On Collective Robustness of Bagging Against Data Poisoning

Ruoxin Chen [1] **Zenan Li** [1] **Jie Li** [1] **Chentao Wu** [1] **Junchi Yan** [1]

Abstract

Bootstrap aggregating (bagging) is an effective ensemble protocol, which is believed can enhance robustness by its majority voting mechanism. Recent works further prove the sample-wise robustness certificates for certain forms of bagging (e.g. partition aggregation). Beyond these particular forms, in this paper, *we propose the first collective certification for general bagging to compute the tight robustness against the global poisoning attack*. Specifically, we compute the maximum number of simultaneously changed predictions via solving a binary integer linear programming (BILP) problem. Then we analyze the robustness of vanilla bagging and give the upper bound of the tolerable poison budget. Based on this analysis, *we propose hash bagging* to improve the robustness of vanilla bagging almost for free. This is achieved by modifying the random subsampling in vanilla bagging to a hash-based deterministic subsampling, as a way of controlling the influence scope for each poisoning sample universally. Our extensive experiments show the notable advantage in terms of applicability and robustness. Our code is available at `https://github.com/Emiyalzn/ICML22-CRB`.

1. Introduction

Bagging (Breiman, 1996), refers to an ensemble learning protocol that *trains sub-classifiers on the subsampled subtrainsets and makes predictions by majority voting*, which is a commonly used method to avoid overfitting. Recent works (Biggio et al., 2011; Levine & Feizi, 2021; Jia et al., 2021) show its superior certified robustness in defending data poisoning attacks. Moreover, compared to other certified defenses, bagging is a natural plug-and-play method with a high compatibility with various model architectures and training algorithms, which suggests its great potential.

Some works (Levine & Feizi, 2021; Jia et al., 2021; Wang et al., 2022) have proved the sample-wise robustness certificates against the sample-wise attack (the attacker aims to corrupt the prediction for the target data) for certain forms of bagging. However, we notice that, *there is a white space in the collective robustness certificates against the global poisoning attack* (the attacker attempts to maximize the number of simultaneously changed predictions when predicting the testset), although the global attack is more general and critical than the sample-wise attack for: I) the sample-wise attack is only a variant of the global poisoning attack when the testset size is one; II) unlike adversarial examples (Goodfellow et al., 2014) which is sample-wise, data poisoning attacks are naturally global, where the poisoned trainset has a global influence on all the predictions; III) the global attack is believed more harmful than the sample-wise attack. Current works (Levine & Feizi, 2021; Jia et al., 2021) simply *count the number of robust predictions guaranteed by the sample-wise certification*, as a lower bound of the collective robustness. However, this lower bound often overly under-estimates the actual value. We aim to provide a formal collective certification for general bagging, to fill the gap in analyzing the certified robustness of bagging.

In this paper, *we take the first step towards the collective certification for general bagging.* Our idea is to formulate a binary integer linear programming (BILP) problem, of which objective function is to maximize the number of simultaneously changed predictions w.r.t. the given poison budget. The certified collective robustness equals the testset size minus the computed objective value. To reduce the cost of solving the BILP problem, a decomposition strategy is devised, which allows us to compute a collective robustness lower bound within a linear time of testset size.

Moreover, we analyze the certified robustness of vanilla bagging, demonstrating that it is not an ideal certified defense by deriving the upper bound of its tolerable poison budget. To address this issue, *we propose hash bagging to improve the robustness of vanilla bagging almost for free.* Specifically, we modify the random subsampling in vanilla bagging to hash-based subsampling, to restrict the influence

[1]Department of Computer Science and Engineering and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China. Jie Li and Junchi Yan are also with Shanghai AI Laboratory, Shanghai, China. Correspondence to: Jie Li < lijiecs@sjtu.edu.cn>.

Proceedings of the 39^{th} *International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022.

scope of each training sample within a bounded number of sub-trainsets deterministically. We compare hash bagging to vanilla bagging to show its superior certified robustness and the comparable accuracy. Furthermore, compared to prior elaborately designed bagging-based defenses (Levine & Feizi, 2021; Jia et al., 2021), hash bagging is a more general and practical defense method, which covers almost all forms of bagging. **The main contributions are:**

1) For the first time to our best knowledge, we derive the collective certification for general bagging. We accelerate the solving process by decomposition. Remarkably, our computed certified collective robustness is theoretically better than that of the sample-wise certifications.

2) We derive an upper bound of tolerable poison budget for bagging. Our derived bound is tight if we only have access to the sub-trainsets and sub-classifier predictions.

3) We propose *hash bagging* as a defense technique to improve the robustness for vanilla bagging almost for free, in the sense of neither introducing additional constraints on the hyper-parameters nor restricting the forms of bagging.

4) We evaluate our two techniques empirically and quantitatively on four datasets: collective certification and hash bagging. Results show: i) collective certification can yield a much stronger robustness certificate. ii) Hash bagging effectively improves vanilla bagging on the certified robustness.

2. Related Works

Both machine-learning classifiers (e.g. Bayes and SVM) and neural-network classifiers are vulnerable to data poisoning (Li et al., 2020; 2022; Nelson et al., 2008; Biggio et al., 2012; Xiao et al., 2015; Yao et al., 2019; Zhang et al., 2020; Liu et al., 2019). Since most heuristic defenses (Chen et al., 2019; Gao et al., 2019; Tran et al., 2018; Liu et al., 2019; Qiao et al., 2019) have been broken by the new attacks (Koh et al., 2018; Tramèr et al., 2020), developing certified defenses is critical.

Certified defenses against data poisoning. Certified defenses (Steinhardt et al., 2017; Wang et al., 2020) include random flipping (Rosenfeld et al., 2020), randomized smoothing (Weber et al., 2020), differential privacy (Ma et al., 2019) and bagging-based defenses (Levine & Feizi, 2021; Jia et al., 2021). Currently, only the defenses (Ma et al., 2019; Jinyuan Jia & Gong, 2022; Jia et al., 2021; Levine & Feizi, 2021) are designed for the general data poisoning attack (the attacker can arbitrarily insert/delete/modify a bounded number of samples). However, their practicalities suffer from various limitations. (Ma et al., 2019) is limited to the training algorithms with the differential privacy guarantee. (Jinyuan Jia & Gong, 2022) certify the robustness for the machine-learning classifiers kNN/rNN (Nearest Neighbors), which might be unable to scale to the large tasks. Currently, only two bagging variants (Jia et al., 2021; Levine & Feizi, 2021) have demonstrated the high compatibility w.r.t. the model architecture and the training algorithm, with the state-of-the-art certified robustness. Their success highlights the potential of bagging, which motivates us to study the robustness for general bagging.

Table 1. Notations.

Notation	Description
K	The sub-trainset size.
G	The number of sub-trainsets.
N	The trainset size.
$\mathcal{D}_{train} = \{s_i\}_{i=0}^{N-1}$	The trainset consisting of N training samples $\{s_i\}_{i=0}^{N-1}$.
$\mathcal{D}_{test} = \{x_j\}_{j=0}^{M-1}$	The trainset consisting of M testing samples $\{x_j\}_{j=0}^{M-1}$.
$y \in \mathcal{Y}$	y and $\mathcal{Y}$ denote the class and the output space respectively.
$\mathcal{D}_g$	The g-th sub-trainset.
$f_g(\cdot)$	The g-th sub-classifier in bagging.
$g(\cdot)$	The ensemble classifier consisting of all the sub-classifiers.
$V_x(y)$	The number of votes for the class $y \in \mathcal{Y}$ when predicting x.
$\mathrm{Hash}(\alpha)$	The hash value of α.

Robustness certifications against data poisoning. Current robustness certifications (Wang et al., 2020; Ma et al., 2019; Jia et al., 2021; Jinyuan Jia & Gong, 2022; Levine & Feizi, 2021) against data poisoning are mainly focusing on the sample-wise robustness, which evaluates the robustness against the sample-wise attack. However, the collective robustness certificates are rarely studied, which might be a more practical metric because the poisoning attack naturally is a kind of global attack that can affect all the predictions. To our best knowledge, only (Jinyuan Jia & Gong, 2022) considers the collective robustness against global poisoning attack. Specifically, it gives the collective certification for a machine-learning classifier rNN, but the certification is based on the unique geometric property of rNN.

3. Collective Certification to Bagging

In this section, first we formally define vanilla bagging and the threat model, as the basement of the collective certification. Then we propose the collective certification, and analyze the upper bound of the tolerable poison budget. All our notations are summarized in Table 1.

Definition 1 (Vanilla bagging)**.** *Given a trainset $\mathcal{D}_{train} = \{s_i\}_{i=0}^{N-1}$ where s_i refers to the i-th training sample, following (Breiman, 1996; Jia et al., 2021; Levine & Feizi, 2021), vanilla bagging can be summarized into three steps:*
i) Subsampling: construct G sub-trainsets $\mathcal{D}_g$ (of size K) ($g = 0, \ldots, G-1$), by subsampling K training samples from $\mathcal{D}_{train}$ G times;
ii) Training: train the g-th sub-classifier $f_g(\cdot)$ on the sub-trainset $\mathcal{D}_g$ ($g = 0, \ldots, G-1$);
iii) Prediction: the ensemble classifier (denoted by $g(x)$) makes the predictions, as follow:

$$g(x) = \arg\min_{y} \arg\max_{y \in \mathcal{Y}} V_x(y) \qquad (1)$$

where $V_x(y) \coloneqq \sum_{g=0}^{G-1} \mathbb{I}\{f_g(x) = y\}$. *(*$\mathbb{I}\{\}$ *is the indicator function) is the number of sub-classifiers that predict class* y. $\arg\min_y$ *means that,* $g(x)$ *predicts* ***the majority class of the smallest index*** *if there exist multiple majority classes.*

3.1. Threat Model

We assume that *the sub-classifiers are extremely vulnerable to the changes in their sub-trainsets*, since our certification is agnostic towards the sub-classifier architecture. In another word, the attacker is considered to fully control the sub-classifier f_g once the sub-trainset $\mathcal{D}_g$ is changed.

Attacker capability: the attacker is allowed to insert r_{ins} samples, delete r_{del} samples, and modify r_{mod} samples.

Attacker objective: for the *sample-wise attack* (corresponding to the sample-wise certification), the attacker aims to change the prediction for the target data. For the *global poisoning attack* (corresponding to the collective certification), the attacker aims to maximize the number of simultaneously changed predictions when predicting the testset.

3.2. $(\mathbf{P1})$: Collective Certification of Vanilla Bagging

Given the sub-trainsets and class distribution of each testing sample, we can compute the collective robustness for vanilla bagging, as shown in Prop. 1.

Proposition 1 (Certified collective robustness of vanilla bagging)**.** *For testset* $\mathcal{D}_{test} = \{x_j\}_{j=0}^{M-1}$*, we denote* $\hat{y}_j = g(x_j)$ $(j = 0, \dots, M-1)$ *the original ensemble prediction, and* $\mathcal{S}_i = \{g \mid s_i \in \mathcal{D}_g\}$ *the set of the indices of the sub-trainsets that contain* s_i *(the* i*-th training sample). Then, the maximum number of simultaneously changed predictions (denoted by* M_{ATK}*) under* r_{mod} *adversarial modifications, is computed by* $(\mathbf{P1})$*:*

$$(\mathbf{P1}): \quad M_{\text{ATK}} = \max_{P_0,\dots,P_{N-1}} \sum_{x_j \in \mathcal{D}_{test}} \mathbb{I}\Big\{\overline{V}_{x_j}(\hat{y}_j) < \max_{y \neq \hat{y}_j} \Big[\overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y < \hat{y}_j\}\Big]\Big\} \tag{2}$$

$$s.t. \quad [P_0, P_1, \dots, P_{N-1}] \in \{0,1\}^N \tag{3}$$

$$\sum_{i=0}^{N-1} P_i \le r_{\text{mod}} \tag{4}$$

$$\overline{V}_{x_j}(\hat{y}_j) = \underbrace{V_{x_j}(\hat{y}_j)}_{\textit{Original votes}} - \underbrace{\sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i = 1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) = \hat{y}_j\}}_{\textit{Influenced votes}}$$
$$\forall x_j \in \mathcal{D}_{test},\ \hat{y}_j = g(x_j) \tag{5}$$

$$\overline{V}_{x_j}(y) = \underbrace{V_{x_j}(y)}_{\textit{Original votes}} + \underbrace{\sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i = 1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) \neq y\}}_{\textit{Influenced votes}}$$
$$\forall x_j \in \mathcal{D}_{test},\ \forall y \in \mathcal{Y}, y \neq \hat{y}_j \tag{6}$$

The certified collective robustness is $M - M_{\text{ATK}}$*.*

We explain each equation. **Eq. (2):** the objective is to maximize the number of simultaneously changed predictions. Note that a prediction is changed if there exists another class with more votes (or with the same number of votes but of the smaller index). **Eq. (3):** $[P_0, \dots, P_{N-1}]$ are the binary variables that represent the poisoning attack, where $P_i = 1$ means that the attacker modifies s_i. **Eq. (4):** the number of modifications is bounded within r_{mod}. **Eq. (5):** $\overline{V}_{x_j}(\hat{y}_j)$, the minimum number of votes for class $\hat{y}_j$ (after being attacked), equals to the original value minus the number of the influenced sub-classifiers whose original predictions are $\hat{y}_j$. **Eq. (6):** $\overline{V}_{x_i}(y)$ $(y \neq y_i)$, the maximum number of votes for class $y : y \neq \hat{y}_j$ (after being attacked), equals to the original value plus the number of influenced sub-classifiers whose original predictions are not y, because that, under our threat model, the attacker is allowed to arbitrarily manipulate the predictions of those influenced sub-classifiers.

