%	25.97%
		DPA*+ROE		61.87%(+0.87%)	52.71%(+1.77%)	41.51%(+2.22%)	33.42%(+2.01%)	27.47%(+1.5%)
		DPA*	32	65.77%	60.89%	55.53%	51.05%	46.95%
		DPA*+ROE		65.99%(+0.22%)	61.51%(+0.62%)	56.13%(+0.6%)	51.83%(+0.78%)	47.61%(+0.66%)
	250			$B \leq 10$	$B \leq 20$	$B \leq 40$	$B \leq 50$	$B \leq 60$
		DPA*	8	45.36%	36.34%	21.71%	16.41%	11.60%
		DPA*+ROE		47.14%(+1.78%)	39.32%(+2.98%)	25.41%(+3.7%)	19.68%(+3.27%)	15.02%(+3.42%)
		DPA*	16	45.45%	36.41%	21.93%	16.55%	11.82%
		DPA*+ROE		46.88%(+1.43%)	39.50%(+3.09%)	25.49%(+3.56%)	19.83%(+3.28%)	15.04%(+3.22%)
GTSRB	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 20$	$B \leq 22$
		DPA*	16	83.71%	77.22%	69.70%	56.71%	48.61%
		DPA*+ROE		83.67%(–0.04%)	77.84%(+0.62%)	70.63%(+0.93%)	57.97%(+1.26%)	49.33%(+0.72%)
		DPA*	32	83.79%	77.21%	69.71%	57.41%	48.91%
		DPA*+ROE		83.67%(–0.13%)	77.93%(+0.71%)	70.67%(+0.97%)	58.38%(+0.97%)	49.61%(+0.7%)
	100			$B \leq 5$	$B \leq 15$	$B \leq 20$	$B \leq 25$	$B \leq 30$
		DPA*	16	47.64%	34.73%	27.50%	21.20%	16.19%
		DPA*+ROE		47.50%(–0.14%)	36.48%(+1.74%)	30.60%(+3.1%)	24.63%(+3.44%)	19.26%(+3.07%)
		DPA*	32	47.82%	34.81%	27.90%	21.51%	16.34%
		DPA*+ROE		47.66%(–0.17%)	36.66%(+1.84%)	30.70%(+2.8%)	24.71%(+3.2%)	19.33%(+2.99%)

A. Code

Our code can be found in this github repository.

B. Figures and Tables

In this section, we provide Tables 3, 4, and 5. Figure 3 is also depicted here.

Table 3. Certified fraction of DPA+ROE, and original DPA with respect to different attack sizes B. Improvements over the DPA baseline are highlighted in blue if they are positive and red otherwise.

dataset	k	method	certified fraction				
MNIST	1200		$B \leq 100$	$B \leq 200$	$B \leq 300$	$B \leq 400$	$B \leq 500$
		DPA	92.11%	86.45%	77.12%	61.78%	32.42%
		DPA+ROE	92.38%(+0.27%)	87.46%(+1.01%)	79.43%(+2.31%)	65.42%(+3.64%)	37.15%(+4.73%)
CIFAR-10	50		$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 18$	$B \leq 20$
		DPA	58.07%	46.44%	33.46%	24.87%	19.36%
		DPA+ROE	59.80%(+1.73%)	49.09%(+2.65%)	36.04%(+2.58%)	27.52%(+2.65%)	21.30%(+1.94%)
	250		$B \leq 10$	$B \leq 20$	$B \leq 40$	$B \leq 50$	$B \leq 60$
		DPA	44.31%	34.01%	18.99%	13.55%	9.25%
		DPA+ROE	46.14%(+1.83%)	37.90%(+3.89%)	23.16%(+4.17%)	17.53%(+3.98%)	12.39%(+3.14%)
GTSRB	50		$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 20$	$B \leq 22$
		DPA	82.32%	74.15%	64.14%	49.12%	37.73%
		DPA+ROE	82.64%(+0.32%)	74.88%(+0.73%)	65.96%(+1.82%)	51.34%(+2.22%)	39.18%(+1.45%)
	100		$B \leq 5$	$B \leq 15$	$B \leq 20$	$B \leq 25$	$B \leq 30$
		DPA	46.16%	30.19%	22.84%	17.16%	12.75%
		DPA+ROE	46.09%(–0.07%)	33.45%(+3.26%)	26.86%(+4.02%)	21.02%(+3.86%)	15.93%(+3.18%)

Table 4. Certified fraction of DPA+ROE, and original FA with various values of hyperparameter d with respect to different attack sizes B. Improvements of DPA+ROE compared to the original FA with different values of d are highlighted in blue if they are positive and red otherwise. **Note that FA with parameter d uses d times as many as classifiers than DPA+ROE. Training FA classifiers therefore takes d times more that that of DPA+ROE.**

