Provable Robustness against a Union of ℓ_0 Adversarial Attacks

Zayd Hammoudeh[1,2*], **Daniel Lowd**[1]

[1]University of Oregon
[2]Qualtrics, Inc.
{zayd, lowd}@cs.uoregon.edu

Abstract

Sparse or ℓ_0 adversarial attacks arbitrarily perturb an unknown subset of the features. ℓ_0 robustness analysis is particularly well-suited for heterogeneous (tabular) data where features have different types or scales. State-of-the-art ℓ_0 certified defenses are based on randomized smoothing and apply to evasion attacks only. This paper proposes *feature partition aggregation* (FPA) – a certified defense against the union of ℓ_0 evasion, backdoor, and poisoning attacks. FPA generates its stronger robustness guarantees via an ensemble whose submodels are trained on disjoint feature sets. Compared to state-of-the-art ℓ_0 defenses, FPA is up to $3{,}000\times$ faster and provides larger median robustness guarantees (e.g., median certificates of 13 pixels over 10 for CIFAR10, 12 pixels over 10 for MNIST, 4 features over 1 for Weather, and 3 features over 1 for Ames), meaning FPA provides the additional dimensions of robustness essentially for free.

1 Introduction

Machine learning models are vulnerable to numerous types of adversarial attacks, including (1) *evasion attacks* which manipulate a model by perturbing test instances (Szegedy et al. 2014), (2) *poisoning attacks* which manipulate predictions by perturbing a model's training set (Biggio, Nelson, and Laskov 2012), (3) *backdoor attacks* which combine training and test perturbations (Li et al. 2022), and (4) *patch attacks* – a specialized evasion attack where the adversarial perturbation is restricted to a specific shape (Brown et al. 2017). *Certified defenses* provide provable guarantees of a prediction's robustness against adversarial attack.

This work focuses on ℓ_0 or *sparse* attacks, where an adversary controls an unknown subset of the features. By certifying robustness w.r.t. the number of perturbed features, ℓ_0 analysis is particularly well-suited to heterogeneous (tabular) data where the features have different types (e.g., numerical, categorical) or scales. Moreover, ℓ_0 defenses provide provable robustness against real-world patch attacks (Levine and Feizi 2020a). Several certified ℓ_0 defenses have been proposed (Lee et al. 2019; Levine and Feizi 2020b; Calzavara et al. 2021; Jia et al. 2022b; Levine and Feizi 2022), but these methods apply to evasion only, which can be limiting. For example, consider a distributed sensor network where each (tabular) feature is independently measured by a different sensor. Under this type of *vertical partitioning* where features are sourced from multiple parties, an attacker that controls a single feature (i.e., sensor) can partially perturb every instance – training and test – up to 100% poisoning rate (Li, Dowsley, and De Cock 2021; Wei et al. 2022). Existing ℓ_0 evasion defenses do not certify robustness over any training perturbation rendering them moot under such an attack. Moreover, existing ℓ_0 defenses could not be combined with instance-wise poisoning defenses here since, typically, the latter are only provably robust under small poisoning rates, e.g., $\leq 1\%$ (Rezaei et al. 2023).

To address these limitations, we propose *feature partition aggregation* (FPA) – a certified sparse defense jointly robust against both training and test feature perturbations. FPA uses a model ensemble approach, where each submodel is trained on a disjoint feature set, meaning any adversarially perturbed feature – training or test – affects at most one submodel prediction. Hence, FPA guarantees robustness over the *union* of ℓ_0 evasion, backdoor, and poisoning attacks – a strictly stronger guarantee than existing ℓ_0 methods (Levine and Feizi 2020b). In our empirical evaluation, FPA's certified median guarantees are up to $4\times$ larger than state-of-the-art ℓ_0 defenses (Jia et al. 2022b) with little to no decrease in classification accuracy; FPA is also up to $3{,}000\times$ faster. In other words, FPA provided additional dimensions of ℓ_0 robustness essentially for free. Our primary contributions are summarized below; additional theoretical analysis and all proofs are in the supplement.

- We define a new robustness paradigm we term *certified feature robustness* that generalizes ℓ_0 (sparse) robustness to encompass training set feature perturbations.
- We propose feature partition aggregation, a certified feature defense that uses an ensemble of submodels trained on disjoint feature sets. We detail two certification schemes – a simple one based on plurality voting and the other based on multi-round elections.
- We empirically evaluate FPA on two classification and two regression datasets. FPA provided simultaneously larger and stronger median guarantees than the state-of-the-art certified ℓ_0 defenses while also being 2 to 3 orders of magnitude faster.

*Work primarily done while at the University of Oregon.