3.3. Remarks on Proposition 1

We give our discussion and the remark marked with $*$ mean that the property is undesirable needing improvement.

1) Tightness. The collective robustness certificates computed from $(\mathbf{P1})$ is tight.

2) Sample-wise certificate. We can compute the tight sample-wise certificate for the prediction on the target data x_{target}, by simply setting $\mathcal{D}_{test} = \{x_{\text{target}}\}$.

3) Certified accuracy. We can compute *certified accuracy* (the minimum number of correct predictions after being attacked) if given the oracle labels. Specifically, we compute the certified accuracy over the testset $\mathcal{D}_{test}$, simply by modifying $\sum_{x_j \in \mathcal{D}_{test}}$ in Eq. (2) to $\sum_{x_j \in \Omega}$, where Ω is $\Omega = \{x_j \in \mathcal{D}_{test} : g(x_j) \text{ predicts correctly}\}$. The certified accuracy is $(|\Omega| - M_{\text{ATK}})/M$ where $|\Omega|$ refers to the cardinality of the set Ω. Actually, certified accuracy measures the worst accuracy under all the possible accuracy degradation attacks within the poison budget. Our computed certified accuracy is also tight.

4) Reproducibility requirement*. Both subsampling and training are required to be reproducible, because certified robustness is only meaningful for deterministic predictions. Otherwise, without the reproducibility, given the same trainset and testset, the predictions might be discrete random variables for the random operations in subsampling/training, such that we may observe two different predictions for the same input if we run the whole process (bagging and prediction) twice, even without being attacked.

5) NP-hardness*. $(\mathbf{P1})$ is NP-hard as it can be formulated as a BILP problem. We present more details in Appendix (Section B.2).

3.4. Addressing NP-hardness by Decomposition

Decomposition (Pelofske et al., 2020; Rao, 2008) allows us to compute a certified collective robustness *lower bound* instead of the exact value. Specifically, we first split $\mathcal{D}_{test}$ into Δ-size sub-testsets (denoted by $\mathcal{D}^{\mu}\ :\ \mu = 0, \ldots, \lceil M/\Delta \rceil - 1$). Here we require the size of the last sub-testset is allowed to be less than Δ. Then we compute the maximum number of simultaneously changed predictions (denoted by M^{μ}_{ATK}) for each sub-testset $\mathcal{D}^{\mu}$ under the given poison budget. **We output $M - \sum_{\mu} M^{\mu}_{\text{ATK}}$ as a collective robustness lower bound.** Remarkably, by decomposition, the time complexity is significantly reduced from an exponential time (w.r.t. M) to a linear time (w.r.t. M), as the time complexity of solving the Δ-scale sub-problem can be regarded as a constant. Generally, Δ controls a trade-off between the certified collective robustness and the computation cost: *as we consider the influence of the poisoning attack more holistically (larger Δ), we can obtain a tighter lower bound at a cost of much larger computation.* In particular, our collective certification is degraded to be the sample-wise certification when $\Delta = 1$.

3.5. Upper Bound of Tolerable Poison Budget

Based on Eq. (5), Eq. (6) in $(\mathbf{P1})$, we can compute the upper bound of tolerable poison budget for vanilla bagging.

Proposition 2 (Upper bound of tolerable poison budget)**.** *Given $\mathcal{S}_i = \{g \mid s_i \in \mathcal{D}_g\}$ $(i = 0, \ldots, N-1)$,* the upper bound of the tolerable poisoned samples *(denoted by $\overline{r}$) is*

$$\overline{r} = \min |\Pi|\ s.t.\ |\bigcup_{i \in \Pi} \mathcal{S}_i| > G/2 \tag{7}$$

where Π denotes a set of indices. The upper bound of the tolerable poisoned samples equals the minimum number of training samples that can influence more than a half of sub-classifiers.

The collective robustness must be zero when the poison budget $\geq \overline{r}$. We emphasize that computing $\overline{r}$ is an NP-hard max covering problem (Fujishige, 2005). A simple way of enlarging $\overline{r}$ is to *bound the influence scope for each sample* $|S_i|\ :\ i = 0, \ldots, N-1$. In particular, if we bound the influence scope of each sample to be less than a constant $|S_i| \leq \Gamma\ :\ i = 0, \ldots, N-1$ (Γ is a constant), we have $\overline{r} \geq N/(2\Gamma)$. This is the insight behind hash bagging.

4. Proposed Approach: Hash Bagging

Objective of hash bagging. We aim to improve vanilla bagging by designing a new subsampling algorithm. According to the remarks on Prop. 1, Prop. 2, the new subsampling is expected to own the properties: **i) Determinism:** subsampling should be reproducible. **ii) Bounded influence scope:** inserting/deleting/modifying an arbitrary sample can only influence a limited number of sub-trainsets. **iii) Solvability:** the robustness can be computed within the given time. **iv) Generality:** the subsampling applies to arbitrary K (the sub-trainset size) and G (the number of sub-trainsets).

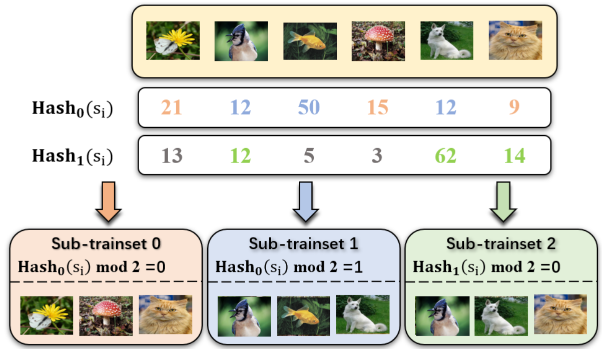


Figure 1. Hash bagging when $N = 6$ (trainset size), $K = 3$ (sub-trainset size), $G = 3$ (number of sub-trainsets). $\hat{G} = \lfloor N/K \rfloor = 2$. By Eq. (9), the 0-th sub-trainset ($\hat{h} = 0, \hat{g} = 0$) is constructed based on $\text{Hash}_0(\text{s}_\text{i}) \mod 2 = 0$ (the samples whose hash values are colored by red). The 1-st sub-trainset ($\hat{h} = 0, \hat{g} = 1$) is constructed by $\text{Hash}_0(\text{s}_\text{i}) \mod 2 = 1$ (the samples whose hash values are colored by blue). The 2-nd sub-trainset ($\hat{h} = 1, \hat{g} = 0$) is constructed by $\text{Hash}_1(\text{s}_\text{i}) \mod 2 = 0$ (the samples whose hash values are colored by green).

The realization of hash bagging is based on the hash values. First let's see a simple case when $GK = N$.

Hash bagging when $GK = N$. Given $\mathcal{D}_{train}$, the g-th sub-trainset $\mathcal{D}_g$ $(g = 0, 1, \ldots, G-1)$ is as follow:

$$\mathcal{D}_g = \{s_i \in \mathcal{D}_{train} \mid \text{Hash}(\text{s}_\text{i}) \mod \text{G} = \text{g}\} \tag{8}$$

where $\text{Hash}(\cdot)$ is the pre-specified hash function. Such that the number of sub-trainsets exactly equals G and the sub-trainset size approximates $N/G = GK/G = K$, because the hash function will (approximately) uniformly allocate each sample to different hash values. Such hash-based subsampling satisfies the following properties: **i) Determinism**: fixing G, K, all G sub-trainsets are uniquely determined by $\mathcal{D}_{train}$ and $\text{Hash}(\cdot)$, which we denoted as the trainset-hash pair $(\mathcal{D}_{train}, \text{Hash}(\cdot))$ for brevity. **ii) Bounded influence scope**: r_{ins} insertions, r_{del} deletions and r_{mod} modifications can influence at most $r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$ sub-trainsets.

Hash bagging for general cases. Given $\mathcal{D}_{train}$ and a series of hash functions $\text{Hash}_h(\cdot)$ $(h = 0, \ldots)$, the g-th sub-trainset $\mathcal{D}_g$ $(g = 0, 1, \ldots, G-1)$ is as follow:

$$\mathcal{D}_g = \{s_i \in \mathcal{D}_{train} \mid \text{Hash}_{\hat{\text{h}}}(\text{s}_\text{i}) \mod \hat{\text{G}} = \hat{\text{g}}\} \tag{9}$$

where $\hat{G} = \lfloor N/K \rfloor, \hat{h} = \lfloor g/\hat{G} \rfloor, \hat{g} = g \mod \hat{G}$. Specifically, we set $\hat{G} = \lfloor N/K \rfloor$, so that the size of each sub-trainset approximates $N/\hat{G} \rightarrow K$. We specify a series of hash functions because that a trainset-hash pair can generate at most $\hat{G}$ sub-trainsets, thus we construct $\lceil G/\hat{G} \rceil$

Algorithm 1: Certify the collective robustness for our proposed hash bagging.

Input: testset $\mathcal{D}_{test} = \{x_j\}_{i=0}^{M-1}$, sub-classifiers $\{f_g\}_{g=1}^{G}$, the poison budget $r_{\text{ins}}, r_{\text{del}}, r_{\text{mod}}$, sub-problem scale Δ.

for $x_j : j = 0, 1, ..., M-1$ **do**
  Compute predictions $\hat{y}_j = f_g(x_j) : g = 1, \ldots, G$;
See the simplification for $(\mathbf{P2})$ (Eq. 15)
Compute the set of breakable predictions Ω ;
Decompose the original problem to Δ-scale sub-problems.
Decompose $\Omega = \bigcup_{\mu=0}^{\lceil M/\Delta \rceil - 1} \mathcal{D}^{\mu}$, where $|\mathcal{D}^{\mu}| = \Delta$ $(\mu = 0, \ldots, \lceil M/\Delta \rceil - 2)$;
for $\mathcal{D}^{\mu} : \mu = 0, 1, ... \lceil M/\Delta \rceil - 1$ **do**
  # Solve the Δ-scale sub-problems.
  Compute the maximum number of simultaneously changed predictions M_{ATK}^{μ} by solving $(\mathbf{P2})$ over $\mathcal{D}^{\mu}$ w.r.t. the poison budget $r_{\text{ins}}, r_{\text{del}}, r_{\text{mod}}$;
Compute the lower bound of the certified collective robustness: $M - \sum_{\mu} M_{\text{ATK}}$;

Output: $M - \sum_{\mu} M_{\text{ATK}}$

trainset-hash pairs, which is enough to generate G sub-trainsets. Then the g-th sub-trainset is the $\hat{g}$-th sub-trainset within the sub-trainsets from the $\hat{h}$-th trainset-hash pair. Fig. 1 illustratively shows an example of hash bagging. Remarkably, hash bagging satisfies: **i) Determinism**: the sub-sampling results only depends on the trainset-hash pairs $\{(\mathcal{D}_{train}, \text{Hash}_h(\cdot)) : h = 0, 1, \ldots, \lceil G/\hat{G} \rceil - 1\}$ if fixing G, K. **ii) Bounded influence scope**: r_{ins} insertions, r_{del} deletions and r_{mod} modifications can influence at most $r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$ sub-trainsets, within the $\hat{G}$ sub-trainsets from each trainset-hash pair. **iii) Generality**: hash bagging can be applied to all the combinations of G, K.

Reproducible training of hash bagging. After constructing G sub-trainsets based on Eq. (9), we train the sub-classifiers in a *reproducible* manner. In our experiments, we have readily realized reproducibility by specifying the random seed for all the random operations.

4.1. $(\mathbf{P2})$: Collective Certification of Hash Bagging

Proposition 3 (Simplified collective certification of hash bagging)**.** *For testset $\mathcal{D}_{test} = \{x_j\}_{j=0}^{M-1}$, we denote $\hat{y}_j = g(x_j)$ $(j = 0, \ldots, M-1)$ the ensemble prediction. The maximum number of simultaneously changed predictions (denoted by M_{ATK}) under r_{ins} insertions, r_{del} deletions and r_{mod} modifications, is computed by $(\mathbf{P2})$:*

$$(\mathbf{P2}): \quad M_{\text{ATK}} = \max_{A_0, \ldots, A_{G-1}} \sum_{x_j \in \mathcal{D}_{test}} \mathbb{I}\Big\{\overline{V}_{x_j}(\hat{y}_j) < \max_{y \neq \hat{y}_j} \Big[\overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y < \hat{y}_j\}\Big]\Big\} \tag{10}$$

$$s.t. \quad [A_0, A_1, \ldots, A_{G-1}] \in \{0,1\}^G \tag{11}$$

$$\sum_{g=(l-1)\hat{G}}^{\min(l\hat{G}-1, G)} A_g \leq r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}} \qquad l = 1, \ldots, \lceil G/\hat{G} \rceil \tag{12}$$

$$\overline{V}_{x_j}(\hat{y}_j) = \underbrace{V_{x_j}(\hat{y}_j)}_{\textit{Original votes}} - \underbrace{\sum_{g=1}^{G} A_g \mathbb{I}\{f_g(x_j) = \hat{y}_j\}}_{\textit{Influenced votes}} \qquad \forall x_j \in \mathcal{D}_{test} \tag{13}$$

$$\overline{V}_{x_j}(y) = \underbrace{V_{x_j}(y)}_{\textit{Original votes}} + \underbrace{\sum_{g=1}^{G} A_g \mathbb{I}\{f_g(x_j) \neq y\}}_{\textit{Influenced votes}} \qquad \forall x_j \in \mathcal{D}_{test}, \ \forall y \neq \hat{y}_j \tag{14}$$

The collective robustness is $M - M_{\text{ATK}}$.