dataset	k	method	d	certified fraction				
MNIST	1200			$B \leq 100$	$B \leq 200$	$B \leq 300$	$B \leq 400$	$B \leq 500$
		FA	16	92.75%	87.89%	78.91%	62.42%	31.97%
		DPA+ROE		92.38%(–0.37%)	87.46%(–0.43%)	79.43%(+0.52%)	65.42%(+3.00%)	37.15%(+5.18%)
		FA	32	92.97%	88.49%	80.17%	64.34%	31.09%
		DPA+ROE		92.38%(–0.59%)	87.46%(–1.03%)	79.43%(–0.74%)	65.42%(+1.08%)	37.15%(+6.06%)
CIFAR-10	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 18$	$B \leq 20$
		FA	16	60.55%	48.85%	34.61%	25.46%	19.90%
		DPA+ROE		59.80%(–0.75%)	49.09%(+0.24%)	36.04%(+1.43%)	27.52%(+2.06%)	21.30%(+1.40%)
		FA	32	61.31%	50.31%	36.03%	26.55%	19.93%
		DPA+ROE		59.80%(–1.51%)	49.09%(–1.22%)	36.04%(+0.01%)	27.52%(+0.97%)	21.30%(+1.37%)
	250			$B \leq 10$	$B \leq 20$	$B \leq 40$	$B \leq 50$	$B \leq 60$
		FA	8	45.38%	36.05%	20.08%	14.39%	9.70%
		DPA+ROE		46.14%(+0.76%)	37.90%(+1.85%)	23.16%(+3.08%)	17.53%(+3.14%)	12.39%(+2.69%)
		FA	16	46.52%	37.56%	21.99%	15.79%	11.09%
		DPA+ROE		46.14%(–0.38%)	37.90%(+0.34%)	23.16%(+1.17%)	17.53%(+1.74%)	12.39%(+1.30%)
GTSRB	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 20$	$B \leq 22$
		FA	16	82.71%	74.66%	63.77%	47.52%	35.54%
		DPA+ROE		82.64%(–0.07%)	74.88%(+0.22%)	65.96%(+2.19%)	51.34%(+3.82%)	39.18%(+3.63%)
		FA	32	83.52%	76.26%	66.32%	49.68%	38.31%
		DPA+ROE		82.64%(–0.88%)	74.88%(–1.39%)	65.96%(–0.36%)	51.34%(+1.66%)	39.18%(+0.86%)
	100			$B \leq 5$	$B \leq 15$	$B \leq 20$	$B \leq 25$	$B \leq 30$
		FA	16	48.19%	33.95%	25.96%	18.92%	13.82%
		DPA+ROE		46.09%(–2.10%)	33.45%(–0.50%)	26.86%(+0.90%)	21.02%(+2.10%)	15.93%(+2.11%)
		FA	32	48.39%	34.96%	27.05%	19.83%	14.47%
		DPA+ROE		46.09%(–2.30%)	33.45%(–1.51%)	26.86%(–0.18%)	21.02%(+1.19%)	15.93%(+1.46%)

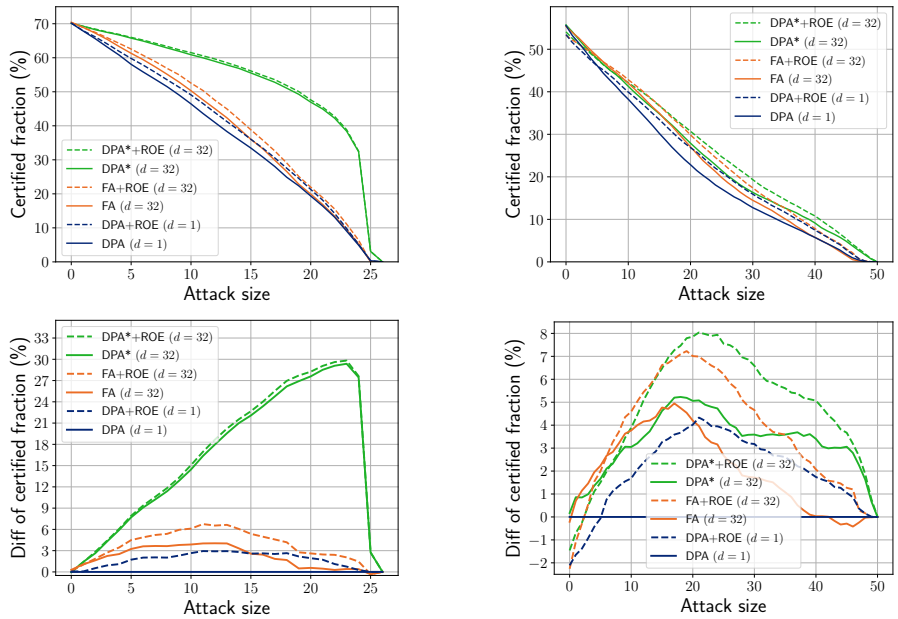

Figure 3. **First row**: The curves of certified fraction of different methods on different datasets. **Second row**: The improvements of certified fraction over DPA. Plots in the first columns refers to CIFAR-10 ($k = 50$), plots in the second column refers to GRSTB ($k = 100$).

Table 5. Certified fraction of DPA+ROE, and DPA* with various values of hyperparameter d with respect to different attack sizes B. Improvements of DPA+ROE compared to the DPA* with different values of d are highlighted in blue if they are positive and red otherwise. **Note that DPA* with parameter d uses d times as many as classifiers than DPA+ROE. Training DPA* classifiers therefore takes d times more that that of DPA+ROE.**

