A Theory of Fault-Tolerant Learning

Changlong Wu [1] **Yifan Wang** [1] **Ananth Grama** [1]

Abstract

Developing machine learning models that account for potential faults encountered in real-world environments presents a fundamental challenge for mission-critical applications. In this paper, we introduce a novel theoretical framework grounded in learning theory for dealing with faults. In particular, we propose a framework called *fault-tolerant PAC learning*, aimed at identifying the most fault-tolerant models from a given hypothesis class (such as neural networks). We show that if faults occur randomly, fault-tolerant learning is equivalent to regular PAC learning. However, for *adversarial* faults, we show that the sample complexity of fault-tolerant PAC learning can grow linearly w.r.t. the number of perturbing functions induced by the faults, even for a hypothesis class with VC-dimension 1. We then provide a matching upper bound by restricting the number of perturbing functions. Finally, we show that the linear dependency on the number of perturbing functions can be substantially improved for *deletion faults* in neural networks. Our work provides a powerful formal framework and avenues for a number of future investigations on the precise characterization of fault-tolerant learning.

1. Introduction

Learning reliable and trustworthy models is an important challenge in contemporary machine learning. Mission-critical applications like autonomous vehicles (Ma et al., 2020; Hengstler et al., 2016), spacecraft (Izzo et al., 2019; Tipaldi et al., 2020), learning-enabled industrial control systems (Al Shahrani et al., 2022; Kershaw et al., 2021), and those operating in extreme environments (Verma et al., 2023; Khan, 2020; Lu, 2023), demand high performance, reliability, and graceful degradation. Constructing models that are resilient to hardware faults is a critical component of such systems.

Experimental results on fault tolerance highlight the fragility of modern machine learning systems to even minor faults. For example, recent work (Nguyen et al., 2023) shows that stuck-at faults (Phatak & Koren, 1995; Syed et al., 2023) in only a few critical intermediate outputs can significantly reduce the accuracy of deep neural networks like ResNet-18, rendering them ineffective. Adversarial faults in the form of bit-flip attacks (BFA) (Rakin et al., 2019) that alter small fractions of the weights can massively impact performance. These results underscore concerns relating to fault tolerance issues in machine learning systems operating in mission-critical settings. Due to the diversity of experimental settings, there has been a lack of theoretical formulations, models, and associated analyses that can inform practical solutions.

This paper introduces a novel theoretical framework for analyzing fault tolerance, proposing rigorous analytical tools to guide the development of reliable models in diverse faulty environments. Formally, let $\mathcal{X}$ be a set of features and $\mathcal{Y}$ be the set of outcomes. A *model* is a function $h_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$ indexed by $\mathbf{w}$. This could be, for instance, a function represented by a neural network with parameters (i.e., weights) $\mathbf{w}$. For any function $h_{\mathbf{w}}$, we associate a set of functions $\mathcal{U}(h_{\mathbf{w}}) \subset \mathcal{Y}^{\mathcal{X}}$, which includes all possible perturbed functions when a fault occurs during the execution of $h_{\mathbf{w}}$. For any given $(\mathbf{x}, y)$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{\geq 0}$, we define the *adversarial* faulty-loss as:

$$\ell_{\mathcal{U}}(h_{\mathbf{w}}; \mathbf{x}, y) = \sup_{h \in \mathcal{U}(h_{\mathbf{w}})} \ell(h(\mathbf{x}), y).$$

Note that the adversarial faulty loss measures the *worst* loss incurred by executing $h_{\mathbf{w}}$ on $\mathbf{x}$ when a fault occurs. Similarly, for any given distribution p over $\mathcal{U}(h_{\mathbf{w}})$, we define the *random* fault-loss as $\ell_{\mathcal{U}}(h_{\mathbf{w}}; \mathbf{x}, y) = \mathbb{E}_{h \sim p}[\ell(h(\mathbf{x}), y)]$. Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{H} = \{h_{\mathbf{w}} : \mathbf{w} \in \mathcal{W}\}$ be a class of models. Our goal is to learn a model $\hat{h} \in \mathcal{H}$ that achieves the minimal *faulty-regret* on $\mathcal{D}$ under fault-type $\mathcal{U}$:

$$\mathsf{Reg}_n(\hat{h}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell_{\mathcal{U}}(\hat{h}; \mathbf{x}, y)\right] - \inf_{h_{\mathbf{w}} \in \mathcal{H}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell_{\mathcal{U}}(h_{\mathbf{w}}; \mathbf{x}, y)\right] \quad (1)$$