We now explain each equation respectively. Eq. (10): the objective function is same as $(\mathbf{P1})$. Eq. (11): $A_1, A_2, \ldots, A_G$ are the binary variables represent the attack, where $A_g = 1$ means that the g-th classifier is influenced. Eq. (12): in hash bagging, r_{ins} insertions, r_{del} deletions and r_{mod} modifications can influence at most $r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$ within each trainset-hash pair. Eq. (13) and Eq. (14): count the minimum/maximum number of votes (after being attacked) for $\hat{y}_j$ and $y \neq \hat{y}_j$. The main advantage of $(\mathbf{P2})$ over $(\mathbf{P1})$ is that, the size of the feasible region is reduced from 2^N to 2^G by exploiting the property of hash bagging, which significantly accelerates the solving process.

4.2. Remarks on Proposition 3

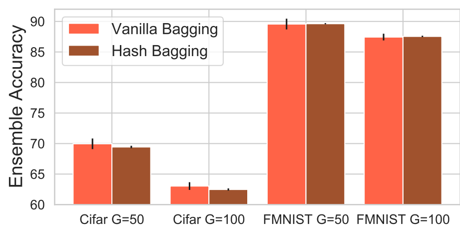
1) Tightness. The collective robustness by $(\mathbf{P2})$ is tight.

2) Simplification. $(\mathbf{P2})$ can be simplified by ignoring the unbreakable predictions within the given poison budget. $\sum_{x_j \in \mathcal{D}_{test}}$ in Eq. (10) can be simplified as $\sum_{x_j \in \Omega}$, and Ω:

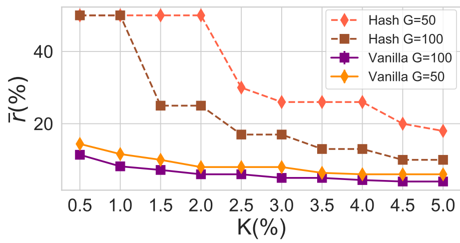
$$\Omega = \{x_j \in \mathcal{D}_{test} : V_{x_j}(\hat{y}_j) - \max_{y \neq \hat{y}_j} [V_{x_j}(y) + \mathbb{I}\{y < \hat{y}_j\}] \leq 2\lceil G/\hat{G} \rceil (r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}})\} \tag{15}$$

3) NP-hardness. $(\mathbf{P2})$ is NP-hard. We can speedup the solution process by decomposition (see Section 3.4).

Implementation. Alg. 1 shows our algorithm for certifying collective robustness. Specifically, we apply simplification and decomposition to accelerate solving $(\mathbf{P2})$.



(a) Comparison on ensemble accuracy ($K = N/G$).



(b) Comparison on $\overline{r}$ on FMNIST.

Figure 2. Comparing hash bagging to vanilla bagging.

Compare hash bagging to vanilla bagging. In Fig. 2a and Fig. 2b, we compare hash bagging to vanilla bagging on the ensemble accuracy and $\overline{r}$ (see Prop. 2) respectively. We observe in Fig. 2a that the ensemble accuracy of hash bagging roughly equals vanilla bagging. Notably, the accuracy variance of hash bagging (over different hash functions) is much smaller than vanilla bagging. We observe in Fig. 2b that $\overline{r}$ of hash bagging is consistently higher than vanilla bagging, especially when K is small. The comparisons suggest that, hash bagging is much more robust than vanilla bagging without sacrificing the ensemble accuracy.

5. Comparisons to Prior Works

We compare to prior works that are tailored to the general data poisoning attack (Ma et al., 2019; Levine & Feizi, 2021; Jia et al., 2021; Jinyuan Jia & Gong, 2022).

Comparison to (Ma et al., 2019) Compared to differential privacy based defense (Ma et al., 2019), hash bagging is more practical for two reasons: I) hash bagging does not require the training algorithm to be differentially private. II) The differential privacy often harms the performance of the learnt model (Duchi et al., 2013), which also limits the scalability of this type of defenses.

Comparison to (Jinyuan Jia & Gong, 2022) Compared to (Jinyuan Jia & Gong, 2022) which derives the sample-wise/collective certificates for kNN/rNN, hash bagging is compatible with different model architectures. Note that the effectiveness of kNN/rNN relies on the assumption: close data are typically similar. Since this assumption might do not hold in some classification tasks, we believe hash bagging is much more practical.

Table 2. Experimental setups in line with literature.

Dataset	Trainset	Testset	Class	Classifier
Bank	35,211	10,000	2	Bayes
Electricity	35,312	10,000	2	SVM
FMNIST	60,000	10,000	10	NIN
CIFAR-10	50,000	10,000	10	NIN (Augmentation)

Comparison to (Jia et al., 2021) (Jia et al., 2021) proposes a bagging variant as a certified defense, which predicts the majority class among the predictions of all the possible sub-classifiers (total N^K sub-classifiers). In practice, training N^K sub-classifiers is often unaffordable, (Jia et al., 2021) approximately estimates the voting distribution by a confidence interval method, which needs to train hundreds of sub-classifiers for a close estimate (G is required to be large). In comparison, hash bagging has no additional constraint. Moreover, unlike our deterministic robustness certificates, its robustness certificates are probabilistic, which have an inevitable failure probability.

Comparison to (Levine & Feizi, 2021) (Levine & Feizi, 2021) propose a partition-based bagging as a certified defense, which is corresponding to **Hash subsampling when** $GK = N$ (Section 10). In comparison, both our collective certification and hash bagging are more general than (Levine & Feizi, 2021). Specifically, hash bagging ablates the constraint that (Levine & Feizi, 2021) places on the bagging hyper-parameters G, K. Our collective certification is able to certify both the tight collective robustness and sample-wise robustness, while (Levine & Feizi, 2021) only considers the sample-wise certificate.

6. Experiments

6.1. Experimental Setups

Datasets and models. We evaluate hash bagging and collective certification on two classic machine learning datasets: Bank (Moro et al., 2014), Electricity (Harries & Wales, 1999), and two image classification datasets: FMNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009). Specifically, for Bank and Electricity, we adapt vanilla bagging/hash bagging to the machine-learning models: Bayes and SVM. For FMNIST and CIFAR-10, we adapt vanilla bagging/hash bagging to the deep-learning model Network in Network (NiN) (Min Lin, 2014). The detailed experimental setups are shown in Table 2.

Implementation details. We use Gurobi 9.0 (Gurobi Optimization, 2021) to solve $(\mathbf{P1})$ and $(\mathbf{P2})$, which can return a lower/upper bound of the objective value within the pre-specific time period. Generally, a longer time can yield a tighter bound. For efficiency, we limit the time to be 2s per sample[1]. More details are in Appendix (Section E).

[1]The solving time for $(\mathbf{P1})$ is universally set to be $2|\mathcal{D}_{test}| =$

Table 3. (Bank: $M = 10,000$; $K = 5\%N$) Certified collective robustness and certified accuracy at $r = 5\%, \ldots, 25\%$ $(\times G)$. r refers to the poison budget $r = r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$. **Sample-wise**: sample-wise certification. **Collective**: collective certification. **CR** and **CA**: certified collective robustness and certified accuracy. $\downarrow \alpha\%$: the relative gap between M_{ATK} guaranteed by collective certification and M_{ATK} of sample-wise certification. NaN: division by zero.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%
20	Vanilla	Sample-wise	CR	3917	0	0	0	0
			CA	3230	0	0	0	0
		Collective	CR	4449	0	0	0	0
			M_{ATK}	$\downarrow 8.74\%$	NaN	NaN	NaN	NaN
			CA	3588	0	0	0	0
			M_{ATK}	$\downarrow 7.47\%$	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9599	9009	7076	5778	4686
			CA	7788	7403	5755	4644	3817
		Collective	CR	**9718**	**9209**	**7270**	**5968**	**4930**
			M_{ATK}	$\downarrow 29.7\%$	$\downarrow 20.2\%$	$\downarrow 6.63\%$	$\downarrow 4.50\%$	$\downarrow 4.59\%$
			CA	**7831**	**7464**	**5806**	**4685**	**3881**
			M_{ATK}	$\downarrow 18.5\%$	$\downarrow 9.89\%$	$\downarrow 2.25\%$	$\downarrow 1.21\%$	$\downarrow 1.52\%$
40	Vanilla	Sample-wise	CR	5250	1870	0	0	0
			CA	4160	1408	0	0	0
		Collective	CR	5385	2166	0	0	0
			M_{ATK}	$\downarrow 2.84\%$	$\downarrow 3.64\%$	NaN	NaN	NaN
			CA	4190	1647	0	0	0
			M_{ATK}	$\downarrow 0.77\%$	$\downarrow 3.58\%$	NaN	NaN	NaN
	Hash	Sample-wise	CR	9638	9301	6401	5376	4626
			CA	7881	7679	5198	4354	3718
		Collective	CR	**9762**	**9475**	**6603**	**5572**	**4796**
			M_{ATK}	$\downarrow 34.2\%$	$\downarrow 24.9\%$	$\downarrow 5.61\%$	$\downarrow 4.24\%$	$\downarrow 3.16\%$
			CA	**7914**	**7718**	**5236**	**4396**	**3751**
			M_{ATK}	$\downarrow 17.2\%$	$\downarrow 9.90\%$	$\downarrow 1.32\%$	$\downarrow 1.13\%$	$\downarrow 0.76\%$

Evaluation metrics and peer methods. Following (Levine & Feizi, 2021; Jia et al., 2021; Jinyuan Jia & Gong, 2022), we evaluate the performance by two metrics: *collective robustness* and *certified accuracy*[2]. We also report the relative gap (denoted by $\downarrow \alpha\%$) between the maximum number of simultaneously changed (correct) predictions guaranteed by the collective certification (denoted by $M^{\text{col}}_{\text{ATK}}$) and that of the sample-wise certification (denoted by $M^{\text{sam}}_{\text{ATK}}$). Namely, $\downarrow \alpha\% = (M^{\text{sam}}_{\text{ATK}} - M^{\text{col}}_{\text{ATK}})/M^{\text{sam}}_{\text{ATK}}$. High α means that the sample-wise certification highly over-estimates the poisoning attack. All the experiments are conducted on the clean dataset without being attacked, which is a common experimental setting for certified defenses (Levine & Feizi, 2021; Jia et al., 2021; Jinyuan Jia & Gong, 2022). We compare *hash bagging* to *vanilla bagging*, and compare *collective certification* to sample-wise certification (Levine & Feizi, 2021). We also compare to *probabilistic certification* (Jia et al., 2021) in Appendix (Section F.2).

$20,000$ seconds. The solving time for $(\mathbf{P2})$ is set to be $2|\Omega|$ for $(\mathbf{P2})$ where Ω is defined in Eq. (15).

[2]We report the minimum number of accurate predictions as the certified accuracy, instead of a ratio, which is in line with the practice in the literature of collective robustness.

Table 4. (Electricity: $M = 10,000$; $K = 5\%N$) Certified collective robustness and certified accuracy.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%
20	Vanilla	Sample-wise	CR	9230	0	0	0	0
			CA	7321	0	0	0	0
		Collective	CR	9348	0	0	0	0
			M_{ATK}	$\downarrow 15.3\%$	NaN	NaN	NaN	NaN
			CA	7394	0	0	0	0
			M_{ATK}	$\downarrow 17.5\%$	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9858	9738	9602	9461	9293
			CA	7681	7621	7538	7462	7362
		Collective	CR	**9915**	**9821**	**9726**	**9608**	**9402**
			M_{ATK}	$\downarrow 40.1\%$	$\downarrow 31.7\%$	$\downarrow 31.1\%$	$\downarrow 27.3\%$	$\downarrow 23.9\%$
			CA	**7701**	**7663**	**7608**	**7547**	**7458**
			M_{ATK}	$\downarrow 34.5\%$	$\downarrow 35.6\%$	$\downarrow 34.8\%$	$\downarrow 30.7\%$	$\downarrow 25.5\%$
40	Vanilla	Sample-wise	CR	9482	8648	0	0	0
			CA	7466	6986	0	0	0
		Collective	CR	9566	8817	0	0	0
			M_{ATK}	$\downarrow 16.2\%$	$\downarrow 12.5\%$	NaN	NaN	NaN
			CA	7513	7086	0	0	0
			M_{ATK}	$\downarrow 16.5\%$	$\downarrow 13.1\%$	NaN	NaN	NaN
	Hash	Sample-wise	CR	9873	9769	9636	9491	9366
			CA	7681	7625	7546	7459	7399
		Collective	CR	**9919**	**9842**	**9755**	**9601**	**9461**
			M_{ATK}	$\downarrow 36.2\%$	$\downarrow 31.6\%$	$\downarrow 32.7\%$	$\downarrow 21.6\%$	$\downarrow 15.0\%$
			CA	**7700**	**7661**	**7613**	**7536**	**7457**
			M_{ATK}	$\downarrow 27.5\%$	$\downarrow 28.8\%$	$\downarrow 32.8\%$	$\downarrow 26.5\%$	$\downarrow 16.5\%$

6.2. Experimental Results

Bank and Electricity. Table 4 and Table 3 report the performances of sample-wise/collective certification on vanilla/hash bagging. There is no need to apply decomposition to these two binary-classification datasets since we can compute the tight certified collective robustness within 10^2 seconds. In comparison, the collective robustness of vanilla bagging drops to zero at $r = 15\%G$, while hash bagging is able to achieve a non-trivial collective robustness at $r = 25\%G$. The values of $\downarrow \alpha\%$ demonstrate that the exact value of M_{ATK} is $5\% \sim 30\%$ less than the values derived from the sample-wise certification. There is an interesting phenomenon that $\downarrow \alpha\%$ generally decreases with r for the number of the candidate poisoning attacks $\binom{N}{r}$ exponentially increases with r. When r is large, there is a high probability to find an attack that can corrupt a high percent of the breakable predictions, thus M_{ATK} guaranteed by the collective certification is close to the sample-wise certification. As we can see, the collective robustness/certified accuracy at $G = 20$ are roughly equal to that of $G = 40$. This is because an insertion/deletion is considered to influence 1 (5%) vote among total 20 votes when $G = 20$, while it can influence 2 (5%) votes among 40 votes for the sub-trainset overlapping. Since the voting distribution of $G = 20$ and $G = 40$ are similar, $G = 20$ and $G = 40$ own the similar collective robustness.