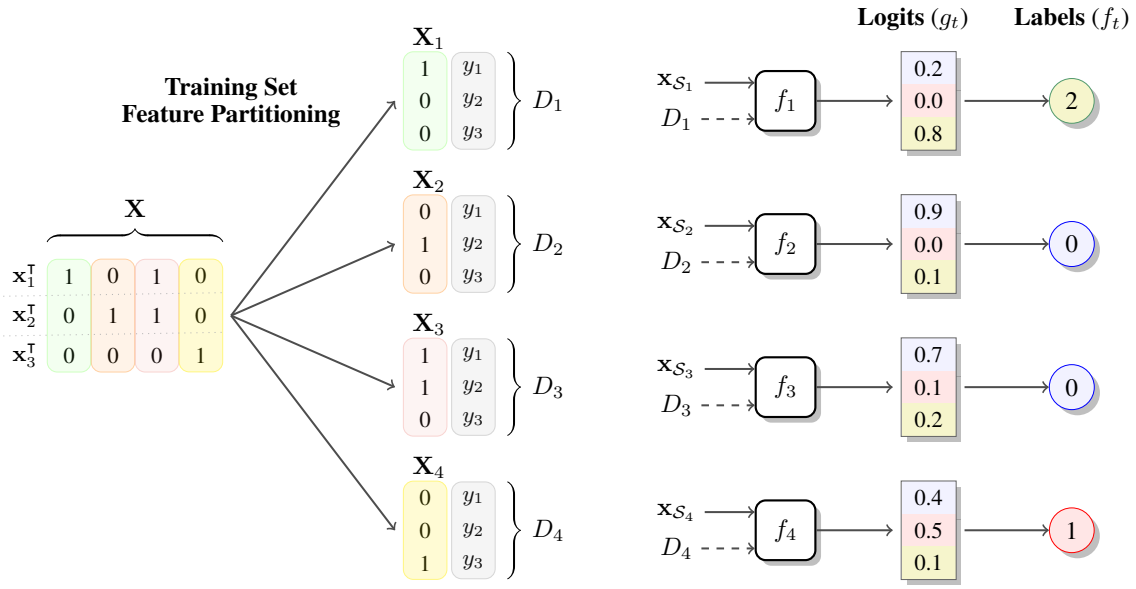

Figure 1: Feature partition aggregation example prediction for: test instance $\mathbf{x} \in \mathcal{X}$, $n = 3$, $d = 4$, and $|\mathcal{Y}| = 3$. Feature partitioning across $T = 4$ submodels, where the t-th submodel uses only feature dimensions $\mathcal{S}_t = \{t\} \subset [4]$ and training set D_t, i.e., the tuple containing the t-th column of feature matrix $\mathbf{X}$ (denoted $\mathbf{X}_t$) and label vector $\mathbf{y} \coloneqq [y_1, y_2, y_3]$. $\mathbf{x}_{\mathcal{S}_t}$ denotes the subvector of $\mathbf{x}$ restricted to the feature dimensions in $\mathcal{S}_t$. Plurality label $y_{\text{pl}} = 0$; runner-up label $y_{\text{ru}} = 1$; and the predicted label with the run-off decision function is $y_{\text{RO}} = 0$. Under the plurality voting decision function (Sec. 4.1), $f(\mathbf{x})$ has certified feature robustness $r_{\text{pl}} = 0$. With the run-off decision function (Sec. 4.2), $f(\mathbf{x})$'s certified feature robustness is $r_{\text{RO}} = 1$.

2 Preliminaries

Notation Supplemental Section A provides a full nomenclature reference. Let $[m]$ denote integer set $\{1, \ldots, m\}$. $\mathbb{1}[a]$ is the *indicator function*, which equals 1 if predicate a is true and 0 otherwise. ℓ_0 *norm* $\|\mathbf{w}\|_0$ is the number of non-zero elements in vector $\mathbf{w}$. Given some matrix $\mathbf{A}$, denote its j-th column as $\mathbf{A}_j$. In a slight abuse of notation, let $\mathbf{A} \ominus \mathbf{A}' \coloneqq \{j : \mathbf{A}_j \neq \mathbf{A}'_j\}$ denote the set of column *indices* over which equal-size matrices $\mathbf{A}$ and $\mathbf{A}'$ differ. Similarly, let $\mathbf{v} \ominus \mathbf{v}' \subseteq [|\mathbf{v}|]$ denote the set of *dimensions* where vectors $\mathbf{v}$ and $\mathbf{v}'$ differ.

Let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ be a *feature vector* ($d \coloneqq |\mathbf{x}|$) and $y \in \mathcal{Y} \subseteq \mathbb{N}$ a *label*. A *training set* $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ consists of n instances. Denote the training set's *feature matrix* as $\mathbf{X} \coloneqq [\, \mathbf{x}_1 \cdots \mathbf{x}_n \,]^\intercal$ where $\mathbf{X} \in \mathbb{R}^{n \times d}$, and denote the label vector $\mathbf{y} \coloneqq [y_1, \ldots, y_n]$. Let $f : \mathcal{X} \to \mathcal{Y}$ be a *model*.

For our method, feature partition aggregation (FPA), f is an ensemble of T *submodels* (see Figure 1). A *decision function* aggregates the T submodel predictions to form f's overall prediction. The model architecture and decision function combined dictate how a prediction's *certified robustness* is calculated. For instance $(\mathbf{x}, y)$, let $g_t(\mathbf{x}, y)$ be the t-th submodel's *logit* value for label y, where $g_t : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Let $f_t(\mathbf{x})$ denote the t-th submodel's predicted *label* for $\mathbf{x}$, where $f_t : \mathcal{X} \to \mathcal{Y}$ and $f_t(\mathbf{x}) \coloneqq \arg\max_{y \in \mathcal{Y}} g_t(\mathbf{x}, y)$. Throughout this work, all ties are broken by selecting the label with the smallest index.

Feature set $[d]$ is *partitioned* across FPA's T submodels. Let $\mathcal{S}_t \subset [d]$ be the features used by the t-th submodel where $\bigsqcup_{t=1}^T \mathcal{S}_t = [d]$. In other words, each FPA submodel considers a fixed, disjoint subset of the features for all training and test instances. The t-th submodel's training set, D_t, consists of: label vector $\mathbf{y}$ and the $\mathcal{S}_t$ columns in $\mathbf{X}$. FPA submodels are *deterministic*, meaning fixing D_t, $\mathcal{S}_t$, and $\mathbf{x}$, in turn, fixes label $f_t(\mathbf{x})$ and logits $\forall_y\, g_t(\mathbf{x}, y)$.

Given $\mathbf{x}$ and y, the pointwise *submodel vote count* is $\dot{c}_y(\mathbf{x}) \coloneqq \sum_{t=1}^T \mathbb{1}[f_t(\mathbf{x}) = y]$. The *plurality* and *runner-up* labels receive the most and second-most votes (resp.), i.e., $y_{\text{pl}} = \arg\max_{y \in \mathcal{Y}} \dot{c}_y(\mathbf{x})$ and $y_{\text{ru}} = \arg\max_{y \in \mathcal{Y} \setminus y_{\text{pl}}} \dot{c}_y(\mathbf{x})$. The pointwise *submodel vote gap* between labels $y, y' \in \mathcal{Y}$ is

$$\text{GAP}_{\text{vote}}(y, y'; \mathbf{x}) \coloneqq \dot{c}_y(\mathbf{x}) - \dot{c}_{y'}(\mathbf{x}) - \mathbb{1}\big[y' < y\big], \qquad (1)$$

with the indicator function used to break ties. Let $\ddot{c}_y(\mathbf{x}; y') \coloneqq \sum_{t=1}^T \mathbb{1}[g_t(\mathbf{x}, y) > g_t(\mathbf{x}, y')]$ be y's *logit vote count* w.r.t. $y' \in \mathcal{Y}$. The pointwise *logit vote gap* for y w.r.t. y' is

$$\text{GAP}_{\text{logit}}(y, y'; \mathbf{x}) \coloneqq \ddot{c}_y(\mathbf{x}; y') - \ddot{c}_{y'}(\mathbf{x}; y) - \mathbb{1}\big[y' < y\big]. \qquad (2)$$

Below, $\mathbf{x}$ is dropped from GAP_{vote} and $\text{GAP}_{\text{logit}}$ when the feature vector of interest is clear from context.

Threat Model Given arbitrary $(\mathbf{x}, y)$, the attacker's objective is for $y \neq f(\mathbf{x})$. The adversary achieves this objective via two methods: (1) modify training features $\mathbf{X}$ or (2) modify test instance $\mathbf{x}$'s features.[1] An adversary may use either method individually or both methods jointly. An attacker can *perturb up to 100% of the training instances*.

[1] Our primary threat model assumes a *clean-label attacker* that does not modify training labels. Suppl. Sec. E provides additional theoretical results for an adversary that modifies training labels.

Our Objective Given $(\mathbf{x}, y)$, determine the *certified feature robustness*, r (defined below). *Pointwise* guarantees certify the robustness of each instance $(\mathbf{x}, y)$ individually.