dataset	k	method	d	certified fraction				
MNIST	1200			$B \leq 100$	$B \leq 200$	$B \leq 300$	$B \leq 400$	$B \leq 500$
		DPA*	16	92.55%	87.99%	80.23%	67.25%	41.32%
		DPA+ROE		92.38%(–0.17%)	87.46%(–0.53%)	79.43%(–0.80%)	65.42%(-1.83%)	37.15%(–4.17%)
CIFAR-10	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 18$	$B \leq 20$
		DPA*	16	61.00%	50.94%	39.29%	31.41%	25.97%
		DPA+ROE		59.80%(–1.20%)	49.09%(–1.85%)	36.04%(–3.25%)	27.52%(–3.89%)	21.30%(–4.67%)
		DPA*	32	65.77%	60.89%	55.53%	51.05%	46.95%
		DPA+ROE		59.80%(–5.97%)	49.09%(–11.80%)	36.04%(–19.49%)	27.52%(–23.53%)	21.30%(–25.65%)
	250			$B \leq 10$	$B \leq 20$	$B \leq 40$	$B \leq 50$	$B \leq 60$
		DPA*	8	45.36%	36.34%	21.71%	16.41%	11.60%
		DPA+ROE		46.14%(+0.78%)	37.90%(+1.56%)	23.16%(+1.45%)	17.53%(+1.12%)	12.39%(+0.79%)
		DPA*	16	45.45%	36.41%	21.93%	16.55%	11.82%
		DPA+ROE		46.14%(+0.69%)	37.90%(+1.49%)	23.16%(+1.23%)	17.53%(+0.98%)	12.39%(+0.57%)
GTSRB	50			$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 20$	$B \leq 22$
		DPA*	16	83.71%	77.22%	69.70%	56.71%	48.61%
		DPA+ROE		82.64%(–1.07%)	74.88%(–2.34%)	65.96%(–3.74%)	51.34%(–5.38%)	39.18%(–9.44%)
		DPA*	32	83.79%	77.21%	69.71%	57.41%	48.91%
		DPA+ROE		82.64%(–1.16%)	74.88%(–2.34%)	65.96%(–3.75%)	51.34%(–6.07%)	39.18%(–9.73%)
	100			$B \leq 5$	$B \leq 15$	$B \leq 20$	$B \leq 25$	$B \leq 30$
		DPA*	16	47.64%	34.73%	27.50%	21.20%	16.19%
		DPA+ROE		46.09%(–1.55%)	33.45%(–1.28%)	26.86%(–0.63%)	21.02%(–0.17%)	15.93%(–0.26%)
		DPA*	32	47.82%	34.81%	27.90%	21.51%	16.34%
		DPA+ROE		46.09%(–1.73%)	33.45%(–1.36%)	26.86%(–1.04%)	21.02%(–0.49%)	15.93%(–0.41%)

C. Pseudocodes

In this section, we provide pseudocodes that were omitted from the main text.

Algorithm 1 ROE algorithm

Input: Trianed classifiers $\{f_i\}_{i=1}^k$, test sample $x \in \mathcal{X}$.
Output: Prediction of Run-off election for x.

function PREDICT($\{f_i\}_{i=1}^k, x$)
  /* Round 1 */
  for $c \in \mathcal{C}$ **do**
    $N_c^{\text{R1}} \leftarrow \sum_{i=1}^k \mathbb{1}\left[f_i(x) = c\right]$
  $c_1 \leftarrow \arg\max_c N_c^{\text{R1}}, \quad c_2 \leftarrow \arg\max_{c \neq c_1} N_c^{\text{R1}}$.
  /* Round 2 */
  $N_{c_1}^{\text{R2}} \leftarrow \sum_{i=1}^k \mathbb{1}\left[f_i^{\text{logits}}(x, c_1) > f_i^{\text{logits}}(x, c_2)\right]$
  $N_{c_2}^{\text{R2}} \leftarrow k - N_{c_2}^{\text{R2}}$
  return $\arg\max_{c \in \{c_1, c_2\}} N_c^{\text{R2}}$

Algorithm 2 Training of classifiers in DPA+ROE

Input: Dataset $D \subseteq \mathcal{X} \times \mathcal{C}$, hash function $h : \mathcal{X} \to [k]$, test sample $x \in \mathcal{X}$.
Output: Prediction of DPA for x.

function TRAIN(D, h)
  for $i \leftarrow 1, \ldots, k$ **do**
    $D_i \leftarrow \{x \in D : h(x) = i\}, \quad f_i \leftarrow f_{D_i}$
  return $\{f_i\}_{i=1}^k$

Algorithm 3 Training of classifiers in FA+ROE

Input: Dataset $D \subseteq \mathcal{X} \times \mathcal{C}$, hash functions $h_{\text{spl}} : \mathcal{D} \to [kd], h_{\text{spr}} : [kd] \to [kd]^d$.
Output: Trained $\{f_i\}_{i=1}^{kd}$ classifiers.

function TRAIN($D, h_{\text{spl}}, h_{\text{spr}}$)
  for $i \leftarrow 1, \ldots, kd$ **do**
    $D_i \leftarrow \{x : i \in h_{\text{spr}}(h_{\text{spl}}(x))\}, \quad f_i \leftarrow f_{D_i}$
  return $\{f_i\}_{i=1}^{kd}$

Algorithm 4 `Certv1` algorithm for FA+ROE

Input: Array pw of size kd, an integer gap.
Output: Minimum number of poisoned samples needed to make the gap non-positive.

function CERTFA(pw, gap)
  Sort array pw in decreasing order
  count $\leftarrow -1$
  while gap> 0 **do**
    gap$\leftarrow$ gap$-$pw$_{\text{count}}$
    count $\leftarrow$ count $+ 1$
  return count

D. Effect of lucky seed

We note that everything in DPA, FA, DPA*, and our method is deterministic, so when base classifiers are fixed, then all certificates are deterministic. Same as existing work, we evaluated our method in as many different settings as possible, i.e.,

Table 6. Average certified accuracy of DPA, FA, DPA+ROE, and FA+ROE on CIFAR-10 with $k = 50$ partitions. Results of DPA and DPA+ROE are averaged over 16 **trials** and results of FA and FA+ROE are reported when $d = 16$ over 4 **trials.** (certified accuracy is reported in the form of mean $\pm$ std)

average of certified fraction over multiple trials					
method	$B \leq 5$	$B \leq 10$	$B \leq 15$	$B \leq 18$	$B \leq 22$
DPA	58.63% ($\pm$0.20%)	46.45% ($\pm$0.20%)	33.43% ($\pm$0.21%)	25.26% ($\pm$0.17%)	19.54% ($\pm$0.22%)
DPA+ROE	60.00% ($\pm$0.19%)	49.14% ($\pm$0.21%)	36.34% ($\pm$0.22%)	27.65% ($\pm$0.22%)	21.49% ($\pm$0.22%)
FA	60.21% ($\pm$0.25%)	48.49% ($\pm$0.26%)	34.27% ($\pm$0.24%)	25.44% ($\pm$0.12%)	19.92% ($\pm$0.06%)
FA+ROE	61.48% ($\pm$0.18%)	50.90% ($\pm$0.20%)	37.16% ($\pm$0.03%)	28.45% ($\pm$0.11%)	22.07% ($\pm$0.09%)

we evaluated our method on different datasets, different values for the number of partitions (k), and different values of d. In our experiments, we have noticed the error bars are very small (thus we focused more on different settings instead of repeating experiments). To show this empirically, we run multiple trials on the CIFAR-10 dataset with $k = 50$, on both DPA and FA. Results can be seen in Table 6. Error bars are negligible.