FMNIST and CIFAR-10. Table 5 and Table 6 report the performance of sample-wise/collective certification (with/without decomposition) on vanilla/hash bagging. We adapt decomposition for speedup, because $(\mathbf{P1})$ and $(\mathbf{P2})$

Table 5. (FMNIST: $M = 10,000$; $K = N/G$) Certified collective robustness and certified accuracy. **Decomposition**: collective certification with decomposition.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%
50	Vanilla	Sample-wise	CR	7432	0	0	0	0
			CA	7283	0	0	0	0
		Collective	CR	7727	0	0	0	0
			M_{ATK}	↓ 11.5%	NaN	NaN	NaN	NaN
			CA	7515	0	0	0	0
			M_{ATK}	↓ 13.8%	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9576	9307	8932	8671	8238
			CA	8768	8635	8408	8246	7943
		Collective	CR	**9726**	9410	9024	8761	8329
			M_{ATK}	↓ 35.4%	↓ 14.9%	↓ 8.61%	↓ 6.77%	↓ 5.16%
			CA	**8833**	**8719**	8493	8327	8022
			M_{ATK}	↓ 32.8%	↓ 25.4%	↓ 15.2%	↓ 11.2%	↓ 7.72%
		Decomposition	CR	9666	**9472**	**9124**	**8887**	**8491**
			M_{ATK}	↓ 21.2%	↓ 23.8%	↓ 18.0%	↓ 16.2%	↓ 14.4%
			CA	8812	8716	**8527**	**8385**	**8119**
			M_{ATK}	↓ 22.2%	↓ 24.5%	↓ 21.3%	↓ 19.3%	↓ 17.2%
100	Vanilla	Sample-wise	CR	7548	0	0	0	0
			CA	7321	0	0	0	0
		Collective	CR	8053	0	0	0	0
			M_{ATK}	↓ 20.6%	NaN	NaN	NaN	NaN
			CA	7746	0	0	0	0
			M_{ATK}	↓ 29.4%	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9538	9080	8653	8249	7823
			CA	8554	8316	8049	7797	7486
		Collective	CR	9611	9167	8754	8344	7912
			M_{ATK}	↓ 15.8%	↓ 9.46%	↓ 7.50%	↓ 5.42%	↓ 4.09%
			CA	**8610**	8375	8116	7857	7558
			M_{ATK}	↓ 26.7%	↓ 13.2%	↓ 9.37%	↓ 6.20%	↓ 5.63%
		Decomposition	CR	**9631**	**9232**	**8837**	**8450**	**8036**
			M_{ATK}	↓ 20.1%	↓ 16.5%	↓ 13.6%	↓ 11.5%	↓ 9.78%
			CA	8595	**8407**	**8152**	**7917**	**7639**
			M_{ATK}	↓ 19.5%	↓ 20.3%	↓ 14.4%	↓ 12.4%	↓ 12.0%

Table 6. (CIFAR-10: $M = 10,000$; $K = N/G$) Certified collective robustness and certified accuracy.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%
50	Vanilla	Sample-wise	CR	2737	0	0	0	0
			CA	2621	0	0	0	0
		Collective	CR	3621	0	0	0	0
			M_{ATK}	↓ 12.2%	NaN	NaN	NaN	NaN
			CA	3335	0	0	0	0
			M_{ATK}	↓ 16.3%	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	8221	7268	6067	5320	4229
			CA	6305	5864	5186	4705	3884
		Collective	CR	8393	7428	6204	5435	4290
			M_{ATK}	↓ 9.67%	↓ 5.86%	↓ 3.48%	↓ 2.46%	↓ 1.06%
			CA	6410	5985	5342	4848	4006
			M_{ATK}	↓ 15.2%	↓ 10.7%	↓ 8.62%	↓ 6.24%	↓ 3.92%
		Decomposition	CR	**8694**	**7854**	**6686**	**5912**	**4826**
			M_{ATK}	↓ 26.6%	↓ 21.4%	↓ 15.7%	↓ 12.6%	↓ 10.3%
			CA	**6490**	**6147**	**5553**	**5113**	**4341**
			M_{ATK}	↓ 26.8%	↓ 25.0%	↓ 20.2%	↓ 17.8%	↓ 14.7%
100	Vanilla	Sample-wise	CR	2621	0	0	0	0
			CA	1876	0	0	0	0
		Collective	CR	2657	0	0	0	0
			M_{ATK}	↓ 7.93%	NaN	NaN	NaN	NaN
			CA	2394	0	0	0	0
			M_{ATK}	↓ 11.8%	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	7685	5962	4612	3504	2593
			CA	5396	4571	3787	3008	2315
		Collective	CR	7744	5974	4618	3509	2598
			M_{ATK}	↓ 2.54%	↓ 0.30%	↓ 0.11%	↓ 0.08%	↓ 0.07%
			CA	5475	4650	3825	3030	2330
			M_{ATK}	↓ 9.21%	↓ 4.69%	↓ 1.54%	↓ 0.68%	↓ 0.38%
		Decomposition	CR	**8137**	**6469**	**5061**	**4035**	**2987**
			M_{ATK}	↓ 19.5%	↓ 12.5%	↓ 8.33%	↓ 8.17%	↓ 5.32%
			CA	**5570**	**4841**	**4098**	**3338**	**2635**
			M_{ATK}	↓ 20.3%	↓ 16.0%	↓ 12.6%	↓ 10.2%	↓ 8.12%

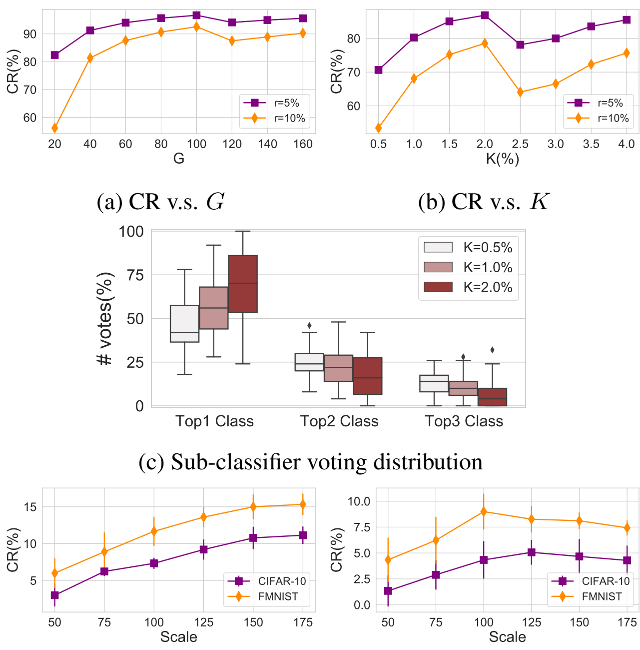
are not solvable over those two ten-classes classification datasets within the limited time. The Δ choices are reported in Appendix (Section F.1). We see that hash bagging consistently outperforms vanilla bagging across different poison budgets. The results demonstrate that: collective certification with decomposition $>$ collective certification $>$ sample-wise certification in terms of the certified collective robustness and the certified accuracy, which suggests collective certification with decomposition is an efficient way to compute the collective robustness certificate.

6.3. Ablation Study

Impact of G. Fig. 3a reports the impact of G on the certified collective robustness of hash bagging. The figure illustrates that as G increases, the collective robustness increases first and then decreases, which reaches the top at $GK = N$. The reason is, as G increases to N/K, the total number of votes increases, thus the attacker needs to modify more votes (higher poison budget) to modify the majority class. As G exceeds the threshold of N/K, despite the growing number of votes, the influence scope of a poisoned sample also increases, as an insertion can simultaneously influence two sub-trainsets when $KG > N$, which causes a slight decline on the certified collective robustness.

Impact of K. Fig. 3b reports the impact of K on the certified collective robustness of hash bagging. Similar to G, as K increases, the collective robustness increases first till $K = N/G$ and then decreases. The insight is, as K rises to N/G, the collective robustness first increases for the improved prediction accuracy of each sub-classifier, because all the sub-classifiers have a higher probability to predict the correct class, as validated in Fig. 3c. As K exceeds the threshold of N/G, the collective robustness decreases for the overlapping between the sub-trainsets, with the same reason of G.

Impact of sub-testset scale Δ. Fig. 3d and Fig. 3e report the impact of Δ on the certified collective robustness of hash bagging at $r = 15\%G$. Specifically, Fig. 3d reports the impact of Δ at no time limit, where we can compute the tight collective robustness for each Δ-size sub-testset. As shown in the figure, the certified collective robustness grows with Δ, but higher Δ also enlarges the computation cost. Thus, Δ controls the trade-off between the collective robustness and the computation cost. Fig. 3d shows the impact of Δ when the time is limited by 2s per sample. We observe that the robustness first increases with Δ and then decreases. The increase is for that we can compute the optimal objective value when Δ is low, and the computed collective robustness lower bound increases with Δ as validated in Fig. 3d. The decrease is because that the required time for solving $(\mathbf{P2})$ is exponential to Δ. Consequently, we can only obtain



(a) CR v.s. G (b) CR v.s. K

(c) Sub-classifier voting distribution

(d) CR v.s. Δ (no time limit) (e) CR v.s. Δ (2s per sample)

Figure 3. Ablation study results on CV datasets. (a): $K = 1\%N$ on FMNIST. (b): $G = 50$ on CIFAR-10. (c): $G = 50$ on CIFAR-10. (d) (e): $G = 50$, $K = 2\%N$, $r = 30\%G$.

a loose bound that is far from the optimal value within the limited time, which causes the decline on the certified collective robustness.

7. Conclusion

Bagging, as a widely-used ensemble learning protocol, owns the certified robustness against data poisoning. In this paper, we derive the tight collective robustness certificate against the global poisoning attack for bagging. Current sample-wise certification is a specific variant of our collective certification. We also propose decomposition to accelerate the solving process. We analyze the upper bound of tolerable poison budget for vanilla bagging. Based on the analysis, we propose hash bagging to improve the certified robustness almost for free. Empirical results show the effectiveness of both our devised collective certification as well as the hash bagging. Our empirical results validate that: i) hash bagging is much robuster; ii) collective certification can yield a stronger collective robustness certificate.

Acknowledgements

This work has been partially supported by the National Key R&D Program of China No. 2020YFB1806700, NSFC Grant 61932014, NSFC Grant 61972246, Project BE2020026 supported by the Key R&D Program of Jiangsu, China, and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., and Gavaldà, R. New ensemble methods for evolving data streams. In *KDD*, 2009.

Biggio, B., Corona, I., Fumera, G., Giacinto, G., and Roli, F. Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In *International workshop on multiple classifier systems*, 2011.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *ICML*, 2012.

Breiman, L. Bagging predictors. *Machine learning*, 1996.

Chen, H., Fu, C., Zhao, J., and Koushanfar, F. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 4658–4664, 2019.

Chinneck, J. W. *Practical Optimization: a Gentle Introduction*. 2015. URL https://www.optimization101.org.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. Local privacy, data processing inequalities, and minimax rates. *arXiv preprint arXiv:1302.3203*, 2013.

Fujishige, S. *Submodular Functions and Optimization*. ISSN. Elsevier Science, 2005. ISBN 9780080461625. URL https://books.google.co.jp/books?id=gdcRXdoV89QC.

Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., and Nepal, S. Strip: a defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference on*, pp. 113–125, 2019.

Geiping, J., Fowl, L., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Madry, A., Li, B., and Goldstein, T. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gurobi Optimization. Gurobi optimizer reference manual, 2021. URL https://www.gurobi.com.

Harries, M. and Wales, N. S. Splice-2 comparative evaluation: Electricity pricing. *Technical report*, 1999.

Huang, W. R., Geiping, J., Fowl, L., Taylor, G., and Goldstein, T. Metapoison: Practical general-purpose clean-label data poisoning. *arXiv preprint arXiv:2004.00225*, 2020.

Jia, J., Cao, X., and Gong, N. Z. Intrinsic certified robustness of bagging against data poisoning attacks. In *AAAI*, 2021.

Jinyuan Jia, Yupei Liu, X. C. and Gong, N. Z. Certified robustness of nearest neighbors against data poisoning and backdoor attacks. In *AAAI*, 2022.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

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

Levine, A. and Feizi, S. Deep partition aggregation: Provable defenses against general poisoning attacks. In *ICLR*, 2021.

Li, Y., Jiang, Y., Li, Z., and Xia, S.-T. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020.

Li, Y., Zhong, H., Ma, X., Jiang, Y., and Xia, S.-T. Few-shot backdoor attacks on visual object tracking. In *ICLR*, 2022.

Liu, Y., Lee, W.-C., Tao, G., Ma, S., Aafer, Y., and Zhang, X. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *CCS '19 Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1265–1282, 2019.

Ma, Y., Zhu, X., and Hsu, J. Data poisoning against differentially-private learners: Attacks and defenses. *IJCAI*, 2019.