[1]Center for Science of Information, Department of Computer Science, Purdue University, West Lafayette, USA. Correspondence to: Changlong Wu <wuchangl@hawaii.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

by accessing a sample $S_n = (\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$ sampled $i.i.d.$ from $\mathcal{D}$. Intuitively, we aim to learn a model $\hat{h}$ that is most tolerant to fault-type $\mathcal{U}$ when deployed in an environment where data is generated from $\mathcal{D}$. We say that a pair $(\mathcal{H},\mathcal{U})$ is (ϵ, δ)-fault tolerant PAC learnable if there exists a learning rule $\mathcal{A}$ and a number n, such that for *any* distribution $\mathcal{D}$, $\mathsf{Reg}_n(\mathcal{A}(S_n)) \leq \epsilon$ w.p. $\geq 1-\delta$ over an $i.i.d.$ sample S_n of length n from $\mathcal{D}$. The minimal number n allowing such a learning rule is referred to as the sample complexity.

Our contributions. Our contributions in this paper establish tight upper and lower bounds on the sample complexity for the fault-tolerant PAC learning across a wide range of model classes $\mathcal{H}$ and fault types $\mathcal{U}$. Specifically, we show that for any *random* fault types, the sample complexity of fault-tolerant learning is *equivalent* to (regular) PAC learning of the class $\mathcal{H}$, provided that $\mathcal{U}(h_{\mathbf{w}}) \subset \mathcal{H}$ [1]. For *adversarial* faults, we demonstrate that the sample complexity of fault-tolerant PAC learning can be arbitrarily larger than the complexity of (regular) PAC learning of $\mathcal{H}$. In particular, we show that there exists a class $\mathcal{H}$ of VC-dimension 1, such that for any number B, there exists a fault-type $\mathcal{U}$ with $|\mathcal{U}(h_{\mathbf{w}})| \leq B$ for all $h_{\mathbf{w}} \in \mathcal{H}$, where the fault-tolerant sample complexity grows *linearly* with B. We then provide a matching upper bound by showing that for *any* adversarial fault-type satisfying $|\mathcal{U}(h_{\mathbf{w}})| \leq B$ and $\mathcal{U}(h_{\mathbf{w}}) \subset \mathcal{H}$ for all $h_{\mathbf{w}} \in \mathcal{H}$, the sample complexity grows as $\tilde{O}(\frac{B\cdot \mathsf{VC}(\mathcal{H})+\log(1/\delta)}{\epsilon^2})$, where $\mathsf{VC}(\mathcal{H})$ denotes the VC-dimension of $\mathcal{H}$. Finally, we show that for neural networks with threshold activation functions and *deletion faults* (i.e., the fault occurs by deleting a subset of edges of the network), sample complexity grows polynomially with respect to the network size. This is surprising, given that the size of deletion fault-type can be *exponentially* large compared to the network size (i.e., any subset of edges may be deleted), yet a polynomial sample complexity is achievable even for adversarial fault-types.

In summary, our main contributions are: (i) a fundamentally new formulation of fault-tolerant PAC learning; (ii) tight characterizations of fault-tolerant PAC learnability for both random and adversarial faults; (iii) a case study of fault-tolerant PAC learnability of feed-forward neural networks with deletion faults, demonstrating potential real-world applications; and (iv) novel formal model, and algorithmic and analytic techniques that are of independent interest.

Related work A number of recent efforts focus on enhancing fault tolerance in machine learning. Effective abstractions of various fault types that occur in real world settings have been developed (Torres-Huitzil & Girau, 2017), with stuck-at (Phatak & Koren, 1995; Eghbal et al., 2015; Syed et al., 2023; Abraham & Fuchs, 1986) and random-bit flip (Dodd & Massengill, 2003) being particularly common, effectively representing permanent and transient (Sosnowski, 1994) faults. Investigations into the vulnerability of neural networks (Piuri, 2001; Rakin et al., 2019; Liu et al., 2017; Protzel et al., 1993; Nguyen et al., 2023) have revealed significant drops in accuracy due to even a small number of faults. Common strategies for developing fault-tolerant learning include: (i) adding redundancy (e.g. replication of critical nodes /weights) (Chu & Wah, 1990; Emmerson & Damper, 1993; Chin et al., 1994); (ii) modifying learning/ training algorithms (e.g. fault injection during training) (Arad & El-Amawy, 1997; Wei et al., 1996; Edwards & Murray, 1997); and (iii) optimization under fault tolerance constraints (Deodhare et al., 1998; Zhou & Chen, 2003). There have also been efforts focused on active fault tolerance, including error detection and recovery (Khunasaraphan et al., 1994; Hashmi et al., 2011; Deng et al., 2015). In contrast to these efforts, our work focuses on establishing the foundations of achievable performance in terms of tight constructive bounds for different learning models. To this end, our contributions complement these prior results.