E. Proofs

In this section, we first provide some basic lemmas. Secondly, we prove Theorem 3.4. Then, we analyze how to calculate the certificate in DPA+ROE and FA+ROE by proving lemmas 3.5, 3.6, 3.7, 3.8 and another lemma which is provided later.

E.1. Preliminaries

Lemma E.1. *For any two different classes $c_1, c_2 \in \mathcal{C}$, c_1 beats c_2 if and only if* $\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2) > 0$.

Proof. Let N_c denote the number of votes that class c has, i.e., $N_c = \sum_{i=1}^k \mathbb{1}\left[f_i(x) = c\right]$. If c_1 beats c_2, then either $N_{c_1} > N_{c_2}$, or $N_{c_1} = N_{c_2}$ and $c_1 < c_2$. Therefore $\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2) = N_{c_1} - N_{c_2} + \mathbb{1}\left[c_2 > c_1\right] > 0$.

If $\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2) = N_{c_1} - N_{c_2} + \mathbb{1}\left[c_2 > c_1\right] > 0$, then either $N_{c_1} > N_{c_2}$, or $N_{c_1} = N_{c_2}$ and $c_2 > c_1$. Therefore, c_1 is dominant over c_2 and beats c_2. □

Lemma E.2. *If the adversary wants to change the prediction of models such that class c_2 beats class c_1, then he needs to make sure that* $\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2)$ *becomes non-positive, i.e.,* $\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2) \leq 0$.

Proof. According to Lemma E.1, after the adversary poisons models such that c_2 beats c_1, $\text{gap}(\{f_i\}_{i=1}^k, c_2, c_1) > 0$. Now, we show that it further implies that $\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2) \leq 0$. Since $\text{gap}(\{f_i\}_{i=1}^k, c_2, c_1) > 0$, $N_{c_2} - N_{c_1} + \mathbb{1}\left[c_1 > c_2\right] > 0$. There are two cases:

- $c_1 > c_2$. In this case, $N_{c_2} - N_{c_1} \geq 0$, which further implies that

$$\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2) = N_{c_1} - N_{c_2} + \mathbb{1}\left[c_2 > c_1\right] = N_{c_1} - N_{c_2} \leq 0$$

- $c_2 > c_1$. In this case, $N_{c_2} - N_{c_1} > 0$, which further implies that

$$\text{gap}(\{f_i\}_{i=1}^k, c_1, c_2) = N_{c_1} - N_{c_2} + \mathbb{1}\left[c_2 > c_1\right] = N_{c_1} - N_{c_2} + 1 \leq 0$$

□

E.2. Proof of Theorem 3.4

Proof. We consider how the adversary can change the prediction of our aggregation method on sample x. Note that we further eliminate x from notations for the sake of simplicity. Let c^{pred} denote the predicted class and c^{sec} denote the other selected class in Round 1. The adversary should ensure that either c^{pred} is not selected as the top two classes in Round 1 or if it makes it to Round 2, it loses this round to another class.

We start with the first strategy. As c^{pred} should be eliminated from the top-two classes of Round 1, it means that the adversary needs to choose two different classes $c_1, c_2 \in \mathcal{C}\backslash\{c^{\text{pred}}\}$ and ensure that c_1 and c_2 both are dominant over c^{pred} in this round. This is the exact definition of `Certv2`, i.e., it needs more than $\texttt{Certv2}(c^{\text{pred}}, c_1, c_2)$ poisoned samples. As the adversary can choose classes c_1, c_2, eliminating c^{pred} in Round 1 requires more than $\texttt{Cert}^{\text{R1}}$ poisoned sample where

$$\texttt{Cert}^{\text{R1}} := \min_{c_1, c_2 \in \mathcal{C}\backslash\{c^{\text{pred}}\}} \texttt{Certv2}(c^{\text{pred}}, c_1, c_2).$$

Therefore, as long as the number of poisoned samples is $\leq \texttt{Cert}^{\text{R1}}$, the first strategy will not work.

Next, in the second strategy, the adversary ensures that class c' beats c^{pred} in Round 2. To do so, it needs to poison models such that (a) c' is selected in the top-two classes of Round 1, (b) c' beats c^{pred} in Round 2.

For (a), c' should beat either of c^{pred} or c^{sec} in Round 1. As we have already considered the case that c^{pred} is eliminated in Round 1, we focus on the case that c' needs to beat c^{sec}. According to the definition of `Certv1`, this requires more than $\texttt{Cert}^{\text{R2}}_{c',1}$ samples where

$$\texttt{Cert}^{\text{R2}}_{c',1} := \texttt{Certv1}(c^{\text{sec}}, c').$$

Note that c' can be c^{sec}.

For (b), let $g_i^{c'} : \mathcal{X} \to \{c', c^{\text{pred}}\}$ denote the binary c^{pred} vs c' classifier obtained from f_i, i.e.,

$$g_i^{c'}(x) := \begin{cases} c' & \text{if } f_i^{\text{logits}}(x, c') > f_i^{\text{logits}}(x, c^{\text{pred}}) \\ c^{\text{pred}} & \text{otherwise} \end{cases}.$$

The adversary needs to ensure that in this binary classification problem, class c' beats c^{pred}. This is the definition of `Certv1`, i.e., it requires at least $\texttt{Cert}^{\text{R2}}_{c',2}$ poisoned samples where

$$\texttt{Cert}^{\text{R2}}_{c',2} := \texttt{Certv1}(\{g_i^{c'}\}_{i=1}^k, c^{\text{pred}}, c').$$

Overall, since the adversary can choose the class c', we obtain the bound

$$\texttt{Cert}^{\text{R2}} := \min_{c' \neq c^{\text{pred}}} \max\{\texttt{Cert}^{\text{R2}}_{c',1}, \texttt{Cert}^{\text{R2}}_{c',2}\}.$$

□

Now we explain how `Certv1` and `Certv2` can be efficiently calculated in both DPA+ROE and FA+ROE.