Def. 1. Certified Feature Robustness *Given training set* $(\mathbf{X}, \mathbf{y})$*, model* f' *trained on* $(\mathbf{X}', \mathbf{y})$*, and arbitrary feature vector* $\mathbf{x}' \in \mathcal{X}$*, certified feature robustness* $r \in \mathbb{N}$ *is a pointwise, deterministic guarantee w.r.t. instance* $(\mathbf{x}, y)$ *where* $|\mathbf{X} \ominus \mathbf{X}' \cup \mathbf{x} \ominus \mathbf{x}'| \leq r \implies y = f'(\mathbf{x}')$.

Certified robustness r is not w.r.t. individual feature values. Rather, certified feature robustness provides a stronger guarantee allowing all values of a feature – training and test – to be perturbed.

3 Related Work

FPA marries ideas from two classes of certified adversarial defenses, which are discussed below. A more detailed discussion of related work is deferred to suppl. Section C.

ℓ_0-Norm Certified Evasion Defenses Representing the work most closely related to ours, these methods certify *ℓ_0-norm robustness* (also known as "*sparse robustness*"), which we formalize below.

Def. 2. ℓ_0-Norm Certified Robustness *Given model* f*,* $\alpha \in (0, 1)$*, and arbitrary feature vector* $\mathbf{x}' \in \mathcal{X}$*,* ℓ_0*-norm certified robustness* $\rho \in \mathbb{N}$ *is a pointwise guarantee w.r.t. instance* $(\mathbf{x}, y)$ *where if* $\|\mathbf{x} - \mathbf{x}'\|_0 \leq \rho$*, then* $y = f(\mathbf{x}')$ *with probability at least* $1 - \alpha$*.*

There are two main differences between certified ℓ_0-norm robustness (Def. 2) and our certified feature robustness (Def. 1). (1) ℓ_0-norm methods are not certifiably robust against any adversarial training perturbations (e.g., poisoning and backdoors). (2) ℓ_0-norm robustness guarantees are *probabilistic*, while our feature guarantees are deterministic. Put simply, our certified feature guarantees are *strictly stronger* than ℓ_0-norm guarantees.

Randomized ablation (RA) is the state-of-the-art certified ℓ_0-norm defense (Levine and Feizi 2020b; Jia et al. 2022b). RA adapts ideas from *randomized smoothing* (Cohen, Rosenfeld, and Kolter 2019) to ℓ_0 evasion attacks. Specifically, RA creates a single *smoothed classifier* by repeatedly evaluating different *ablated inputs*, each of which *keeps* a random subset of the features unchanged and masks out (*ablates*) all other features. RA's *ablated training* generally permits only stochastically trained, parametric model architectures. At inference, certifying a single prediction with RA requires evaluating up to 100k ablated inputs (Jia et al. 2022b). Jia et al. (2022b) improve RA's guarantees via novel certification analysis that is tight for top-1 predictions, meaning Jia et al.'s version of RA always performs at least as well as the original. Jia et al. (2022b) also extend RA to certify ℓ_0-norm robustness for top-k predictions.

Certified patch robustness is a restricted form of ℓ_0-norm robustness where the perturbed test features are constrained to a specific, contiguous shape, e.g., square (Metzen and Yatsura 2021). Existing patch defenses include (de)randomized smoothing (DRS) (Levine and Feizi 2020a) – a specialized version of randomized ablation for patch attacks. Like RA, DRS performs ablated training and inference. By assuming a single patch shape, the number of possible attacks becomes linear in d, allowing DRS to only evaluate $\mathcal{O}(d)$ ablations during inference; this derandomizes the ablation set, making DRS's patch guarantees deterministic.[2] More recently, Metzen and Yatsura (2021) propose BAGCERT – a certified patch defense that is less sensitive to patch shape than DRS. Note any certified feature or ℓ_0-norm defense (e.g., FPA, RA) is also a certified patch defense, given the former's stronger guarantees.

Instance-wise Certified Poisoning Defenses The second class of related defenses certify robustness under the arbitrary insertion or deletion of entire *instances* in the training set (Chen et al. 2022) – generally a small poisoning rate (e.g., $\leq 1\%$). Like FPA, most instance-wise poisoning defenses are voting-based (Jia, Cao, and Gong 2021; Jia et al. 2022a; Wang, Levine, and Feizi 2022a). For example, *deep partition aggregation* (DPA) randomly partitions the training *instances* across an ensemble of T submodels (Levine and Feizi 2021). More recently, Rezaei et al. (2023) propose *run-off elections*, a novel decision function for DPA that can improve DPA's certified robustness by several percentage points. While certified instance-wise poisoning defenses show promise, they are still vulnerable to test perturbations – even of a single feature.

4 Certifying Feature Robustness

Our certified defense, feature partition aggregation (FPA), can be viewed as the *transpose* of Levine and Feizi's (2021) deep partition aggregation (DPA). Both defenses are (1) ensembles, (2) rely on voting-based decision functions, and (3) partition the training set; the key difference is in the partitioning operation. DPA horizontally partitions the set of training *instances* (rows of feature matrix $\mathbf{X}$), enabling DPA to certify *instance-wise* robustness. In contrast, FPA vertically partitions along an orthogonal dimension – the feature set (columns of $\mathbf{X}$) – enabling FPA to certify *feature-wise* robustness. Intuitively, *partitioning along orthogonal dimensions means that DPA and FPA certify orthogonal types of robustness*. Training FPA submodels on disjoint feature subsets (e.g., Figure 1) entails that a perturbed feature affects, at most, one submodel prediction. FPA leverages this property to certify feature robustness r. Below we describe two FPA *decision functions*: (1) a simpler scheme using plurality voting and (2) an enhanced multi-round voting procedure specialized for multiclass classification. The decision function combined with FPA's architecture dictates how our robustness guarantee is calculated.

4.1 Feature Robustness Under Plurality Voting

For $\mathbf{x} \in \mathcal{X}$, the *plurality voting* decision function defines the model prediction as $f(\mathbf{x}) \coloneqq y_{\text{pl}}$, i.e., the label that receives the most submodel votes. A successful attack requires perturbing enough submodels to change y_{pl}. Specifically, each submodel perturbation decreases the submodel vote gap (GAP_{vote}) between y_{pl} and the adversary's selected

[2] (De)randomized smoothing's deterministic guarantees do not scale to RA which considers $\mathcal{O}(2^d)$ possible attacks.

label by two. Hence, the minimum number of submodel perturbations equals half the vote gap between y_{pl} and runner-up label y_{ru}. Theorem 3 formalizes this idea as a deterministic feature robustness guarantee. Eq. (3)'s decomposed form is similar to other voting-based certified defenses, including DPA (Levine and Feizi 2021; Jia et al. 2022a).