Min Lin, Qiang Chen, S. Y. Network in network. In *ICLR*, 2014.

Moro, S., Cortez, P., and Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 2014.

Nelson, B., Barreno, M., Chi, F. J., Joseph, A. D., Rubinstein, B. I., Saini, U., Sutton, C., Tygar, J. D., and Xia, K. Exploiting machine learning to subvert your spam filter. *LEET*, 2008.

Pelofske, E., Hahn, G., and Djidjev, H. Decomposition algorithms for solving np-hard problems on a quantum annealer. *Journal of Signal Processing Systems*, 2020.

Qiao, X., Yang, Y., and Li, H. Defending neural backdoors via generative distribution modeling. In *NeurIPS 2019 : Thirty-third Conference on Neural Information Processing Systems*, pp. 14004–14013, 2019.

Rao, M. Solving some np-complete problems using split decomposition. *Discrete Applied Mathematics*, 2008.

Rosenfeld, E., Winston, E., Ravikumar, P., and Kolter, Z. Certified robustness to label-flipping attacks via randomized smoothing. In *ICML*, 2020.

Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J. P., and Goldstein, T. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, 2021.

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*, 2018.

Steinhardt, J., Koh, P. W., and Liang, P. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3520–3532, 2017.

Tramèr, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.

Tran, B., Li, J., and Madry, A. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, pp. 8000–8010, 2018.

Wang, B., Cao, X., Jia, J.-Y., and Gong, N. Z. On certifying robustness against backdoor attacks via randomized smoothing. *CVPR Workshop*, 2020.

Wang, W., Levine, A., and Feizi, S. Improved certified defenses against data poisoning with (deterministic) finite aggregation. *arXiv preprint arXiv:2202.02628*, 2022.

Wang, Y. and Chaudhuri, K. Data poisoning attacks against online learning. *arXiv preprint arXiv:1808.08994*, 2018.

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

Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., and Roli, F. Is feature selection secure against training data poisoning. In *Proceedings of The 32nd International Conference on Machine Learning*, volume 2, pp. 1689–1698, 2015.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv*, 2017.

Yao, Y., Li, H., Zheng, H., and Zhao, B. Y. Latent backdoor attacks on deep neural networks. In *CCS '19 Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2041–2055, 2019.

Zhang, D., Ye, M., Gong, C., Zhu, Z., and Liu, Q. Black-box certification with randomized smoothing: A functional optimization based framework. *arXiv preprint arXiv:2002.09169*, 2020.

A. Significance of Collective Robustness

The fundamental difference between collective robustness and sample-wise robustness lies in *the setting about the attacker objective*. For sample-wise robustness, the attacker aims to change a single prediction, while for collective robustness, the attacker aims to degrade the overall accuracy of a collection of predictions. Most data poisoning works (Wang & Chaudhuri, 2018; Goldblum et al., 2022; Geiping et al., 2020; Huang et al., 2020; Shafahi et al., 2018; Wang et al., 2022) adopt the latter setting, which aim to maximize the attack success rate (the only metric in Poisoning Benchmark (Schwarzschild et al., 2021)), hinting that collective robustness is more practical. In fact, sample-wise robustness is a special case of collective robustness when the collection size M=1, meaning that collective robustness is more general. In practice, if the model predicts a large collection of images at once, M can be the collection size. If the model intermittently predicts a few images, M can be the total number of the history predictions.

B. Proofs

B.1. Proof of Prop. 1

Proposition 4 (Collective robustness of vanilla bagging). *For testset $\mathcal{D}_{test} = \{x_j\}_{j=0}^{M-1}$, we denote $\hat{y}_j = g(x_j)$ ($j = 0, \ldots, M-1$) the original ensemble prediction, and $\mathcal{S}_i = \{g \mid s_i \in \mathcal{D}_g\}$ the set of the indices of the sub-trainsets that contain s_i. Then, the maximum number of simultaneously changed predictions (denoted by M_{ATK}) under r_{mod} adversarial modifications, is computed by* $(\mathbf{P1})$*:*

$$(\mathbf{P1}):\quad M_{\text{ATK}} = \max_{P_0,\ldots,P_{N-1}} \sum_{x_j \in \mathcal{D}_{test}} \mathbb{I}\left\{\overline{V}_{x_j}(\hat{y}_j) < \max_{y \neq \hat{y}_j}\left[\overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y < \hat{y}_j\}\right]\right\} \tag{16}$$

$$s.t.\quad [P_0, P_1, \ldots, P_{N-1}] \in \{0,1\}^N \tag{17}$$

$$\sum_{i=0}^{N-1} P_i \leq r_{\text{mod}} \tag{18}$$

$$\overline{V}_{x_j}(\hat{y}_j) = \underbrace{V_{x_j}(\hat{y}_j)}_{\textit{Original votes}} - \underbrace{\sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i=1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) = \hat{y}_j\}}_{\textit{Influenced votes}}$$

$$\forall x_j \in \mathcal{D}_{test},\ \hat{y}_j = g(x_j) \tag{19}$$

$$\overline{V}_{x_j}(y) = \underbrace{V_{x_j}(y)}_{\textit{Original votes}} + \underbrace{\sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i=1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) \neq y\}}_{\textit{Influenced votes}}$$

$$\forall x_j \in \mathcal{D}_{test},\ \forall y \in \mathcal{Y}, y \neq \hat{y}_j \tag{20}$$

The collective robustness of vanilla bagging is $M - M_{\text{ATK}}$.

Proof. The collective robustness is defined as the minimum number of simultaneously unchanged predictions, which is equal to the total number of predictions M minus the maximum number of simultaneously changed predictions (denoted as M_{ATK}). To compute the collective robustness, we only need to compute M_{ATK}. M_{ATK} equals the objective value of:

$$\max_{P_0,\ldots,P_{N-1}} \sum_{x_j \in \mathcal{D}_{test}} \mathbb{I}\{\overline{V}_{x_j}(\hat{y}_j) < \max_{y \neq \hat{y}_j}\left[\overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y < \hat{y}_j\}\right]\} \tag{21}$$

where $\overline{V}_{x_j}(y)$ denotes the number of votes for class y when predicting x_j, after being attacked. We now explain each equation. Eq. 16: for the prediction of x_j, the prediction is changed only if *there exists a class that obtains more votes than y_j or the same number of votes but with a smaller index*. We consider three cases for the prediction of x_j:
Case I: $\overline{V}_{x_j}(\hat{y}_j) < \max_{y \neq \hat{y}_j} \overline{V}_{x_j}(y)$: we have $\overline{V}_{x_j}(\hat{y}_j) < \max_{y \neq \hat{y}_j} \overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y < \hat{y}_j\}$, and the prediction of x_j is changed.
Case II: $\overline{V}_{x_j}(\hat{y}_j) = \max_{y \neq \hat{y}_j} \overline{V}_{x_j}(y)$: whether the prediction is changed is determined by $\mathbb{I}\{y < \hat{y}_j\}$. If $\mathbb{I}\{y < \hat{y}_j\} = 0$, meaning that there is no majority class with the smaller index than $\hat{y}_j$, then the prediction $\hat{y}_j$ is unchanged. Otherwise the prediction is changed.
Case III: $\overline{V}_{x_j}(\hat{y}_j) > \max_{y \neq \hat{y}_j} \overline{V}_{x_j}(y)$: we have $\overline{V}_{x_j}(\hat{y}_j) > \max_{y \neq \hat{y}_j} \overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y < \hat{y}_j\}$, and the prediction of x_j is unchanged.
We model the attack as $[P_0, P_1, \ldots, P_{N-1}] \in \{0,1\}^N$ where $P_i = 1$ means that the attacker modifies the i-th training sample s_i. Since the attacker is only allowed to modify r_{mod} samples, we bound $\sum_{i=0}^{N-1} P_i \leq r_{\text{mod}}$. We consider the predictions from the sub-classifiers whose sub-trainsets are changed, as the influenced predictions. Those influenced predictions are considered to be fully controlled by the attacker under our threat model. For the fixed $[P_0, P_1, \ldots, P_{N-1}]$, to maximize the number of simultaneously changed predictions, the optimal strategy is to change all the influenced predictions that equals $\hat{y}_j$ to other classes. Thus we have

$$\overline{V}_{x_j}(\hat{y}_j) = \underbrace{V_{x_j}(\hat{y}_j)}_{\textbf{Original votes}} - \underbrace{\sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i=1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) = \hat{y}_j\}}_{\textbf{Influenced votes}} \tag{22}$$

Note that the attacker can arbitrarily manipulate the influenced predictions, so the number of votes for $y \neq y_j$ is

$$\overline{V}_{x_j}(y) = \underbrace{V_{x_j}(y)}_{\textbf{Original votes}} + \underbrace{\sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i=1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) \neq y\}}_{\textbf{Influenced votes}} \tag{23}$$

Tightness. The collective robustness $M - M_{\text{ATK}}$ is tight for: 1) if the computed collective robustness $M - M_{\text{ATK}}$ is lower than the actual collective robustness, meaning that our computed M_{ATK} is higher than the maximum number of simultaneously changed predictions, which contradicts the fact that we have find an attack that can achieve M_{ATK} under our threat model. 2) if the computed collective robustness $M - M_{\text{ATK}}$ is higher than the actual collective robustness, meaning that our computed M_{ATK} is lower than the maximum number of simultaneously changed predictions, which contradicts the fact that M_{ATK} is the optimal objective value under our threat model. □

B.2. Proof of NP-hardness

We reformulate $(\mathbf{P1})$ into the standard form of a BILP problem, which has been shown to be an NP-Complete problem (Chinneck, 2015), to prove its NP-hardness.

Proof. First of all, we introduce four sets of binary variables:

$$\begin{aligned}
\mathbf{A} &= [A_0, A_1, \ldots, A_i, \ldots, A_{G-1}] \in \{0,1\}^G, \\
\mathbf{Y} &= [Y_0, Y_1, \ldots, Y_j, \ldots, Y_{M-1}] \in \{0,1\}^M, \\
\mathbf{Z} &= [Z_{0,0}, Z_{0,1}, \ldots, Z_{j,l}, \ldots, Z_{M-1,C-1}] \in \{0,1\}^{M\times C}, \\
\mathbf{W} &= [W_0, W_1, \ldots, W_k, \ldots, W_{N-1}] \in \{0,1\}^N,
\end{aligned} \tag{24}$$

where $\mathbf{A}$ denotes the selected sub-classifiers to attack, $\mathbf{Y}$ denotes the attacked test samples, $\mathbf{Z}$ is an auxiliary set of binary variables for the prediction classes, $\mathbf{W}$ represents the poisoned training samples. In according with the main text, G is the number of sub-classifiers, M denotes the number of test samples, C is the number of prediction classes, N represents the number of training samples.

With the notations defined above, we can reformulate $(\mathbf{P1})$ as follows:

$$\text{Maximize} \quad M_{ATK} = \sum_{j=0}^{M-1} Y_j \tag{25}$$

$$s.t. \quad \sum_{k=0}^{N-1} W_k \leq r_{mod} \tag{26}$$

$$\forall i,\ A_i \leq \sum_{k=1}^{N-1} W_k \mathbb{I}\{i \in \mathcal{S}_k\} \tag{27}$$

$$\forall j, l \neq \hat{y}_j, i,\ \text{either } Z_{j,l} \leq 0 \text{ or } V_{x_j}(\hat{y}_j) - V_{x_j}(l) \leq \sum_{i=0}^{G-1} A_i(\mathbb{I}\{f_i(x_j) \neq l\} + \mathbb{I}\{f_i(x_j) = \hat{y}_j\}) \tag{28}$$

$$\forall j,\ \text{either } Y_j \leq 0 \text{ or } \sum_{l=0}^{C-1} Z_{j,l} \geq 2 \tag{29}$$

We now explain each equation respectively. Eq. (25) is the variant of Eq. (16), denoting that our objective is to maximize the number of attacked test samples. Eq. (26) shares the same meaning as Eq. (18), which restricts the number of poisoned training samples to be less than r_{mod}. Eq. (27) restricts the selected sub-classifiers should be in $\bigcup_{\forall k, P_k=1} \mathcal{S}_k$. Eq. (28) shows that $Z_{j,l}$ could be 1 only when the ensemble prediction of the test sample j can be changed from $\hat{y}_j$ to l (we ignore the minimum index constraint for simplicity). Eq. (29) shows that Y_j could be 1 (the test sample j is attacked successfully) only when there exists some classes that the ensemble prediction can be changed to. We use the equation $\sum_{l=0}^{C-1} Z_{i,l} \geq 2$ since we always have $Z_{j,\hat{y}_j} = 1$.

The formulation above has been in the standard form of a BILP problem, except the "either...or..." clause. Using the transformation trick in (Chinneck, 2015), e.g.

$$\text{either} \quad x_1 + x_2 \leq 4 \quad \text{or} \quad x_1 + 1.5x_2 \leq 6$$

is equal to

$$\begin{aligned}
x_1 + x_2 &\leq 4 + My \\
x_1 + 1.5x_2 &\leq 6 + M(1-y)
\end{aligned}$$

where M is a large number, y is an auxiliary introduced binary variable.

Thus, we can transform Eq. (28) and Eq. (29) into the standard form of constraints by introducing additionally number and binary variables, which means that $(\mathbf{P1})$ can be transformed into the standard form of a BILP problem. Now we can tell that $(\mathbf{P1})$ is an NP-hard problem. □

B.3. Proof of Prop. 2

Proposition 5 (Upper bound of tolerable poison budget)**.** *Given $\mathcal{S}_i$ $(i = 0, \ldots, N-1)$, the upper bound of the tolerable poisoned samples (denoted by $\overline{r}$) is*

$$\overline{r} = \min |\Pi| \ s.t. \ |\bigcup_{i\in\Pi} \mathcal{S}_i| > G/2 \tag{30}$$

which equals the minimum number of training samples that can influence more than a half of sub-classifiers.