E.3. Deep Partition Aggregation + Run-off Election

We first prove Lemma 3.5 which calculates the value of `Certv1` in DPA+ROE. After that, we focus on how to calculate `Certv2` in this method by proving Lemma 3.6.

E.3.1. Proof of Lemma 3.5

Proof. We assume without loss of generality that $c_1 \neq c_2$ because if $c_1 = c_2$, then the claim holds trivially because $\texttt{Certv1} = -1$ is a valid certificate.

We want to find the value of $\texttt{Certv1}(\{f_i\}_{i=1}^k, c_1, c_2)$. Based on Lemma E.2, in order to ensure that c_2 beats c_1, we require $\text{gap}(c_1, c_2)$ to be non-positive, i.e., $\text{gap}(c_1, c_2) \leq 0$. Now we consider how poisoning each partition can change the gap. For simplicity, we show this gap with g.

When poisoning partition D_i, the adversary can change the prediction of f_i to be whatever it wants. We will consider how the adversary can change g by fooling model f_i. Note that by poisoning D_i, none of the other classifiers $\{f_j\}_{j \neq i}$ change.

1. $f_i(x) = c_1$. In this case, if the adversary fools this model and changes its prediction to $\tilde{c} \neq c_1$, we have two cases:
   - $\tilde{c} = c_2$. In this case, g decreases by 2.

- $\tilde{c} = c'$. In this case, g decreases by 1.

2. $f_i(x) = c_2$. In this case, if the adversary changes the prediction to $\tilde{c} \neq c_2$, we have two cases:
   - $\tilde{c} = c_1$. In this case, g increases by 2.
   - $\tilde{c} = c'$. In this case, g increases by 1.
3. $f_i(x) = c'$ where $c' \notin \{c_1, c_2\}$ In this case, if the adversary changes the prediction to $\tilde{c} \neq c'$, we have three cases:
   - $\tilde{c} = c_1$. In this case, g increases by 1.
   - $\tilde{c} = c_2$. In this case, g decreases by 1.
   - $\tilde{c} = c''$. In this case, g does not change.

As seen above, by poisoning a single partition, the adversary can reduce g by at most 2. Hence, ensuring $\mathrm{g} \leq 0$ requires at least $\max\left(0, \left\lceil \frac{\mathrm{g}}{2} \right\rceil\right)$ poisoned samples. Therefore, as long as the number of poisoned samples is at most $\max\left(0, \left\lceil \frac{\mathrm{g}}{2} \right\rceil\right) - 1$, the prediction will not change, finishing the proof. □

E.3.2. Proof of Lemma 3.6

Proof. We first consider the case $c \in \{c_1, c_2\}$. Assuming that $c = c_1$, then since $\mathrm{gap}(c, c_1) = 0$, calculating a 2v1 certificate for c, c_1, c_2 is the same as calculating a 1v1 certificate for c, c_2. The claim therefore follows from Lemma 3.5 as the numbers given by the dp and Lemma 3.5 coincide: we have $\mathrm{gap}_1 = 0$, which implies $\max(\mathrm{gap}_1, \mathrm{gap}_2) = \mathrm{gap}_2$ and therefore

$$\mathrm{dp}[\mathrm{gap}_1, \mathrm{gap}_2] = \left\lceil \frac{\max(0, \mathrm{gap}(c, c_2))}{2} \right\rceil - 1$$

We therefore assume without loss of generality that $c \notin \{c_1, c_2\}$.

We want to find the value of $\texttt{Certv2}(\{f_i\}_{i=1}^k, c, c_1, c_2)$. Based on Lemma E.2, in order to ensure that both c_1 and c_2 beat c, both $\mathrm{gap}(c, c_1)$ and $\mathrm{gap}(c, c_2)$ should become non-positive. For simplicity, we denote those gaps by g_1 and g_2, respectively.

When poisoning partition D_i, the adversary can change the prediction of f_i to be whatever it wants. Now we consider the effect of poisoning model f_i on g_1 and g_2. Note that by poisoning D_i, none of the other classifiers $\{f_j\}_{j \neq i}$ change.

1. $f_i(x) = c$. In this case, if the adversary fools this model and changes its prediction to $\tilde{c} \neq c$, we have three cases:
   - $\tilde{c} = c_1$. In this case, g_1 decreases by 2 while g_2 decreases by 1 (***type (i)***).
   - $\tilde{c} = c_2$. In this case, g_1 decreases by 1 while g_2 decreases by 2 (***type (ii)***).
   - $\tilde{c} = c'$ where $c' \notin \{c_1, c_2\}$. In this case, both g_1 and g_2 are reduced by 1.
2. $f_i(x) = c_1$. In this case, if the adversary changes the prediction to $\tilde{c} \neq c_1$, we have three cases:
   - $\tilde{c} = c$. In this case, g_1 increases by 2 while g_2 increases by 1.
   - $\tilde{c} = c_2$. In this case, g_1 increases by 1 while g_2 decreases by 1.
   - $\tilde{c} = c'$ where $c' \notin \{c, c_2\}$. In this case, g_1 increases by 1 while g_2 remains same.
3. $f_i(x) = c_2$. In this case, if the adversary changes the prediction to $\tilde{c} \neq c_2$, we have three cases:
   - $\tilde{c} = c$. In this case, g_1 increases by 1 while g_2 increases by 2.
   - $\tilde{c} = c_1$. In this case, g_1 decreases by 1 while g_2 increases by 1.
   - $\tilde{c} = c'$ where $c' \notin \{c, c_1\}$. In this case, g_1 remains same while g_2 increases by 1.
4. $f_i(x) = c'$ where $c' \notin \{c, c_1, c_2\}$ In this case, if the adversary changes the prediction to $\tilde{c} \neq c'$, we have four cases:
   - $\tilde{c} = c$. In this case, both g_1 and g_2 increase by 1.
   - $\tilde{c} = c_1$. In this case, g_1 decreases by 1 while g_2 remains same.
   - $\tilde{c} = c_2$. In this case, g_1 remains the same while g_2 decreases by 1.
   - $\tilde{c} = c''$ where $c'' \notin \{c, c_1, c_2\}$. In this case, none of g_1 and g_2 change.