Theorem 3. Certified Feature Robustness with Plurality Voting *For feature partition $\mathcal{S}_1, \ldots, \mathcal{S}_T$, let f be an ensemble of T submodels using the plurality-voting decision function, where the t-th submodel uses the features in $\mathcal{S}_t$. For instance $(\mathbf{x}, y)$, the pointwise certified feature robustness is*

$$r_{\text{pl}} \coloneqq \left\lfloor \frac{\text{GAP}_{\text{vote}}(y_{\text{pl}}, y_{\text{ru}})}{2} \right\rfloor. \tag{3}$$

Understanding Theorem 3 More Intuitively Let $\mathcal{A}_{\text{tr}} \subseteq [d]$ be the set of features (i.e., dimensions) an attacker modified in the training set, and let $\mathcal{A}_{\mathbf{x}} \subseteq [d]$ be the set of features the attacker modified in instance $\mathbf{x}$. As long as $|\mathcal{A}_{\text{tr}} \cup \mathcal{A}_{\mathbf{x}}| \leq r$, the adversarial perturbations did not change the model prediction. The union over the perturbed feature sets entails that a feature perturbed in both training and test counts only once against guarantee r. Theorem 3's certified guarantees are implicitly agnostic to the ℓ_0 attack type. Certified feature robustness r applies equally to an ℓ_0 evasion attack ($\mathcal{A}_{\mathbf{x}}$ only) as it does to ℓ_0 poisoning ($\mathcal{A}_{\text{tr}}$ only). Theorem 3's guarantees also encompass more complex ℓ_0 backdoor attacks ($\mathcal{A}_{\text{tr}} \cup \mathcal{A}_{\mathbf{x}}$).

Against a *worst-case adversary* where a feature perturbation arbitrarily changes the corresponding submodel's prediction, Eq. (3)'s guarantee is tight under plurality-voting.

Top-k Certified Feature Robustness In top-k predictions, a classifier predicts k labels for each instance $\mathbf{x}$, with the accuracy calculated based on whether $\mathbf{x}$'s true label is among the k predicted labels. In line with Jia et al.'s (2022b) extension of RA to top-k predictions, supplemental Section D extends FPA with plurality voting to certify top-k feature robustness.

4.2 Feature Robustness Under Run-Off Elections

Under plurality voting, only submodels that predict either y_{pl} or y_{ru} are considered when determining the certified feature robustness (Eq. (3)). In other words, submodels predicting other labels essentially contribute nothing to plurality voting's pointwise guarantees. Decision functions that leverage these "wasted" submodels may certify larger guarantees (see Figure 1). For instance, Rezaei et al. (2023) propose *run-off elections*, an enhanced two-round DPA decision function for multiclass classification.[3] Since FPA and DPA share the same basic architecture (excluding the partitioning dimension), run-off can be directly combined with FPA to improve our certified robustness.

We now describe run-off. Our presentation is mostly similar to Rezaei et al.'s (2023) beyond standardizing the formulation to align with previous work. Formally, run-off's decision function procedure is:

Round #1: Determine plurality and runner-up labels y_{pl} and y_{ru} (resp.) as above.

Round #2: Set *run-off prediction* y_{RO} to either label y_{pl} or y_{ru} based on the logit vote gap where

$$f(\mathbf{x}) = y_{\text{RO}} \coloneqq \begin{cases} y_{\text{pl}} & \text{GAP}_{\text{logit}}(y_{\text{pl}}, y_{\text{ru}}) \geq 0 \\ y_{\text{ru}} & \text{Otherwise} \end{cases}. \tag{4}$$

Under run-off, ensemble prediction y_{RO} can only be perturbed in two ways: (1) overtake y_{RO} in round #2 or (2) eject y_{RO} from round #1's top-two labels, preventing y_{RO} reaching round #2. Run-off's robustness is lower bounded by whichever of these two cases takes fewer submodel perturbations. Each case is analyzed separately below; Theorem 4 then combines the analyses to form run-off's overall robustness r_{RO}.

Case #1: Overtake y_{RO} in Round #2 Let $\widetilde{y}_{\text{RO}} \coloneqq \{y_{\text{pl}}, y_{\text{ru}}\} \setminus y_{\text{RO}}$ denote the label not selected in round #2. For a label y to overtake y_{RO} in round #2, y must simultaneously satisfy two requirements: (a) be in round #1's top-two labels (in turn ejecting $\widetilde{y}_{\text{RO}}$ from the top two) and (b) receive more logit votes than y_{RO} in round #2. Hence, the certified robustness for this case is bounded by whichever of these requirements requires more feature perturbations. Therefore, an attacker may control up to

$$r_{\text{RO}}^{\text{Case1}} \coloneqq \min_{y \in \mathcal{Y} \setminus y_{\text{RO}}} \max \left\{ \left\lfloor \frac{\text{GAP}_{\text{vote}}(\widetilde{y}_{\text{RO}}, y)}{2} \right\rfloor, \left\lfloor \frac{\text{GAP}_{\text{logit}}(y_{\text{RO}}, y)}{2} \right\rfloor \right\} \tag{5}$$

features, without y_{RO} being overtaken in round #2 (see Lemma 6 in the supplement).

Case #2: Eject y_{RO} from Round #1's Top-Two Labels In round #1, run-off prediction y_{RO} is preferred over label y iff $\text{GAP}_{\text{vote}}(y_{\text{RO}}, y) \geq 0$ (Lemma 5 in the supplement). For y_{RO} to qualify for run-off's second round, then for any pair of labels $y, y' \in \mathcal{Y} \setminus y_{\text{RO}}$ either $\text{GAP}_{\text{vote}}(y_{\text{RO}}, y) \geq 0$ or $\text{GAP}_{\text{vote}}(y_{\text{RO}}, y') \geq 0$. Calculating case #2's certified robustness reduces to determining the maximum number of submodels that can be perturbed with the above property still holding for all $\binom{|\mathcal{Y}|}{2}$ label pairs.

Recall that perturbing a submodel's vote from y_{RO} to y decreases $\text{GAP}_{\text{vote}}(y_{\text{RO}}, y)$ by 2; this submodel perturbation also decreases $\text{GAP}_{\text{vote}}(y_{\text{RO}}, y')$ by $1\ \forall\, y' \in \mathcal{Y} \setminus \{y_{\text{RO}}, y\}$. We leverage this simple insight to determine case #2's certified robustness. Formally, let $\texttt{dp}$ be a function that takes two submodel vote gaps (e.g., $\Delta, \Delta' \in \mathbb{N}$) and returns a lower bound on the number of possible submodel perturbations where either $\text{GAP}_{\text{vote}}(y_{\text{RO}}, y) \geq 0$ or $\text{GAP}_{\text{vote}}(y_{\text{RO}}, y') \geq 0$. Applying the insight above, Lemma 7 in the supplement shows that

$$\texttt{dp}\big[\Delta, \Delta'\big] = 1 + \min\{\texttt{dp}\big[\Delta - 2, \Delta' - 1\big], \texttt{dp}\big[\Delta - 1, \Delta' - 2\big]\}. \tag{6}$$

Eq. (6)'s base case sets $\texttt{dp}[\Delta, \Delta'] = 0$ when $\max\{\Delta, \Delta'\} \leq 1$ and $(\Delta, \Delta') \neq (1, 1)$; this ensures the vote gap non-negativity condition always holds for at least one of the two labels of interest.[4]

[3] Run-off only changes the decision function; no training or model architecture changes are required.