2. Problem Formulation

Let $\mathcal{X}$ be the instance (feature) space, $\mathcal{Y} = [0,1]$ be the label space and $\mathcal{W}$ be an index set of hypotheses. For any function $f : \mathcal{W} \times \mathcal{X} \to \mathcal{Y}$, we define the hypothesis class induced by f as $\mathcal{H}_f = \{h_{\mathbf{w}}(\mathbf{x}) = f(\mathbf{w}, \mathbf{x}) : \mathbf{w} \in \mathcal{W}\}$. Note that, for any distinct $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{W}$, we will treat $h_{\mathbf{w}_1}$ and $h_{\mathbf{w}_2}$ as *different* functions, even if they instantiate the same functions in $[0,1]^{\mathcal{X}}$. This is unlike most of the classical learning theory literature, where properties of the hypothesis class depend only on its functionality. In our setup, we also distinguish hypotheses according to their *representation* (i.e., the index $\mathbf{w}$). This distinction can arise, for example, when two sets of parameters within a fixed neural network architecture represent the same function but have different fault-tolerance properties for a particular fault type.

Fault-types. A *fault-type* of the function class $\mathcal{H}_f$ is a map $\mathcal{U} : \mathcal{W} \to 2^{\Delta(\mathcal{W})}$, where $\Delta(\mathcal{W})$ denotes the class of all probability distributions overs $\mathcal{W}$, and $2^{\Delta(\mathcal{W})}$ is the power set (i.e., the class of all subsets) of $\Delta(\mathcal{W})$ [2]. In words, fault-type $\mathcal{U}$ maps each $\mathbf{w} \in \mathcal{W}$ to a *set* $\mathcal{U}(\mathbf{w})$ of probability distributions over $\mathcal{W}$. For any $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, $\mathbf{w} \in \mathcal{W}$, and loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, we define the *faulty-loss* as:

$$\ell_{\mathcal{U}}(h_{\mathbf{w}}; \mathbf{x}, y) = \sup_{p \in \mathcal{U}(\mathbf{w})} \mathbb{E}_{\mathbf{w}' \sim p}[\ell(h_{\mathbf{w}'}(\mathbf{x}), y)]. \tag{2}$$

[1] This is a very mild assumption, automatically satisfied for faults occurring in the parameters of a neural network.

[2] In this paper, we consider only *proper* fault types, where the perturbed functions must also be in $\mathcal{H}_f$. The case of *improper* fault types can be defined similarly.

Faulty-loss is interpreted as follows: given $(\mathbf{x}, y)$ and $\mathbf{w}$, an adversary first selects the worst distribution $p \in \mathcal{U}(\mathbf{w})$ and *perturbs* $\mathbf{w}$ by sampling $\mathbf{w}' \sim p$; the faulty-loss is then the expected loss of $h_{\mathbf{w}'}$ on $(\mathbf{x}, y)$. This formulation simultaneously covers *random* and *adversarial* faults by carefully defining the function $\mathcal{U}$, as illustrated in Examples 1 and 2 below:

Example 1 (Random Faults)**.** *If for any $\mathbf{w} \in \mathcal{W}$, we have $|\mathcal{U}(\mathbf{w})| = 1$, and denote $p_{\mathbf{w}}$ as the single element in $\mathcal{U}(\mathbf{w})$. Then, faulty loss $\ell_{\mathcal{U}}$ is reduced to the case of* random *faults such that for any sample $(\mathbf{x}, y)$, the loss is incurred by replacing $\mathbf{w}$ with a random sample $\mathbf{w}' \sim p_{\mathbf{w}}$. This can happen, for instance, with a random flip of the output of a single neuron in a neural network.*

Example 2 (Adversarial Faults)**.** *If for any $\mathbf{w} \in \mathcal{W}$, the set $\mathcal{U}(\mathbf{w})$ contains only singleton distributions (i.e., distributions that assign probability 1 on a single $\mathbf{w}'$), then, one may view the class $\mathcal{U}(\mathbf{w})$ as a subset of $\mathcal{W}$. The faulty loss $\ell_{\mathcal{U}}$ is reduced to the loss incurred by adversarially perturbing $\mathbf{w}$ to an element $\mathbf{w}' \in \mathcal{U}(\mathbf{w})$.*

It is worth noting that our definition of faulty loss in (2) also provides a way of modeling scenarios that interpolate between the pure random and pure adversarial fault types.