As the adversary's goal is to make both g_1 and g_2 non-positive with the minimum number of poisoned samples, based on the scenarios above, **by poisoning a single model**, its power is bounded either with *type (i)* decreasing g_1 by 2 and decreasing g_2 by 1, or with *type (ii)* decreasing g_1 by 1 and decreasing g_2 by 2.

Now we use induction to prove that array dp given in the lemma statement, can find a lower bound on the minimum number of poisoned samples. For base cases, we consider two cases when $\min(\mathrm{g}_1, \mathrm{g}_2) \leq 1$.

- $\mathrm{g}_1 = \mathrm{g}_2 = 0$. In this case, no poisoned samples needed so $\mathrm{dp}[0, 0] = -1$
- $\max(\mathrm{g}_1, \mathrm{g}_2) > 0$ and $\min(\mathrm{g}_1, \mathrm{g}_2) \leq 1$. In this case, the adversary needs at least one poisoned sample. Furthermore, by one poisoned sample, both g_1 and g_2 can be reduced by at most 2. Hence, the adversary needs at least $\left\lceil \frac{\max(\mathrm{g}_1, \mathrm{g}_2)}{2} \right\rceil$ poisoned samples. We note that this number of poisoned samples is sufficient. Assume without loss of generality that $\mathrm{g}_1 \leq \mathrm{g}_2$. Therefore, $\mathrm{g}_1 \leq 1$ and $\mathrm{g}_2 > 0$ by assumption. Assuming the adversary changes $\left\lceil \frac{g_2}{2} \right\rceil$ predictions from c to c_2, then gap_2 will become non-positive because each change reduces the gap by 2. Additionally, gap_1 will become non-positive because each change reduces gap_1 by 1 and $\mathrm{gap}_1 - \left\lceil \frac{g_2}{2} \right\rceil \leq 1 - 1 \leq 0$.

  This implies that $\mathrm{dp}[\mathrm{g}_1, \mathrm{g}_2] = \left\lceil \frac{\max(\mathrm{g}_1, \mathrm{g}_2)}{2} \right\rceil - 1$.

Now we find a lower bound when $\mathrm{g}_1, \mathrm{g}_2 > 1$. Define $i = \mathrm{g}_1$ and $j = \mathrm{g}_2$. The adversary has two options when using one poisoned sample. (1) It reduces i by 2 and j by 1, then according to induction, it needs at least $\mathrm{dp}[i-2, j-1]$ more poisoned samples. (2) It reduces i by 1 and j by 2. Hence it requires at least $\mathrm{dp}[i-1, j-2]$ more poisoned samples.

As a result, the minimum number of poisoned samples is at least $\mathrm{dp}[i, j] = 1 + \min(\mathrm{dp}[i-2, j-1], \mathrm{dp}[i-1, j-2])$.

This finishes the proof. □

E.4. Finite Aggregation + Run-off Election

In this part, we first provide a lemma that we use to prove Lemmas 3.7 and 3.8.

Lemma E.3. *For each $i \in [kd]$, let G_i be a function of the model f_i and define the gap g as $\mathrm{g} := \sum_{i \in [kd]} G_i + C$ where C is some fixed constant. For a bucket $b \in [kd]$, define its poisoning power pw_b as the amount g would decrease if we set all G_j for $j \in h_{spr}(b)$ to their minimum possible value. Given the array $(\mathrm{pw}_b)_{b \in [kd]}$ which represents the adversary's power in terms of reducing the gap g, let $\pi = (\pi_1, \pi_2, ..., \pi_{kd})$ be a permutation on $[kd]$ such that $\mathrm{pw}_{\pi_1} \geq \mathrm{pw}_{\pi_2} \geq ... \geq \mathrm{pw}_{\pi_{kd}}$. The adversary needs to poison more than t buckets to make $\mathrm{g} \leq 0$ where t is the maximum non-negative integer such that $\sum_{i=1}^{t} \mathrm{pw}_{\pi_i} < \mathrm{g}$. Furthermore, $t = \textsc{CertFA}((\mathrm{pw}_b)_{b \in [kd]}, \mathrm{g})$.*

Proof. Let $B = \{b_1, b_2, ..., b_{t'}\}$ be a set of buckets that if the adversary poisons them, can ensure that $\mathrm{g} \leq 0$. We show that $t \leq t'$. Since poisoning bucket b_i can reduce the gap by pw_{b_i}, poisoning buckets in B can reduce the gap by **at most** $\sum_{i=1}^{t'} \mathrm{pw}_{b_i}$. The reason we say "at most" is the fact that a base classifier uses several buckets in its training sets, so by poisoning more than one of those buckets, we count the effect of poisoning that particular classifier several times. As π sorts the array in decreasing order, $\sum_{i=1}^{t'} \mathrm{pw}_{\pi_i} \geq \sum_{i=1}^{t'} \mathrm{pw}_{b_i}$. This implies that $\sum_{i=1}^{t'} \mathrm{pw}_{\pi_i} \geq \mathrm{g}$. According to the definition of t which is the maximum non-negative integer such that $\sum_{i=1}^{t} \mathrm{pw}_{\pi_i} < \mathrm{g}$, we conclude that $t \leq t'$. In order to find t, we sort array pw in decreasing order and while the sum of the elements we have picked does not reach g, we keep picking elements, moving from the left side to the right side of the array. A pseudocode of how to find t is given in Algorithm 4, which further proves that $t = \textsc{CertFA}((\mathrm{pw}_b)_{b \in [kd]}, \mathrm{g})$.