[4] This base case differs slightly from Rezaei et al.'s (2023) run-off formulation to ensure our robustness guarantee is tight against a worst-case adversary.

A worst-case adversary attacks whichever label pair, (y, y'), requires the fewest perturbations, making case #2's overall robustness

$$r_{\text{RO}}^{\text{Case2}} \coloneqq \min_{y,y' \in \mathcal{Y} \setminus y_{\text{RO}}} \texttt{dp}\big[\text{GAP}_{\text{vote}}(y_{\text{RO}}, y), \text{GAP}_{\text{vote}}(y_{\text{RO}}, y')\big]. \quad (7)$$

Eq. (6)'s recursive formulation is solvable using classic dynamic programming. $\mathcal{O}(T^2)$-space matrix $\texttt{dp}$ is prepopulated once for all $\mathbf{x}$, making $r_{\text{RO}}^{\text{Case2}}$'s amortized time complexity $\mathcal{O}(|\mathcal{Y}|^2)$.

Combining Cases #1 and #2 to Certify Feature Robustness Theorem 4 provides the certified feature robustness for an FPA prediction using the run-off decision function. Intuitively, an optimal attacker selects whichever of the two cases above requires fewer feature perturbations; hence, Eq. (8) below takes the minimum of $r_{\text{RO}}^{\text{Case1}}$ and $r_{\text{RO}}^{\text{Case2}}$.

Theorem 4. Certified Feature Robustness with Run-off *For feature partition $\mathcal{S}_1, \ldots, \mathcal{S}_T$, let f be an ensemble of T submodels using the run-off decision function, where the t-th submodel uses only the features in $\mathcal{S}_t$. Then, for instance $(\mathbf{x}, y)$, the pointwise certified feature robustness is*

$$r_{\text{RO}} = \min\{r_{\text{RO}}^{\text{Case1}}, r_{\text{RO}}^{\text{Case2}}\}. \quad (8)$$

4.3 Advantages of Feature Partition Aggregation

Below, we summarize FPA's advantages over state-of-the-art certified ℓ_0-norm defense randomized ablation (RA). These advantages apply irrespective of whether FPA uses plurality voting or run-off.

(1) **Stronger Guarantees** FPA's certified feature robustness guarantee (Def. 1) is strictly stronger than RA's ℓ_0-norm guarantee (Def. 2). First, FPA's guarantees apply equally to ℓ_0 evasion, poisoning, and backdoor attacks while RA only applies to evasion. Second, FPA's guarantees are deterministic while RA's guarantees are only probabilistic.

(2) **Faster** RA requires up to 100k forward passes to certify one prediction. FPA requires only T forward passes – one for each submodel – where $T < 200$ in general. FPA certification is, therefore, orders of magnitude faster than RA.

(3) **Model Architecture Agnostic** RA's feature ablation is specialized for parametric models like neural networks and generally prevents the use of tree-based models like gradient-boosted decision trees (GBDTs). By contrast, FPA supports any submodel architecture.

5 Feature Partitioning Strategies

The certification analysis above holds irrespective of the feature partitioning strategy. However, how the features are partitioned can have a *major* impact on the size of FPA's certified guarantees. Below, we very briefly describe two insights into the properties of good feature partitions.

Insight #1 *Ensure sufficient feature information is available to each submodel*. Each incorrect submodel or logit vote cancels out a correct vote, meaning the goal should be to maximize the number of correct submodel predictions while simultaneously minimizing incorrect ones. In other words, robustness is maximized when all submodels perform well, and feature information is divided equally.

Insight #2 *Limit information loss due to feature partitioning*. Models use (implicit) feature interaction information when making a prediction. Intuitively, if a pair of features is assigned to different FPA submodels, none of the submodels can use these features' pairwise interaction during inference. Put simply, feature partitioning causes some feature (interaction) information to be completely lost. Fixing T, some feature partitions are more lossy than others, and good partitions limit the total information lost.

5.1 Feature Partitioning Paradigms

Applying the above insights, we propose two general feature partitioning paradigms. In practice, the partitioning strategy is essentially a hyperparameter tunable on validation data. The validation set need not be clean so long as the perturbations are representative of the test distribution.

Balanced Random Partitioning Given no domain-specific knowledge, each feature's expected information content is equal. *Balanced random partitioning* assigns each submodel a disjoint feature subset sampled uniformly at random, with subsets differing in size by at most one. Random partitioning has two primary benefits. First, each submodel has the same a priori expected information content. Second, random partitioning can be applied to any dataset. FPA with random partitioning is usually a good initial strategy and empirically performs quite well.

Deterministic Partitioning One may have application-related insights into quality feature partitions. For example, consider feature partitioning of images. Features (i.e., pixels) in an image are ordered, and that structure can be leveraged to design better feature partitions. Often the most salient features are clustered in an image's center. To ensure all submodels are high-quality, each submodel should be assigned as many highly salient features as possible. Moreover, adjacent pixels can be highly correlated, i.e., contain mostly the same information. Given a fixed set of pixels to analyze, the information contained in those limited features should be maximized, so a good strategy can be to select a subset of pixels spread uniformly across the image. Put simply, for images, random partitioning can have larger information loss than deterministic strategies.

Supplemental Section H.7 empirically compares random and deterministic partitioning. In summary, a simple strided strategy that distributes features regularly across an image tends to work well for vision. Formally, given d pixels and T submodels, the t-th submodel's feature set under *strided partitioning* is $\mathcal{S}_t = \{j \in [d] : j \bmod T = t - 1\}$.

5.2 Beyond Partitioned Feature Subsets

Everything above should *not* be interpreted to imply that FPA necessarily requires partitioned feature sets. Submodel feature sets can (partially) overlap, but determining optimal r under overlapping feature sets is NP-hard in general. Supplemental Section F extends FPA to overlapping feature sets and provides an empirical comparison. In summary, overlapping submodel feature sets can marginally outper-

form random partitioning but often lags deterministic partitions.

6 Evaluation

Our empirical evaluation is modeled after Levine and Feizi's (2020b) evaluation of randomized ablation. Due to space, additional results are deferred to supplement Section H including: the base (non-robust) accuracy for each dataset (H.1), full numerical results (H.2 and H.3), hyperparameter sensitivity analysis (H.4 and H.5), plurality voting vs. run-off comparison (H.6), random vs. deterministic feature partitioning comparison (H.7), and model training times (H.8). Our source code is available at https://github.com/ZaydH/feature-partition.

6.1 Experimental Setup

Most evaluation setup details are deferred to supplemental Section G with a brief summary below. We evaluate FPA with both the plurality-voting (Section 4.1) and run-off (Section 4.2) decision functions.