Proof. We prove that, $\forall r_{\text{mod}} \geq \overline{r}$, the collective robustness computed from $(\mathbf{P1})$ is 0. Specifically, when $r_{\text{mod}} \geq \overline{r}$, if we choose to poison the training samples whose indices are within Π, for all $\hat{y}_j$, the number of votes for the original ensemble prediction $\hat{y}_j$ is

$$\overline{V}_{x_j}(\hat{y}_j) = V_{x_j}(\hat{y}_j) - \sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i=1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) = \hat{y}_j\} \tag{31}$$

$$= V_{x_j}(\hat{y}_j) - \sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{i\in\Pi} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) = \hat{y}_j\} \tag{32}$$

$$= \sum_{g=0}^{G-1} \mathbb{I}\{f_g(x_j) = \hat{y}_j\} - \sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{i\in\Pi} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) = \hat{y}_j\} \tag{33}$$

$$\leq \sum_{g=0}^{G-1} \mathbb{I}\{g \notin \bigcup_{i\in\Pi} \mathcal{S}_i\} \tag{34}$$

$$< \frac{G}{2} \tag{35}$$

The number of votes for other classes $y \neq \hat{y}_j$ is

$$\overline{V}_{x_j}(y) = V_{x_j}(y) + \sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{\forall i, P_i=1} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) \neq y\} \tag{36}$$

$$= V_{x_j}(y) + \sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{i\in\Pi} \mathcal{S}_i\}\mathbb{I}\{f_g(x_j) \neq y\} \tag{37}$$

$$\geq \sum_{g=0}^{G-1} \mathbb{I}\{g \in \bigcup_{i\in\Pi} \mathcal{S}_i\} \tag{38}$$

$$> \frac{G}{2} \tag{39}$$

We have

$$\overline{V}_{x_j}(\hat{y}_j) - \max_{y\neq\hat{y}_j}\left[\overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y < \hat{y}_j\}\right] \tag{40}$$

$$\leq \frac{G}{2} - \frac{G}{2} + 1 - \frac{1}{2} \tag{41}$$

$$< 0 \tag{42}$$

Therefore, $\forall x_j$, the prediction $\hat{y}_j$ is considered to be corrupted. The certified collective robustness is 0. □

B.4. Proof of Prop. 3

Proposition 6 (Certified collective robustness of hash bagging)**.** *For testset $\mathcal{D}_{test} = \{x_j\}_{j=0}^{M-1}$, we denote $\hat{y}_j = g(x_j)$ ($j = 0, \ldots, M-1$) the ensemble prediction. The maximum number of simultaneously changed predictions (denoted by M_{ATK}) under r_{ins} insertions, r_{del} deletions and r_{mod} modifications, is computed by* $(\mathbf{P2})$*:*

$$(\mathbf{P2}): \quad M_{\text{ATK}} = \max_{A_0,\ldots,A_{G-1}} \sum_{x_j\in\mathcal{D}_{test}} \mathbb{I}\Big\{\overline{V}_{x_j}(\hat{y}_j) < \max_{y\neq\hat{y}_j}\left[\overline{V}_{x_j}(y) + \frac{1}{2}\mathbb{I}\{y<\hat{y}_j\}\right]\Big\} \tag{43}$$

$$s.t. \quad [A_0, A_1, \ldots, A_{G-1}] \in \{0,1\}^G \tag{44}$$

$$\sum_{g=(l-1)\hat{G}}^{l\hat{G}-1} A_g \leq r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}} \quad l = 1, \ldots, \lceil G/\hat{G}\rceil \tag{45}$$

$$\overline{V}_{x_j}(\hat{y}_j) = \underbrace{V_{x_j}(\hat{y}_j)}_{\textit{Original votes}} - \underbrace{\sum_{g=1}^{G} A_g \mathbb{I}\{f_g(x_j) = \hat{y}_j\}}_{\textit{Influenced votes}} \quad \forall x_j \in \mathcal{D}_{test} \tag{46}$$

$$\overline{V}_{x_j}(y) = \underbrace{V_{x_j}(y)}_{\textit{Original votes}} + \underbrace{\sum_{g=1}^{G} A_g \mathbb{I}\{f_g(x_j) \neq y\}}_{\textit{Influenced votes}} \quad \forall x_j \in \mathcal{D}_{test},\ \forall y \neq \hat{y}_j \tag{47}$$

The collective robustness is $M - M_{\text{ATK}}$.

Proof. In fact, $(\mathbf{P2})$ is a simplified version of $(\mathbf{P1})$ which exploits the properties of hash bagging. $(\mathbf{P2})$ is mainly different from $(\mathbf{P1})$ in Eq. (17) and Eq. (18). Specifically, in $(\mathbf{P2})$, the poisoning attack is expressed as $[A_0, A_1, \ldots, A_{G-1}]$, where A_g denotes whether the g-th sub-classifier is influenced, instead of whether the g-th sample is modified in $(\mathbf{P1})$. Based on the property of hash bagging, each trainset-hash pair $(\mathcal{D}_{train}, \text{Hash}(\cdot))$ is partitioned into $\lfloor N/K \rfloor$ disjoint sub-trainsets. Therefore, r_{ins} insertions, r_{del} deletions and r_{mod} modifications can influence at most $r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$ sub-trainsets within each trainset-hash pair, as shown in Eq. (45).

Tightness. When $N \leq GK$, the proof of tightness is the same as that for $(\mathbf{P1})$. Next, we prove that our robustness is tight. In particular, we prove: i) the collective robustness computed from $(\mathbf{P2})$ is a lower bound. ii) the collective robustness $M - M_{\text{ATK}}$ by $(\mathbf{P2})$ is an upper bound.

i) For arbitrary r_{ins} insertions, r_{del} deletions and r_{mod} modifications can influence at most $r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$ sub-trainsets within each trainset-hash pair. Therefore, for any poisoning attack (r_{ins} insertions, r_{del} deletions and r_{mod} modifications), we can denote it by $[A_0, A_1, \ldots, A_{G-1}]$:

$$[A_0, A_1, \ldots, A_{G-1}] \in \{0,1\}^G$$
$$\sum_{g=(l-1)\hat{G}}^{l\hat{G}-1} A_g \leq r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$$

The poisoning attacks denoted by Eq. (44), Eq. (45) are

Table 7. Method comparison. **Model, Training, Bagging** denote whether the defense is compatible with various classifier models, training algorithms and general forms of bagging. **Sample-wise, Collective, Deterministic** denote whether the method can provide sample-wise robustness certificates, collective robustness certificates and deterministic robustness certificates.

Methods	Certified Defense			Robustness Certification		
	Model	Training	Bagging	Sample-wise	Collective	Deterministic
(Levine & Feizi, 2021)	✓	✓	✗	✓	✗	✓
(Jia et al., 2021)	✓	✓	✗	✓	✗	✗
(Ma et al., 2019)	✓	✗	–	✓	✗	✗
(Jinyuan Jia & Gong, 2022)	✗	✗	–	✓	✓	✓
Ours	✓	✓	✓	✓	✓	✓

stronger than the practical poisoning attacks. Therefore, the collective robustness computed from $(\mathbf{P2})$ is a lower bound.

ii) First we denote $\{A_{(l-1)\hat{G}+\beta_{l,o}} \mid o = 0, \dots, r-1;\ l = 1, \dots, \lceil G/\hat{G} \rceil;\ \ \beta_{l,o} \in [0, \hat{G}-1]\}$ the influenced sub-classifiers ($A_{(l-1)\hat{G}+\beta_{l,o}} = 1$). We construct an insertion attack as follow: we insert r new samples (denoted by $\hat{s}_o :\ o = 0, \dots, r-1$), where the hash value of $\hat{s}_o$ computed by the l-th hash function mod $\hat{G}$ is $\beta_{l,o}$. We can achieve M_{ATK} within poison budget r. Therefore, the collective robustness $M - M_{\text{ATK}}$ is an upper bound. $\square$

C. Certification Gap

We intuitively show the gap between the collective robustness guaranteed by our collective certification and that of the sample-wise certification in Fig. 4.

D. Comparison Overview

Table 7 presents an overview of the theoretical comparisons to other certified defenses that are tailored to the general data poisoning attack.

E. Implementation Details

All the experiments are conducted on CPU (16 Intel(R) Xeon(R) Gold 5222 CPU @ 3.80GHz) and GPU (one NVIDIA RTX 2080 Ti).

E.1. Training Algorithm

Alg. 2 summarizes our training process for hash bagging. It needs to set the random seed for reproducible training and train the sub-classifiers on the hash-based sub-trainsets.

E.2. Dataset Information

Table 2 shows our experimental setups in details.

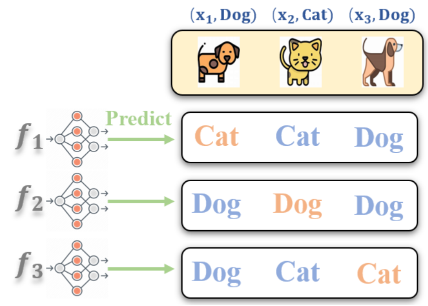


Figure 4. An example to illustrate the gap between the sample-wise certificate and the collective certificate. Suppose the sub-classifiers are $f_1(x), f_2(x), f_3(x)$, and the testing samples are x_1, x_2, x_3. The predictions Cat/Dog are correct, and Cat/Dog are wrong. Consider an attacker (poison budget is 1) can control an arbitrary sub-classifier. **Sample-wise certificate**: we consider $g(x_1), g(x_2), g(x_3)$ independently. To change $g(x_1)/g(x_2)/g(x_3)$, the attacker can flip $f_2(x_1)/f_3(x_2)/f_1(x_3)$ respectively. Therefore, all the three predictions are not robust and the sample-wise robustness is 0. **Collective certificate**: we consider $g(x_1), g(x_2), g(x_3)$ collectively. If the attacker poisons $f_1/f_2/f_3$, the prediction $g(x_1)/g(x_2)/g(x_3)$ is unchangeable respectively. Thus the collective robustness is 1.

Algorithm 2: Train the sub-classifiers.

Input: trainset $\mathcal{D}_{train}$, number of sub-trainsets G, sub-trainset size K, hash functions $\mathrm{Hash_h}(\cdot) : \mathrm{h} = 0, 1, \dots$.

Construct G sub-trainsets $\mathcal{D}_g$ $(g = 0, \dots, G-1)$ based on Eq. (9); # Hash-based subsampling.

Set the random seed for training; # Reproducible training.

Train the sub-classifiers f_g on $\mathcal{D}_g$ $(g = 0, \dots, G-1)$;

Output: The trained sub-classifiers $\{f_g\}_{g=1}^{G}$.

Bank[3] dataset consists of 45,211 instances of 17 attributes (including both numeric attributes and categorical attributes) in total. Each of the instances is labeled to two classes, "yes" or "no". We partition the dataset to 35,211 for training and 10,000 for testing. We use SVM as the sub-classifier architecture.

Electricity[4] has 45,312 instances of 8 numeric attributes. Each of the instances is labeled to two classes, "up" or "down". We partition the dataset to 35,312 for training and 10,000 for testing. Following (Bifet et al., 2009), we use Bayes as the sub-classifier architecture for ensemble.

[3] `https://archive.ics.uci.edu/ml/datasets/Bank+Marketing`.

[4] `https://datahub.io/machine-learning/electricity`.

Table 8. Impact of Δ ($K = N/G$). The numerical results record the mean and variance of the certified robustness ratio. NaN: The number of breakable test samples $M \leq 6|\Delta|$ so we cannot calculate valid variance for CR ratios.