Note that $t \leq kd$ always exists as fooling all models to do in favor of a desired class guarantees that the class will beat all other classes. □

E.4.1. Proof of Lemma 3.7

Proof. We can have the same argument of Section E.3.1 so in order to ensure that c_2 beats c_1, $\mathrm{gap}(c_1, c_2)$ should become non-positive, i.e., $\mathrm{gap}(c_1, c_2) \leq 0$. Now we consider how poisoning each bucket can change the gap. For simplicity, we show this gap with g.

Let A be the set of indices of classifiers that can be fooled after poisoning b. Formally, $A := h_{\mathrm{spr}}(b)$. We consider $j \in A$ and examine how g changes as f_j is poisoned.

1. $f_j(x) = c_1$. In this case, if the adversary fools this model and changes its prediction to $\tilde{c} \neq c_1$, we have two cases:
   - $\tilde{c} = c_2$. In this case, g decreases by 2.
   - $\tilde{c} = c'$ where $c' \neq c_2$. In this case, g is reduced by 1.

   This implies that in this case, in terms of reducing g, the adversary's power is **bounded by** 2.

2. $f_j(x) = c_2$. In this case, if the adversary changes the prediction to $\tilde{c} \neq c_2$, we have two cases:
   - $\tilde{c} = c_1$. In this case, g increases by 2.
   - $\tilde{c} = c'$ where $c' \neq c_1$. In this case, g increases by 1.

   This implies that in this case, in terms of reducing g, the adversary's power is **bounded by** 0.

3. $f_i(x) = c'$. In this case, if the adversary changes the prediction to $\tilde{c} \neq c'$, we have three cases:
   - $\tilde{c} = c_1$. In this case, g increases by 1.
   - $\tilde{c} = c_2$. In this case, g decreases by 1.
   - $\tilde{c} = c''$ where $c'' \notin \{c_1, c_2\}$. In this case, g remains the same.

   This implies that in this case, in terms of reducing g, the adversary's power is **bounded by** 1.

Note that we can have the same argument for all $j \in A$. According to the above cases, We define the poisoning power of bucket b with respect to classes c_1, c_2 as

$$\mathrm{pw}_{c_1,c_2,b} := \sum_{j \in h_{\mathrm{spr}}(b)} 2\mathbb{1}\left[f_j(x) = c_1\right] + \mathbb{1}\left[f_j(x) \notin \{c_1, c_2\}\right].$$

Using Lemma E.3, with $G_i := \mathbb{1}\left[f_i(x) = c_1\right] - \mathbb{1}\left[f_i(x) = c_2\right]$ and $C := \mathbb{1}\left[c_2 > c_1\right]$, it follows that $\texttt{Certv1}(\{f_i\}_{i=1}^k, c_1, c_2) := \textsc{CertFA}((\mathrm{pw}_{c_1,c_2,b})_{b \in [kd]}, \mathrm{g})$ is a 1v1 certificate. □

E.4.2. Proof of Lemma 3.8

Proof. We can exactly follow the same initial steps of proof in Lemma 3.6. Therefore, in order to ensure that both c_1 and c_2 beat c, both $\mathrm{gap}(c, c_1)$ and $\mathrm{gap}(c, c_2)$ should become non-positive. For simplicity, we denote those gaps by g_1 and g_2, respectively. Furthermore, as both g_1 and g_2 should become non-positive, their sum which we denote by $\mathrm{g}_{c_1+c_2} := \mathrm{g}_1 + \mathrm{g}_2$ should become non-positive as well.

Now, we analyze how the adversary can affect g_1, g_2, and $\mathrm{g}_{c_1+c_2}$ by poisoning a bucket b. Let A be the set of indices of classifiers that can be fooled after poisoning b. Formally, $A := h_{\mathrm{spr}}(b)$. We consider $j \in A$ and examine how $\mathrm{g}_1, \mathrm{g}_2$, and $\mathrm{g}_{c_1+c_2}$ change as f_j is poisoned.

1. $f_j(x) = c$. In this case, if the adversary fools this model and changes its prediction to $\tilde{c} \neq c$, we have three cases:
   - $\tilde{c} = c_1$. In this case, g_1 decreases by 2, g_2 decreases by 1, and $\mathrm{g}_{c_1+c_2}$ is reduced by 3.
   - $\tilde{c} = c_2$. In this case, g_1 decreases by 1, g_2 decreases by 2, and $\mathrm{g}_{c_1+c_2}$ is reduced by 3.
   - $\tilde{c} = c'$ where $c' \notin \{c_1, c_2\}$. In this case, both g_1 and g_2 are reduced by 1 while $\mathrm{g}_{c_1+c_2}$ is reduced by 2.

   This implies that in terms of reducing g_1, g_2, and $\mathrm{g}_{c_1+c_2}$, adversary's power is **bounded by** $2, 2$**, and** 3**, respectively**.

2. $f_j(x) = c_1$. In this case, if the adversary changes the prediction to $\tilde{c} \neq c_1'$, we have three cases:
   - $\tilde{c} = c$. In this case, g_1 increases by 2, g_2 increases by 1, and $\mathrm{g}_{c_1+c_2}$ increases by 3.
   - $\tilde{c} = c_2$. In this case, g_1 increases by 1, g_2 decreases by 1, and $\mathrm{g}_{c_1+c_2}$ does not change.
   - $\tilde{c} = c''$ where $c'' \notin \{c, c_2\}$. In this case, g_1 increases by 1, g_2 remains same, and $\mathrm{g}_{c_1+c_2}$ increases by 1.