Baselines Randomized ablation (RA) is FPA's most closely related work and the primary baseline below. We report the performance of both Levine and Feizi's (2020b) original version of RA (denoted "(LF'20b)") and Jia et al.'s (2022b) improved version (denoted "(Jia'22b)") where the certification analysis is tight for top-1 predictions. RA performs feature ablation during training and inference. Each ablated input keeps e randomly selected features unchanged and masks out the remaining $(d - e)$ features; RA evaluates up to 100k ablated inputs to certify each prediction. Recall that RA's ℓ_0-norm robustness only applies to evasion attacks (Def. 2), while FPA provides strictly stronger guarantees covering both training and test perturbations (Def. 1).

We also compare FPA to three certified patch defenses: *(de)randomized smoothing* (Levine and Feizi 2020a), *patch interval bound propagation* (IBP) (Chiang et al. 2020), and BAGCERT (Metzen and Yatsura 2021).

Performance Metrics Certified defenses generally trade-off robustness and (clean) accuracy. Hence, following Levine and Feizi's (2020b) evaluation of randomized ablation, performance is primarily measured using two complementary metrics: (1) *median certified robustness*, the median value of the certified robustness across a dataset's entire test set with misclassified instances assigned robustness $-\infty$ and (2) *classification accuracy*, the fraction of test predictions classified correctly. Below, r_{med} and ρ_{med} denote the median certified feature robustness (Def. 1) and median ℓ_0-norm robustness (Def. 2), respectively.

Mean certification time measures the time to certify a single prediction. *Certified accuracy* is the fraction of correctly-classified test instances that satisfy some specific robustness criterion; this criterion can be patch robustness or certified robustness of at least $\psi \in \mathbb{N}$.

Datasets We compare the methods on standard datasets used in data poisoning evaluation. First, following Levine and Feizi's (2020b) evaluation of baseline RA, we consider MNIST and CIFAR10[5] where each feature corresponds to one (RGB) pixel. Second, Hammoudeh and Lowd (2023) prove that certified regression *reduces* to certified *binary* classification when median is used as the regressor's decision function (see Section G.6 for details). We apply their reduction to both FPA and RA where for instance $(\mathbf{x}, y)$ and hyperparameter $\xi \in \mathbb{R}_{\geq 0}$, the goal is to certify that $y - \xi \leq f(\mathbf{x}) \leq y + \xi$. We consider two tabular regression datasets evaluated by Hammoudeh and Lowd (2023). (1) Weather (Malinin et al. 2021) predicts the temperature using features such as date, longitude, and latitude ($\xi = 3°\text{C}$). (2) Ames (De Cock 2011) predicts housing prices using features such as square footage ($\xi = 15\%y$). These two regression datasets serve as a stand-in for vertically partitioned data, which are commonly tabular and, as Section 1 mentions, particularly vulnerable to our union of ℓ_0 attacks threat model. Note run-off and plurality voting are identical under binary classification so we only report FPA's plurality voting regression results.

Model Architectures For vision datasets CIFAR10 and MNIST, all methods used convolutional neural networks. Baseline randomized ablation models were trained using Levine and Feizi's (2020a) published source code. Since FPA requires training multiple submodels, FPA uses small, simple CNNs that are fast to train. Specifically, the compact ResNet9 (Page 2020) architecture was used for CIFAR10, allowing submodels to be trained from scratch in as little as 60 seconds (Coleman et al. 2017). For MNIST, we follow Levine and Feizi's (2021) evaluation of instance-wise poisoning defense DPA and use the Network-in-Network architecture (NiN) (Lin, Chen, and Yan 2014).

Gradient-boosted decision trees (GBDTs) generally work exceptionally well on tabular data (Brophy, Hammoudeh, and Lowd 2023) so for regression datasets Weather and Ames, FPA used LightGBM GBDTs (Ke et al. 2017). In contrast, RA's feature ablation prevents the use of tree-based models like GBDTs, so RA instead used linear models for these two datasets (Hammoudeh and Lowd (2023) also used linear models for Weather). Even when restricted to linear submodels, FPA still had better median robustness and classification accuracy than RA; see suppl. Tables 24 and 25.

Feature Partitioning Strategy Supplemental Section H.7 shows that deterministic, strided feature partitioning significantly outperforms random partitioning on vision data. Specifically, deterministic partitioning improved FPA's certified accuracy by up to 15.6 percentage points (pp) for CIFAR10 and 11.9pp for MNIST. Hence, this section exclusively reports FPA's performance on these two datasets using strided partitioning, with each submodel considering the full image dimensions and any pixels not in $\mathcal{S}_t$ set to 0.

Section H.7 shows no consistent advantage when using deterministic partitioning for unordered tabular features. As such, for Weather and Ames, this section reports FPA's performance using balanced random partitioning.

Hyperparameters Hyperparameters T (FPA's submodel count) and e (RA's kept feature count) control the corre-

[5]Existing certified poisoning defenses do not evaluate on full ImageNet due to the high training cost (Weber et al. 2020; Levine and Feizi 2021; Jia et al. 2022a; Wang, Levine, and Feizi 2022a,b; Rezaei et al. 2023).

Dataset	Dim. (d)	FPA (ours)		Random. Ablate.	
		Plural	Run-Off	(LF'20b)	(Jia'22b)
CIFAR10	1024	11	**13**	7	10
MNIST	784	9	**12**	8	10
Weather	128	**4**	–	0	1
Ames	352	**3**	–	1	1

Table 1: Median certified robustness (larger is better). Each dataset's best performing method is in bold. Our median robustness was 20–30% larger for classification and 3 to 4× larger for regression while simultaneously providing stronger guarantees. For detailed results, see Section H.2.

Dataset	FPA (ours)				RA (Jia'22b)	
	r_{med}	Acc.	r_{med}	Acc.	ρ_{med}	Acc.
CIFAR10	13	62.4	10	**75.0**	10	64.7
MNIST	12	87.2	10	**96.1**	10	93.1
Weather	4	76.1	1	**85.3**	1	75.2
Ames	3	65.5	1	**84.6**	1	67.2

Table 2: Classification accuracy (% – larger is better). We report FPA's accuracy at both RA's (middle, bold) and FPA's (left) best median robustness levels. At RA's best median robustness, FPA had better classification accuracy for all four datasets. For full results, see Section H.2.

Dataset	RA (Jia'22b)		FPA (ours)		Speedup
	e	Time	T	Time	
CIFAR10	15	5.4E+0	115	7.3E−3	**743**×
MNIST	25	6.8E−1	60	2.9E−3	**235**×
Weather	45	3.1E−1	21	1.0E−4	**3,134**×
Ames	60	3.8E−1	21	3.5E−4	**1,082**×

Table 3: Mean certification time in seconds for FPA and Jia et al.'s (2022b) randomized ablation (RA). FPA is 2 to 3 orders of magnitude faster than baseline RA.