Dataset	G	Δ	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
FMNIST	50	50	**13.00± 2.76**	15.00± 5.86	15.00± 5.98	11.66± 3.54	6.34± 3.54	4.34 ± 2.14	1.00± 1.00	0.66± 0.94	0.00± 0.00	0.00± 0.00
		75	NaN	**19.56± 3.97**	**18.22± 5.59**	**16.22± 2.92**	10.89± 3.88	6.22± 2.27	4.67± 1.84	1.11± 0.92	0.00± 0.00	0.00± 0.00
		100	NaN	18.17± 0.74	15.50± 1.71	13.17± 3.02	**12.47± 1.34**	9.00± 1.73	6.5± 1.61	3.17± 1.34	0.00± 0.00	0.00± 0.00
		125	NaN	NaN	12.00± 1.37	11.33± 0.72	10.8± 1.10	**8.26± 1.28**	**7.2± 1.53**	4.67± 1.07	0.00± 0.00	0.00± 0.00
		175	NaN	NaN	NaN	9.61± 1.01	8.38± 0.63	7.43± 0.74	5.81± 0.95	**5.62± 1.21**	0.38± 0.42	0.00± 0.00
		200	NaN	NaN	NaN	8.66± 1.25	8.08± 0.67	7.08± 1.06	5.66± 1.18	5.25± 0.75	**0.84± 0.75**	0.00± 0.00
	100	50	**13.34±2.74**	**13.34±3.40**	8.00±5.04	8.66±4.42	4.00±3.26	1.66±1.38	2.00±2.30	0.00±0.00	0.00±0.00	0.00±0.00
		100	NaN	11.50±1.71	**10.34±1.70**	**10.00±1.41**	**7.84±2.03**	**5.50±3.0**	4.33±1.97	1.00±1.15	0.00±0.00	0.00±0.00
		150	NaN	NaN	7.89±1.46	7.45±1.51	5.45±1.18	4.78±0.25	**4.78±0.6**	2.45±0.99	0.00±0.00	0.00±0.00
		200	NaN	NaN	6.25±0.56	5.25±0.75	4.50±1.08	4.42±0.78	3.50±0.81	2.34±0.98	0.42±0.34	0.00±0.00
		250	NaN	NaN	NaN	5.20±0.86	4.27±0.72	3.53±0.71	3.47±0.79	**2.47±1.07**	0.60±0.24	0.00±0.00
		300	NaN	NaN	NaN	NaN	4.00±0.58	3.50±0.37	2.44±0.85	2.44±0.85	**0.89±0.25**	0.00±0.00
CIFAR-10	50	50	15.33± 5.73	10.33± 2.43	9.00± 4.73	7.67± 2.13	5.33± 3.94	1.33± 1.49	0.33± 0.75	0.00± 0.00	0.00± 0.00	0.00± 0.00
		75	**17.56± 0.92**	**11.56± 2.73**	12.00± 2.88	**10.67± 1.53**	7.78± 2.23	2.89± 1.43	0.22± 0.49	0.00± 0.00	0.00± 0.00	0.00± 0.00
		100	14.50± 3.69	10.33± 0.74	**12.00± 1.41**	9.50± 2.06	**8.50± 0.96**	4.33± 1.80	1.16± 1.46	0.00± 0.00	0.00± 0.00	0.00± 0.00
		125	11.87± 1.56	9.33± 1.64	10.00± 1.37	8.00± 0.92	7.73± 0.88	**5.07± 1.19**	2.00± 1.44	0.80± 0.80	0.00± 0.00	0.00± 0.00
		175	10.00± 1.83	9.33± 0.63	7.24± 1.13	6.67± 1.03	5.9± 0.63	4.29± 1.43	**3.05± 1.17**	1.14± 0.74	0.00± 0.00	0.00± 0.00
		200	8.17± 3.41	8.33± 0.63	7.17± 0.94	5.83± 0.69	5.33± 0.47	4.25± 0.95	**2.67± 0.95**	**2.00± 0.87**	0.00± 0.00	0.00± 0.00
	100	50	**11.00± 3.42**	**9.66±3.54**	**5.66±4.82**	**3.66±2.42**	2.00±1.64	0.66±0.94	0.00±0.00	0.00±0.00	0.00±0.00	0.00 ± 0.00
		100	7.67± 2.56	5.50±1.89	5.33±2.21	**5.00±1.82**	**4.50±2.14**	**2.50±0.96**	0.17±0.37	0.00±0.00	0.00±0.00	0.00±0.00
		150	7.11± 1.25	5.55±0.63	4.22±0.49	3.55±0.83	2.11±0.46	1.78±0.31	0.89±0.49	0.00±0.00	0.00±0.00	0.00±0.00
		200	5.34± 2.32	5.58±0.34	4.34±0.80	2.92±0.34	2.75±0.48	1.58±0.18	1.00±0.50	0.00±0.00	0.00±0.00	0.00±0.00
		250	3.93± 2.51	4.53±1.32	4.13±0.72	2.87±0.43	2.20±0.30	1.67±0.36	**1.06±0.30**	**0.13±0.19**	0.00±0.00	0.00±0.00
		300	5.44± 0.46	4.61±0.65	3.67±0.54	2.78±0.31	2.17±0.17	1.56±0.16	1.00±0.35	0.06±0.12	0.00±0.00	0.00±0.00

Fashion-MNIST[5](FMNIST) consists of 60,000 training instances and 10,000 testing instances. Each is a 28×28 grayscale image, which is labeled to one of ten classes. We follow the model architecture, Network in Network (NiN) (Min Lin, 2014) used in (Levine & Feizi, 2021) as the sub-classifier architecture for ensemble.

CIFAR-10[6] contains 60,000 images of size 32×32×3 pixels, 50,000 for training and 10,000 for testing. Each of the instances is labeled to one of ten classes. We follow (Levine & Feizi, 2021) to use NiN with full data augmentation as the sub-classifier architecture for ensemble.

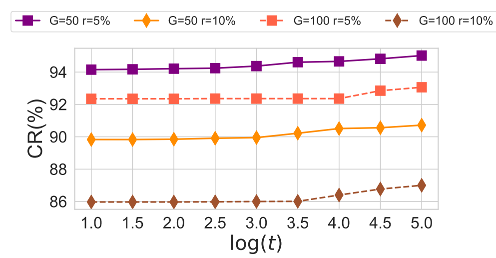


Figure 5. Impact of t on CIFAR-10 ($K = N/G$).

F. More Experimental Results

F.1. More Ablation Studies

Impact of Sub-Problem Scale Δ Table 8 reports the impact of Δ on the collective robustness of hash bagging when the time is limited to 2s per sample. The collective robustness is reported in the form of a percentage. Namely, 13.00 ± 2.76 means that, there are 13% predictions are certifiably simultaneously robust in average, with the variance 2.76, which is to compute over 6 randomly selected Δ-size sub-problems. We can empirically tell that when the poison budget r is low, a large Δ might prevent us from computing the optimal objective value. When the poison budget r is high, we can easily find an attack to corrupt a large portion of predictions for the small Δ-size sub-testset, while finding a better solution for the large Δ-size sub-problem at the meantime. As a result, the optimal Δ increases with the poison budget r as shown in Table 8.

Impact of Solving Time t Fig. 5 reports the impact of solving time t on the certified collective robustness of hash bagging if we do not apply decomposition, on CIFAR-10. We observe that the collective robustness roughly increases linearly with $\log(t)$, which suggests that directly increasing the solving time is not an effective way to improve the certified collective robustness.

F.2. More Evaluation Results

Table 9, Table 10, Table 11, Table 12 report the detailed empirical results on Bank, Electricity, FMNIST, CIFAR-10, respectively. Specifically, we also compare to the probabilistic certification method (Jia et al., 2021), where the confidence is set to be 0.999 (the official implementation), and the num-

[5]https://github.com/zalandoresearch/fashion-mnist.

[6]https://www.cs.toronto.edu/~kriz/cifar.html.

ber of sub-classifiers is set to be the same number used in the other certifications for the computational fairness. Note that the probabilistic certification cannot be applied to hash bagging, because it assumes that the sub-trainsets are randomly subsampled (with replacement) from the trainset. The empirical results demonstrate that, collective certification $>$ sample-wise certification $>$ probabilistic certification in terms of the certified collective robustness and the certified accuracy, on vanilla bagging. We observe that probabilistic certification performs poorly when G is small, because the confidence interval estimation in probabilistic certification highly relies on the number of sub-classifiers.

G. Limitations

As a defense against data poisoning, the main limitation of bagging is that we need to train multiple sub-classifiers to achieve a high certified robustness, because bagging actually exploits the majority voting based redundancy to trade for the robustness. Moreover, our collective certification does not take into account any property of the sub-classifiers, because our certification is agnostic towards the classifier architectures. Therefore, if we can specify the model architecture, we can further improve the certified robustness by exploiting the intrinsic property of the base model. Our collective certification needs to solve a costly NP-hard problem. A future direction is to find a collective robustness lower bound in a more effective way.

Table 9. (Bank: $M = 10,000$; $K = 5\%N$) Comparison on the certified collective robustness and the certified accuracy at $r = 5\%, \dots, 50\%$ ($\times G$), where $r = r_{\text{ins}} + r_{\text{del}} + 2r_{\text{mod}}$ refers to the poison budget. **Sample-wise** and **Collective** refer to sample-wise and collective certification respectively. **Probabilistic** refers to the probabilistic certification proposed in (Jia et al., 2021). **CR** and **CA** refer to the certified collective robustness and the certified accuracy respectively. $\downarrow \alpha\%$ denotes the relative gap between M_{ATK} guaranteed by the collective certification and M_{ATK} of the sample-wise certification. NaN: division by zero.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
20	Vanilla	Sample-wise	CR	3917	0	0	0	0	0	0	0	0	0
			M_{ATK}	6083	10000	10000	10000	10000	10000	10000	10000	10000	10000
			CA	3230	0	0	0	0	0	0	0	0	0
			M_{ATK}	4790	8020	8020	8020	8020	8020	8020	8020	8020	8020
		Probabilistic	CR	0	0	0	0	0	0	0	0	0	0
			CA	0	0	0	0	0	0	0	0	0	0
		Collective	CR	4449	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow 8.74\%$	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	3588	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow 7.47\%$	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9599	9009	7076	5778	4686	3772	2880	2157	1485	289
			M_{ATK}	401	991	2924	4222	5314	6228	7120	7843	8515	9711
			CA	7788	7403	5755	4644	3817	3036	2283	1659	1106	284
			M_{ATK}	232	617	2265	3376	4203	4984	5737	6361	6914	7736
		Collective	CR	**9718**	**9209**	**7270**	**5968**	**4930**	**3915**	**3076**	**2294**	**1503**	**289**
			M_{ATK}	$\downarrow 29.7\%$	$\downarrow 20.2\%$	$\downarrow 6.63\%$	$\downarrow 4.50\%$	$\downarrow 4.59\%$	$\downarrow 2.30\%$	$\downarrow 2.75\%$	$\downarrow 1.75\%$	$\downarrow 0.21\%$	$\downarrow 0.00\%$
			CA	**7831**	**7464**	**5806**	**4685**	**3881**	**3091**	**2349**	**1689**	**1112**	**284**
			M_{ATK}	$\downarrow 18.5\%$	$\downarrow 9.89\%$	$\downarrow 2.25\%$	$\downarrow 1.21\%$	$\downarrow 1.52\%$	$\downarrow 1.10\%$	$\downarrow 1.15\%$	$\downarrow 0.47\%$	$\downarrow 0.09\%$	$\downarrow 0.00\%$
40	Vanilla	Sample-wise	CR	5250	1870	0	0	0	0	0	0	0	0
			M_{ATK}	4750	8130	10000	10000	10000	10000	10000	10000	10000	10000
			CA	4160	1408	0	0	0	0	0	0	0	0
			M_{ATK}	3913	6665	8073	8073	8073	8073	8073	8073	8073	8073
		Probabilistic	CR	1509	1095	751	0	0	0	0	0	0	0
			CA	1049	705	407	0	0	0	0	0	0	0
		Collective	CR	5385	2166	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow 2.84\%$	$\downarrow 3.64\%$	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	4190	1647	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow 0.77\%$	$\downarrow 3.58\%$	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9638	9301	6401	5376	4626	4061	3398	2551	1497	115
			M_{ATK}	362	699	3599	4624	5374	5939	6602	7449	8503	9885
			CA	7881	7679	5198	4354	3718	3229	2693	1976	1037	114
			M_{ATK}	192	394	2875	3719	4355	4844	5380	6097	7036	7959
		Collective	CR	**9762**	**9475**	**6603**	**5572**	**4796**	**4209**	**3562**	**2665**	**1523**	**115**
			M_{ATK}	$\downarrow 34.2\%$	$\downarrow 24.9\%$	$\downarrow 5.61\%$	$\downarrow 4.24\%$	$\downarrow 3.16\%$	$\downarrow 2.49\%$	$\downarrow 2.48\%$	$\downarrow 1.53\%$	$\downarrow 0.30\%$	$\downarrow 0.00\%$
			CA	**7914**	**7718**	**5236**	**4396**	**3751**	**3257**	**2720**	**2010**	**1049**	**114**
			M_{ATK}	$\downarrow 17.2\%$	$\downarrow 9.90\%$	$\downarrow 1.32\%$	$\downarrow 1.13\%$	$\downarrow 0.76\%$	$\downarrow 0.58\%$	$\downarrow 0.50\%$	$\downarrow 0.56\%$	$\downarrow 0.17\%$	$\downarrow 0.00\%$