   This implies that in terms of reducing g_1, g_2, and $\mathrm{g}_{c_1+c_2}$, adversary's power is **bounded by** $0, 1$**, and** 0**, respectively**.

3. $f_j(x) = c_2$. In this case, if the adversary changes the prediction to $\tilde{c} \neq c_2$, we have three cases:
   - $\tilde{c} = c$. In this case, g_1 increases by 1, g_2 increases by 2, and $\mathrm{g}_{c_1+c_2}$ increases by 3.

- $\tilde{c} = c_1$. In this case, g_1 decreases by 1, g_2 increases by 1, and $\mathrm{g}_{c_1+c_2}$ does not change.
- $\tilde{c} = c'$ where $c' \notin \{c, c_1\}$. In this case, g_1 remains same, g_2 increases by 1, and $\mathrm{g}_{c_1+c_2}$ increases by 1.

This implies that in terms of reducing g_1, g_2, and $\mathrm{g}_{c_1+c_2}$, adversary's power is **bounded by** $1, 0$**, and** 0**, respectively**.

4. $f_j(x) = c'$ where $c' \notin \{c, c_1, c_2'\}$ In this case, if the adversary changes the prediction to $\tilde{c} \neq c''$, we have four cases:
   - $\tilde{c} = c$. In this case, both g_1 and g_2 increase by 1 while $\mathrm{g}_{c_1+c_2}$ increases by 2.
   - $\tilde{c} = c_1$. In this case, g_1 decreases by 1, g_2 remains same, and $\mathrm{g}_{c_1+c_2}$ decreases by 1.
   - $\tilde{c} = c_2$. In this case, g_1 remains the same, g_2 decreases by 1, and $\mathrm{g}_{c_1+c_2}$ decreases by 1.
   - $\tilde{c} = c''$ where $c'' \notin \{c, c_1, c_2\}$. In this case, none of g_1, g_2, and $\mathrm{g}_{c_1+c_2}$ change.

   This implies that in terms of reducing g_1, g_2, and $\mathrm{g}_{c_1+c_2}$, adversary's power is **bounded by** $1, 1$**, and** 1**, respectively**.

Note that we can have the same argument for all $j \in A$.

In terms of reducing g_1, we define the *poisoning power of bucket* b, i.e., the maximum amount that g_1 can be reduce by poisoning b as $\mathrm{pw}_{c,c_1,b}$. Based on the scenarios above,

$$\mathrm{pw}_{c,c_1,b} := \sum_{j \in h_{\mathrm{spr}}(b)} 2\mathbb{1}\left[f_j(x) = c\right] + \mathbb{1}\left[f_j(x) \notin \{c_1, c\}\right].$$

According to Lemma E.3, with $G_i := \mathbb{1}\left[f_i(x) = c\right] - \mathbb{1}\left[f_i(x) = c_1\right]$ and $C := \mathbb{1}\left[c_1 > c\right]$, the adversary needs more than $\texttt{Certv2}^{(1)}$ poisoned samples where

$$\texttt{Certv2}^{(1)} := \textsc{CertFA}((\mathrm{pw}_{c,c_1,b})_{b \in [kd]}, \mathrm{g}_1)$$

Based on the scenarios above, in terms of reducing g_2 to make it non-positive, poisoning power of bucket b is at most

$$\mathrm{pw}_{c,c_2,b} := \sum_{j \in h_{\mathrm{spr}}(b)} 2\mathbb{1}\left[f_j(x) = c\right] + \mathbb{1}\left[f_j(x) \notin \{c_2, c\}\right].$$

According to Lemma E.3, with $G_i := \mathbb{1}\left[f_i(x) = c\right] - \mathbb{1}\left[f_i(x) = c_2\right]$ and $C := \mathbb{1}\left[c_2 > c\right]$, the adversary needs more than $\texttt{Certv2}^{(2)}$ poisoned samples where

$$\texttt{Certv2}^{(2)} := \textsc{CertFA}((\mathrm{pw}_{c,c_2,b})_{b \in [kd]}, \mathrm{g}_2)$$

Finally, In terms of reducing $\mathrm{g}_{c_1+c_2}$, bucket b's poisoning power is at most

$$\mathrm{pw}^{+}_{c_1,c_2,b} := \sum_{j \in h_{\mathrm{spr}}(b)} 3\mathbb{1}\left[f_j(x) = c\right] + \mathbb{1}\left[f_j(x) \notin \{c, c_1, c_2\}\right].$$

Therefore, using Lemma E.3 with $G_i := 2\mathbb{1}\left[f_i(x) = c\right] - \mathbb{1}\left[f_i(x) = c_1\right] - \mathbb{1}\left[f_i(x) = c_2\right]$ and $C := \mathbb{1}\left[c_1 > c\right] + \mathbb{1}\left[c_2 > c\right]$, this implies that the adversary is required to provide more than $\texttt{Certv2}^{+}$ poisoned samples where

$$\texttt{Certv2}^{+} := \textsc{CertFA}((\mathrm{pw}^{+}_{c_1,c_2,b})_{b \in [kd]}, \mathrm{g}_{c_1+c_2}).$$

As the adversary has to ensure that all $\mathrm{g}_1, \mathrm{g}_2$, and $\mathrm{g}_{c_1+c_2}$ become non-positive, it needs to satisfy all those three conditions. Hence $\texttt{Certv2}$ defines as follows

$$\texttt{Certv2} := \max\{\texttt{Certv2}^{(1)}, \texttt{Certv2}^{(2)}, \texttt{Certv2}^{+}\}$$

is a 2v1 certificate. □