Method	24 Pixel Rect.		Square
	Min.	Max.	5×5
FPA Plurality ($T = 180$, ours)	⟵ 38.53 ⟶		37.77
FPA Run-Off ($T = 180$, ours)	⟵ 41.60 ⟶		40.95
Randomized Ablation (LF'20b)	⟵ 28.95 ⟶		28.21
Randomized Ablation (Jia'22b)	⟵ 37.31 ⟶		36.43
(De)Randomized Smoothing	0.0	72.68	57.69
BAGCERT	**43.11**	60.17	59.95
Patch IBP	—	—	30.30

Table 4: CIFAR10 certified patch accuracy (% – larger is better) for FPA, RA, and three dedicated patch defenses. FPA is competitive despite making fewer assumptions and providing stronger guarantees than patch defenses.

sponding method's robustness vs. accuracy tradeoff. When optimizing patch and median robustness, hyperparameters T and e were tuned on validation data.[6]

Patch Robustness We consider two CIFAR10 patch attacks: (1) a 5×5 pixel square (Levine and Feizi 2020a) and (2) all 24-pixel rectangles (e.g., 1×24 pixels, 24×1, 2×12, etc.), reporting each method's minimum and maximum certified accuracies across the eight valid shapes (Metzen and Yatsura 2021).

Hardware Requirements Both FPA and baseline RA have minimal hardware requirements. Experiments in this section were performed on a consumer desktop system containing an NVIDIA 3090 GPU, AMD 5950X CPU, and 64GB of RAM.

6.2 Main Results

Tables 1 and 2 summarize the median certified robustness and classification accuracy (resp.) for FPA and baseline RA. Table 3 details each method's mean certification time. Due to space, Tables 2 and 3 only report results for Jia et al.'s (2022b) (significantly) better performing version of baseline RA. Table 4 analyzes FPA as a patch defense. We briefly summarize the experiments' takeaways below. See supplemental Sections H.2 and H.3 for the full numerical results, including comparing the methods at additional robustness levels.

Takeaway #1: *FPA simultaneously provided larger and stronger median robustness guarantees than RA.* As Table 1 details, FPA's median certified robustness was 20–30% larger than RA for classification and 3 to 4× larger for regression. Importantly, FPA's certified feature guarantees apply to evasion, poisoning, and backdoor attacks, while baseline RA only covers evasion attacks.

Takeaway #2: *FPA's median robustness gains come at little cost in classification accuracy.* Table 2 reports FPA's classification accuracy at two robustness levels: (1) FPA's best median robustness (left) and (2) RA's best median robustness (middle, **bold**). Table 2 also reports RA's classification accuracy at its own best median robustness (last column). For CIFAR10 at median robustness of 10 pixels, FPA's classification accuracy was 10.2 percentage points (pp) better than RA (75.0% vs. 64.7%). At $r_{\text{med}} = 13$, FPA's CIFAR10 classification accuracy was 62.4%, only 2.3pp lower than RA's classification accuracy at $\rho_{\text{med}} = 10$. For Weather at median robustness 1, FPA's classification accuracy was 10.1pp better than RA (85.3% vs. 75.2%); even at $r_{\text{med}} = 4$, FPA's classification accuracy was 76.1%, 0.9pp better than RA at $\rho_{\text{med}} = 1$. For MNIST at median robustness 10, FPA's classification accuracy was 3pp better than RA (96.1% vs. 93.1%). At $r_{\text{med}} = 12$, FPA's MNIST classification accuracy was 5.9pp lower than RA's classification accuracy at $\rho_{\text{med}} = 10$ (87.2% vs. 93.1%).

Takeaway #3: *FPA certifies predictions 2 to 3 orders of magnitude faster than RA.* Table 3 compares the mean certification times using the hyperparameter settings with the

[6]Supplemental Sections H.2 and H.3 compare each method's certified accuracy across a range of hyperparameter settings.

best median robustness. To certify one prediction, Jia et al.'s (2022b) improved RA evaluates 100k ablated inputs. In contrast, FPA requires exactly T forward passes per prediction (one per submodel).

Takeaway #4: *FPA provides strong patch robustness without any assumptions about patch shape or the number of patches.* As Table 4 details, FPA certifies 41.6% of CIFAR10 predictions at $r = 24$ perturbed pixels (2.3% of d) – regardless of patch shape or number of patches. In contrast, (de)randomized smoothing's (Levine and Feizi 2020a) 24-pixel certified accuracy varies between 0% to 72.7% based on patch shape alone. BAGCERT's certified accuracy drops as low as 43.1% for 24-pixel column and row patches – only 1.5pp better than FPA. Unlike FPA, patch defenses' certified accuracy guarantees decline further or even evaporate under (1) multiple patches, (2) training data perturbations, and (3) amorphous shapes. While less effective in some settings than dedicated patch defenses that make stronger assumptions and weaker guarantees, FPA is still competitive, providing patch guarantees essentially for free.

Takeaway #5: *FPA is the first integrated defense to provide significant pointwise robustness guarantees over the union of evasion, backdoor, and poisoning attacks – ℓ_0 or otherwise.* Consider CIFAR10 ($n = 50{,}000$) where FPA feature robustness $r \geq 25$ (Table 4) certifies 41.0% of predictions' robustness against 1.25M arbitrarily perturbed pixels. In contrast, the only other certified defense robust over the union of evasion, backdoor, and poisoning attacks (Weber et al. 2020) certifies the equivalent of 3 or fewer arbitrarily perturbed CIFAR10 pixels (i.e., a total training and test ℓ_2 perturbation distance of ≤ 3). Moreover, FPA certifies $r \geq 7$ for 35.1% of Weather predictions ($n > 3$M – supplemental Table 28) – a pointwise guaranteed robustness of up to 21M arbitrarily perturbed feature values.

6.3 Model Training Time

Recall that baseline randomized ablation (RA) trains only a single model, while FPA requires training T submodels. However, FPA's training time is generally *not* T times longer than RA's. As supplemental Section H.8 details, an FPA ensemble is sometimes *faster* to train than RA models since FPA does not require ablated training. For instance, for tabular regression datasets Weather and Ames, FPA supports GBDTs out of the box, while RA requires the use of parametric models. This translates to FPA being $18\times$ and $60\times$ faster to train than RA for Weather ($T = 31$) and Ames ($T = 51$), respectively.

Moreover, Section 6.1 explains that for datasets CIFAR10 and MNIST, FPA used small, simple CNN submodels that are fast to train. When $T = 25$, FPA is $2\times$ and $4\times$ slower to train than RA for CIFAR10 and MNIST, respectively. At $T = 115$ for CIFAR10 and $T = 60$ for MNIST, FPA is only about $10\times$ slower to train than RA. Note that since each FPA submodel is independent, FPA training is embarrassingly parallel. Hence, while FPA is slower to train than RA for CIFAR10 and MNIST on a single system, FPA at maximum parallelism takes significantly less wall time to train than RA.