Table 10. (Electricity: $M = 10,000$; $K = 5\%N$) Certified collective robustness and certified accuracy.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
20	Vanilla	Sample-wise	CR	9230	0	0	0	0	0	0	0	0	0
			M_{ATK}	770	10000	10000	10000	10000	10000	10000	10000	10000	10000
			CA	7321	0	0	0	0	0	0	0	0	0
			M_{ATK}	418	7739	7739	7739	7739	7739	7739	7739	7739	7739
		Probabilistic	CR	0	0	0	0	0	0	0	0	0	0
			CA	0	0	0	0	0	0	0	0	0	0
		Collective	CR	9348	0	0	0	0	0	0	0	0	0
			M_{ATK}	↓ 15.3%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	7394	0	0	0	0	0	0	0	0	0
			M_{ATK}	↓ 17.5%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9858	9738	9602	9461	9293	9121	8928	8656	8294	2597
			M_{ATK}	142	262	398	539	707	879	1072	1344	1706	7403
			CA	7681	7621	7538	7462	7362	7266	7157	6998	6767	2198
			M_{ATK}	58	118	201	277	377	473	582	741	972	5541
		Collective	CR	**9915**	**9821**	**9726**	**9608**	**9402**	**9302**	**9122**	**8829**	**8449**	**2605**
			M_{ATK}	↓ 40.1%	↓ 31.7%	↓ 31.1%	↓ 27.3%	↓ 23.9%	↓ 20.6%	↓ 18.1%	↓ 12.9%	↓ 9.08%	↓ 0.11%
			CA	**7701**	**7663**	**7608**	**7547**	**7458**	**7366**	**7265**	**7102**	**6856**	**2200**
			M_{ATK}	↓ 34.5%	↓ 35.6%	↓ 34.8%	↓ 30.7%	↓ 25.5%	↓ 21.1%	↓ 18.6%	↓ 14.0%	↓ 9.16%	↓ 0.04%
40	Vanilla	Sample-wise	CR	9482	8648	0	0	0	0	0	0	0	0
			M_{ATK}	518	1352	10000	10000	10000	10000	10000	10000	10000	10000
			CA	7466	6986	0	0	0	0	0	0	0	0
			M_{ATK}	284	764	7750	7750	7750	7750	7750	7750	7750	7750
		Probabilistic	CR	8489	8248	7848	0	0	0	0	0	0	0
			CA	6892	6742	6506	0	0	0	0	0	0	0
		Collective	CR	9566	8817	0	0	0	0	0	0	0	0
			M_{ATK}	↓ 16.2%	↓ 12.5%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	7513	7086	0	0	0	0	0	0	0	0
			M_{ATK}	↓ 16.5%	↓ 13.1%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9873	9769	9636	9491	9366	9213	9022	8774	8434	2516
			M_{ATK}	127	231	364	509	634	787	978	1226	1566	7484
			CA	7681	7625	7546	7459	7399	7316	7204	7065	6860	2142
			M_{ATK}	69	125	204	291	351	434	546	685	890	5608
		Collective	CR	**9919**	**9842**	**9755**	**9601**	**9461**	**9312**	**9127**	**8883**	**8537**	**2524**
			M_{ATK}	↓ 36.2%	↓ 31.6%	↓ 32.7%	↓ 21.6%	↓ 15.0%	↓ 12.6%	↓ 10.7%	↓ 8.89%	↓ 6.58%	↓ 0.11%
			CA	**7700**	**7661**	**7613**	**7536**	**7457**	**7378**	**7274**	**7140**	**6918**	**2145**
			M_{ATK}	↓ 27.5%	↓ 28.8%	↓ 32.8%	↓ 26.5%	↓ 16.5%	↓ 14.3%	↓ 12.8%	↓ 10.9%	↓ 6.52%	↓ 0.05%

Table 11. (FMNIST: $M = 10,000$; $K = N/G$) Certified collective robustness and certified accuracy. **Decomposition**: collective certification with decomposition.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
50	Vanilla	Sample-wise	CR	7432	0	0	0	0	0	0	0	0	0
			M_{ATK}	2568	10000	10000	10000	10000	10000	10000	10000	10000	10000
			CA	7283	0	0	0	0	0	0	0	0	0
			M_{ATK}	1683	8966	8966	8966	8966	8966	8966	8966	8966	8966
		Probabilistic	CR	6897	6633	5918	5214	0	0	0	0	0	0
			CA	6799	6557	5891	5201	0	0	0	0	0	0
		Collective	CR	7727	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 11.5%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	7515	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 13.8%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9576	9307	8932	8671	8238	7929	7456	7051	6146	308
			M_{ATK}	424	693	1068	1329	1762	2071	2544	2949	3854	9692
			CA	8768	8635	8408	8246	7943	7700	7295	6943	6107	308
			M_{ATK}	198	331	558	720	1023	1266	1671	2023	2859	8658
		Collective	CR	**9726**	9410	9024	8761	8329	8024	7525	7126	6277	329
			M_{ATK}	$\downarrow$ 35.4%	$\downarrow$ 14.9%	$\downarrow$ 8.61%	$\downarrow$ 6.77%	$\downarrow$ 5.16%	$\downarrow$ 4.59%	$\downarrow$ 2.71%	$\downarrow$ 2.54%	$\downarrow$ 3.40%	$\downarrow$ 0.22%
			CA	**8833**	**8719**	8493	8327	8022	7780	7370	7020	6247	327
			M_{ATK}	$\downarrow$ 32.8%	$\downarrow$ 25.4%	$\downarrow$ 15.2%	$\downarrow$ 11.2%	$\downarrow$ 7.72%	$\downarrow$ 6.32%	$\downarrow$ 4.49%	$\downarrow$ 3.81%	$\downarrow$ 4.90%	$\downarrow$ 0.22%
		Decomposition	CR	9666	**9472**	**9124**	**8887**	**8491**	**8196**	**7672**	**7287**	**6300**	**308**
			M_{ATK}	$\downarrow$ 21.2%	$\downarrow$ 23.8%	$\downarrow$ 18.0%	$\downarrow$ 16.2%	$\downarrow$ 14.4%	$\downarrow$ 12.9%	$\downarrow$ 8.49%	$\downarrow$ 8.00%	$\downarrow$ 4.00%	$\downarrow$ 0.00%
			CA	8812	8716	**8527**	**8385**	**8119**	**7892**	**7491**	**7150**	**6271**	**308**
			M_{ATK}	$\downarrow$ 22.2%	$\downarrow$ 24.5%	$\downarrow$ 21.3%	$\downarrow$ 19.3%	$\downarrow$ 17.2%	$\downarrow$ 15.2%	$\downarrow$ 11.7%	$\downarrow$ 10.2%	$\downarrow$ 5.74%	$\downarrow$ 0.00%
100	Vanilla	Sample-wise	CR	7548	0	0	0	0	0	0	0	0	0
			M_{ATK}	2452	10000	10000	10000	10000	10000	10000	10000	10000	10000
			CA	7321	0	0	0	0	0	0	0	0	0
			M_{ATK}	1443	8764	8764	8764	8764	8764	8764	8764	8764	8764
		Probabilistic	CR	7169	6808	6518	6187	5805	5395	4876	3791	0	0
			CA	6958	6660	6405	6103	5746	5363	4855	3787	0	0
		Collective	CR	8053	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 20.6%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	7746	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 29.4%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	9538	9080	8653	8249	7823	7419	6928	6377	5611	147
			M_{ATK}	462	920	1347	1751	2177	2581	3072	3623	4389	9853
			CA	8554	8316	8049	7797	7486	7173	6759	6279	5568	147
			M_{ATK}	210	448	715	967	1278	1591	2005	2485	3196	8617
		Collective	CR	9611	9167	8754	8344	7912	7483	6980	6405	5631	147
			M_{ATK}	$\downarrow$ 15.8%	$\downarrow$ 9.46%	$\downarrow$ 7.50%	$\downarrow$ 5.42%	$\downarrow$ 4.09%	$\downarrow$ 2.48%	$\downarrow$ 1.69%	$\downarrow$ 0.77%	$\downarrow$ 0.46%	$\downarrow$ 0.00%
			CA	**8610**	8375	8116	7857	7558	7242	6830	6323	5628	147
			M_{ATK}	$\downarrow$ 26.7%	$\downarrow$ 13.2%	$\downarrow$ 9.37%	$\downarrow$ 6.20%	$\downarrow$ 5.63%	$\downarrow$ 4.34%	$\downarrow$ 3.54%	$\downarrow$ 1.77%	$\downarrow$ 1.88%	$\downarrow$ 0.00%
		Decomposition	CR	**9631**	**9232**	**8837**	**8450**	**8036**	**7617**	**7104**	**6513**	**5726**	**147**
			M_{ATK}	$\downarrow$ 20.1%	$\downarrow$ 16.5%	$\downarrow$ 13.6%	$\downarrow$ 11.5%	$\downarrow$ 9.78%	$\downarrow$ 7.67%	$\downarrow$ 5.73%	$\downarrow$ 3.75%	$\downarrow$ 2.62%	$\downarrow$ 0.00%
			CA	8595	**8407**	**8152**	**7917**	**7639**	**7334**	**6897**	**6404**	**5676**	**147**
			M_{ATK}	$\downarrow$ 19.5%	$\downarrow$ 20.3%	$\downarrow$ 14.4%	$\downarrow$ 12.4%	$\downarrow$ 12.0%	$\downarrow$ 10.1%	$\downarrow$ 6.88%	$\downarrow$ 5.03%	$\downarrow$ 3.38%	$\downarrow$ 0.00%

Table 12. (CIFAR-10: $M = 10,000$; $K = N/G$) Certified collective robustness and certified accuracy.

G	Bagging	Certification	Metric	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
50	Vanilla	Sample-wise	CR	2737	0	0	0	0	0	0	0	0	0
			M_{ATK}	7263	10000	10000	10000	10000	10000	10000	10000	10000	10000
			CA	2621	0	0	0	0	0	0	0	0	0
			M_{ATK}	4375	6996	6996	6996	6996	6996	6996	6996	6996	6996
		Probabilistic	CR	1820	1529	876	490	0	0	0	0	0	0
			CA	1781	1501	867	488	0	0	0	0	0	0
		Collective	CR	3621	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 12.2%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	3335	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 16.3%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	8221	7268	6067	5320	4229	3573	2635	2019	978	39
			M_{ATK}	1779	2732	3933	4680	5771	6427	7365	7981	9022	9961
			CA	6305	5864	5186	4705	3884	3339	2520	1961	962	39
			M_{ATK}	691	1132	1810	2291	3112	3657	4476	5035	6034	6957
		Collective	CR	8393	7428	6204	5435	4290	3624	2664	2043	1034	40
			M_{ATK}	$\downarrow$ 9.67%	$\downarrow$ 5.86%	$\downarrow$ 3.48%	$\downarrow$ 2.46%	$\downarrow$ 1.06%	$\downarrow$ 0.79%	$\downarrow$ 0.39%	$\downarrow$ 0.30%	$\downarrow$ 0.62%	$\downarrow$ 0.01%
			CA	6410	5985	5342	4848	4006	3434	2582	2007	1037	39
			M_{ATK}	$\downarrow$ 15.2%	$\downarrow$ 10.7%	$\downarrow$ 8.62%	$\downarrow$ 6.24%	$\downarrow$ 3.92%	$\downarrow$ 2.60%	$\downarrow$ 1.38%	$\downarrow$ 0.91%	$\downarrow$ 1.24%	$\downarrow$ 0.00%
		Decomposition	CR	**8694**	**7854**	**6686**	**5912**	**4826**	**4067**	**2995**	**2277**	**996**	**39**
			M_{ATK}	$\downarrow$ 26.6%	$\downarrow$ 21.4%	$\downarrow$ 15.7%	$\downarrow$ 12.6%	$\downarrow$ 10.3%	$\downarrow$ 7.69%	$\downarrow$ 4.89%	$\downarrow$ 3.23%	$\downarrow$ 0.20%	$\downarrow$ 0.00%
			CA	**6490**	**6147**	**5553**	**5113**	**4341**	**3733**	**2841**	**2234**	**1016**	**39**
			M_{ATK}	$\downarrow$ 26.8%	$\downarrow$ 25.0%	$\downarrow$ 20.2%	$\downarrow$ 17.8%	$\downarrow$ 14.7%	$\downarrow$ 10.8%	$\downarrow$ 7.17%	$\downarrow$ 5.42%	$\downarrow$ 0.90%	$\downarrow$ 0.00%
100	Vanilla	Sample-wise	CR	2621	0	0	0	0	0	0	0	0	0
			M_{ATK}	7379	10000	10000	10000	10000	10000	10000	10000	10000	10000
			CA	1876	0	0	0	0	0	0	0	0	0
			M_{ATK}	4378	6254	6254	6254	6254	6254	6254	6254	6254	6254
		Probabilistic	CR	1473	1092	815	581	368	236	128	29	0	0
			CA	1395	1050	794	567	364	233	127	29	0	0
		Collective	CR	2657	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 7.93%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			CA	2394	0	0	0	0	0	0	0	0	0
			M_{ATK}	$\downarrow$ 11.8%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Hash	Sample-wise	CR	7685	5962	4612	3504	2593	1833	1217	658	222	1
			M_{ATK}	2315	4038	5388	6496	7407	8167	8783	9342	9778	9999
			CA	5396	4571	3787	3008	2315	1694	1166	634	218	1
			M_{ATK}	858	1683	2467	3246	3939	4560	5088	5620	6036	6253
		Collective	CR	7744	5974	4618	3509	2598	1838	1221	660	224	1
			M_{ATK}	$\downarrow$ 2.54%	$\downarrow$ 0.30%	$\downarrow$ 0.11%	$\downarrow$ 0.08%	$\downarrow$ 0.07%	$\downarrow$ 0.06%	$\downarrow$ 0.05%	$\downarrow$ 0.02%	$\downarrow$ 0.02%	$\downarrow$ 0.00%
			CA	5475	4650	3825	3030	2330	1710	1174	638	224	1
			M_{ATK}	$\downarrow$ 9.21%	$\downarrow$ 4.69%	$\downarrow$ 1.54%	$\downarrow$ 0.68%	$\downarrow$ 0.38%	$\downarrow$ 0.35%	$\downarrow$ 0.16%	$\downarrow$ 0.07%	$\downarrow$ 0.10%	$\downarrow$ 0.00%
		Decomposition	CR	**8137**	**6469**	**5061**	**4035**	**2987**	**2032**	**1341**	**691**	**222**	**1**
			M_{ATK}	$\downarrow$ 19.5%	$\downarrow$ 12.5%	$\downarrow$ 8.33%	$\downarrow$ 8.17%	$\downarrow$ 5.32%	$\downarrow$ 2.44%	$\downarrow$ 1.41%	$\downarrow$ 0.35%	$\downarrow$ 0.00%	$\downarrow$ 0.00%
			CA	**5570**	**4841**	**4098**	**3338**	**2635**	**1928**	**1273**	**704**	**218**	**1**
			M_{ATK}	$\downarrow$ 20.3%	$\downarrow$ 16.0%	$\downarrow$ 12.6%	$\downarrow$ 10.2%	$\downarrow$ 8.12%	$\downarrow$ 5.13%	$\downarrow$ 2.10%	$\downarrow$ 1.25%	$\downarrow$ 0.00%	$\downarrow$ 0.00%