7 Conclusions

This paper proposes *feature partition aggregation* – a certified defense against the union of ℓ_0 evasion, poisoning, and backdoor attacks. FPA provided stronger and larger robustness guarantees than the state-of-the-art ℓ_0 evasion defense, randomized ablation. FPA's certified feature guarantees are particularly important for *vertically partitioned* data where a single compromised data source allows an attacker to arbitrarily modify a limited number of features for all instances – training and test.

To our knowledge, FPA is the first integrated defense that provides non-trivial pointwise robustness guarantees against the union of evasion, poisoning, and backdoor attacks – ℓ_0 or otherwise (Weber et al. 2020). Future work remains to develop other ℓ_p certified defenses over this union of attack types.

Acknowledgments

The authors thank Jonathan Brophy for helpful discussions and feedback on earlier drafts of this manuscript. This work was supported by a grant from the Air Force Research Laboratory and the Defense Advanced Research Projects Agency (DARPA) — agreement number FA8750-16-C-0166, subcontract K001892-00-S05, as well as a second grant from DARPA, agreement number HR00112090135. This work benefited from access to the University of Oregon high-performance computer, Talapas.

References

Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning Attacks against Support Vector Machines. In *Proceedings of the 29th International Conference on Machine Learning*, ICML'12. Edinburgh, Great Britain: PMLR.

Brophy, J.; Hammoudeh, Z.; and Lowd, D. 2023. Adapting and Evaluating Influence-Estimation Methods for Gradient-Boosted Decision Trees. *Journal of Machine Learning Research*, 24: 1–48.

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial Patch. arXiv:1712.09665.

Calzavara, S.; Lucchese, C.; Marcuzzi, F.; and Orlando, S. 2021. Feature Partitioning for Robust Tree Ensembles and their Certification in Adversarial Scenarios. *EURASIP Journal on Information Security*, 245–317.

Chen, R.; Li, Z.; Li, J.; Wu, C.; and Yan, J. 2022. On Collective Robustness of Bagging Against Data Poisoning. In *Proceedings of the 39th International Conference on Machine Learning*, ICML'22. PMLR.

Chiang, P.; Ni, R.; Abdelkader, A.; Zhu, C.; Studor, C.; and Goldstein, T. 2020. Certified Defenses for Adversarial Patches. In *Proceedings of the 8th International Conference on Learning Representations*, ICLR'20. Virtual Only.

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, ICML'19. PMLR.

Coleman, C. A.; Narayanan, D.; Kang, D.; Zhao, T.; Zhang, J.; Nardi, L.; Bailis, P.; Olukotun, K.; Ré, C.; and Zaharia, M. 2017. DAWNBench: An End-to-End Deep Learning Benchmark and Competition. In *Proceedings of the 2017 NeurIPS Workshop on Machine Learning Systems*. Long Beach, California, USA: Curran Associates, Inc.

De Cock, D. 2011. Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. *Journal of Statistics Education*, 19(3).

Hammoudeh, Z.; and Lowd, D. 2023. Reducing Certified Regression to Certified Classification for General Poisoning Attacks. In *Proceedings of the 1st IEEE Conference on Secure and Trustworthy Machine Learning*, SaTML'23.

Jia, J.; Cao, X.; and Gong, N. Z. 2021. Intrinsic Certified Robustness of Bagging against Data Poisoning Attacks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, AAAI'21.

Jia, J.; Liu, Y.; Cao, X.; and Gong, N. Z. 2022a. Certified Robustness of Nearest Neighbors against Data Poisoning and Backdoor Attacks. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, AAAI'22.

Jia, J.; Wang, B.; Cao, X.; Liu, H.; and Gong, N. Z. 2022b. Almost Tight ℓ_0-norm Certified Robustness of Top-k Predictions against Adversarial Perturbations. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR'22.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17.

Lee, G.-H.; Yuan, Y.; Chang, S.; and Jaakkola, T. 2019. Tight Certificates of Adversarial Robustness for Randomly Smoothed Classifiers. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, NeurIPS'19.

Levine, A.; and Feizi, S. 2020a. (De)Randomized Smoothing for Certifiable Defense against Patch Attacks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS'20. Red Hook, NY, USA: Curran Associates Inc.

Levine, A.; and Feizi, S. 2020b. Robustness Certificates for Sparse Adversarial Attacks by Randomized Ablation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI Press.

Levine, A.; and Feizi, S. 2021. Deep Partition Aggregation: Provable Defenses against General Poisoning Attacks. In *Proceedings of the 9th International Conference on Learning Representations*, ICLR'21. Virtual Only.

Levine, A. J.; and Feizi, S. 2022. Provable Adversarial Robustness for Fractional Lp Threat Models. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, AISTATS'22.

Li, X.; Dowsley, R.; and De Cock, M. 2021. Privacy-Preserving Feature Selection with Secure Multiparty Computation. In *Proceedings of the 38th International Conference on Machine Learning*, ICML'21.

Li, Y.; Wu, B.; Jiang, Y.; Li, Z.; and Xia, S. 2022. Backdoor Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Lin, M.; Chen, Q.; and Yan, S. 2014. Network in Network. In *Proceedings of the 2nd International Conference on Learning Representations*, ICLR'14.

Malinin, A.; Band, N.; Gal, Y.; Gales, M.; Ganshin, A.; Chesnokov, G.; Noskov, A.; Ploskonosov, A.; Prokhorenkova, L.; Provilkov, I.; Raina, V.; Raina, V.; Roginskiy, D.; Shmatova, M.; Tigas, P.; and Yangel, B. 2021. Shifts: A Dataset of Real Distributional Shift Across Multiple Large-Scale Tasks. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, NeurIPS'21. Curran Associates, Inc.

Metzen, J. H.; and Yatsura, M. 2021. Efficient Certified Defenses Against Patch Attacks on Image Classifiers. In *Proceedings of the 9th International Conference on Learning Representations*, ICLR'21.

Page, D. 2020. How to Train Your ResNet.

Rezaei, K.; Banihashem, K.; Chegini, A.; and Feizi, S. 2023. Run-Off Election: Improved Provable Defense against Data Poisoning Attacks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing Properties of Neural Networks. In *Proceedings of the 2nd International Conference on Learning Representations*, ICLR'14.

Wang, W.; Levine, A.; and Feizi, S. 2022a. Improved Certified Defenses against Data Poisoning with (Deterministic) Finite Aggregation. In *Proceedings of the 39th International Conference on Machine Learning*, ICML'22.

Wang, W.; Levine, A.; and Feizi, S. 2022b. Lethal Dose Conjecture on Data Poisoning. In *Proceedings of the 36th Conference on Neural Information Processing Systems*, NeurIPS'22. Curran Associates, Inc.

Weber, M.; Xu, X.; Karlaš, B.; Zhang, C.; and Li, B. 2020. RAB: Provable Robustness Against Backdoor Attacks. arXiv:2003.08904.

Wei, K.; Li, J.; Ma, C.; Ding, M.; Wei, S.; Wu, F.; Chen, G.; and Ranbaduge, T. 2022. Vertical Federated Learning: Challenges, Methodologies and Experiments. arXiv:2202.04309.