Fault-tolerant PAC learning. We now formulate the main learning paradigm of this paper. Let $\mathcal{D}$ be an unknown distribution over $\mathcal{X} \times \mathcal{Y}$. For any hypothesis class $\mathcal{H}_f$, loss ℓ, fault-type $\mathcal{U}$, and index $\mathbf{w} \in \mathcal{W}$, we define *fault-tolerant* risk of $h_{\mathbf{w}}$ as:

$$R_{\mathcal{U}}(h_{\mathbf{w}}; \mathcal{D}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell_{\mathcal{U}}(h_{\mathbf{w}}; \mathbf{x}, y)\right], \tag{3}$$

where $\ell_{\mathcal{U}}$ is the *faulty-loss* as defined in (2). Let $S_n = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be a sample of size n sampling $i.i.d.$ from $\mathcal{D}$. Our goal is to find an algorithm $\mathcal{A} : S_n \to \mathcal{H}_f$ that minimizes the *fault-tolerant* risk $R_{\mathcal{U}}(\mathcal{A}(S_n); \mathcal{D})$ as in (3). We say a pair $(\mathcal{H}_f, \mathcal{U})$ is *fault-tolerant* PAC-learnable under loss ℓ if there exists an algorithm $\mathcal{A}$ such that for any $\epsilon, \delta > 0$, there exists a number $n := n(\epsilon, \delta)$ such that:

$$\Pr_{S_n\sim\mathcal{D}}\left[R_{\mathcal{U}}(\mathcal{A}(S_n); \mathcal{D}) - \inf_{h_w\in\mathcal{H}_f} R_{\mathcal{U}}(h_w; \mathcal{D}) \geq \epsilon\right] \leq \delta. \tag{4}$$

For any $\epsilon, \delta > 0$, the *minimal* number n achievable by an algorithm $\mathcal{A}$ that satisfies (4), is defined to be the *sample complexity* of (ϵ, δ)-fault-tolerant PAC learning of $(\mathcal{H}_f, \mathcal{U})$. We denote this number by $\mathcal{M}(\epsilon, \delta; \mathcal{H}_f, \mathcal{U})$.

Fault-tolerant PAC learning aims to find the most robust hypothesis $\hat{h} \in \mathcal{H}_f$ that achieves the minimal fault-tolerant risk $R_{\mathcal{U}}(\hat{h}; \mathcal{D})$ under perturbation of $\mathcal{U}$.

Analogous to classical PAC learning framework, we can also define the following *fault-tolerant* Empirical Risk Minimization (ERM) rule. Let $S_n = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be any sample of size n, the *empirical* fault-tolerant risk of any $h_{\mathbf{w}}$ is defined as:

$$\hat{R}_{\mathcal{U}}(h_{\mathbf{w}}; S_n) = \frac{1}{n}\sum_{i=1}^{n} \ell_{\mathcal{U}}(h_{\mathbf{w}}; (\mathbf{x}_i, y_i)). \tag{5}$$

The *fault-tolerant* ERM rule corresponds to finding $\hat{h} \in \mathcal{H}_f$ that satisfies:

$$\hat{R}_{\mathcal{U}}(\hat{h}; S_n) = \inf_{h_{\mathbf{w}}\in\mathcal{H}_f}\{\hat{R}_{\mathcal{U}}(h_{\mathbf{w}}; S_n)\}. \tag{6}$$

Robustness v.s. Fault-Tolerance. Our fault-tolerance framework is closely related to the concept of adversarial robust PAC learning, as introduced in (Montasser et al., 2019). Instead of perturbing the index $\mathbf{w}$, the adversarial robust setting perturbs any input $\mathbf{x}$ with an adversarially selected sample $\mathbf{x}'$ chosen from a set $\mathcal{V}(\mathbf{x}) \subseteq \mathcal{X}$. Here, $\mathcal{V} : \mathcal{X} \to 2^{\mathcal{X}}$ is a function that specifies possible perturbations. The goal is to find a function $\hat{h}$ (not necessary in $\mathcal{H}_f$) that minimizes the following *robustness risk*:

$$R_{\mathcal{V}}(\hat{h}; \mathcal{D}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\sup_{\mathbf{x}'\in\mathcal{V}(\mathbf{x})} \ell(\hat{h}(\mathbf{x}'), y)\right].$$

Given any $\mathcal{V}$ as above and function $h_{\mathbf{w}} \in \mathcal{H}_f$, we can define a set of functions $\mathcal{H}_{\mathbf{w}} = \{h \in \mathcal{Y}^{\mathcal{X}} : \forall \mathbf{x} \in \mathcal{X}, \exists \mathbf{x}' \in \mathcal{V}(\mathbf{x}) \text{ s.t. } h(\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x}')\}$. Now, if we define a *fault-type* $\mathcal{U}$ by mapping each $\mathbf{w}$ to the set of all singleton distributions over functions in $\mathcal{H}_{\mathbf{w}}$, then robust PAC learning is reduced to our fault-tolerant PAC learning. Observe, however, that the fault-type $\mathcal{U}$ we constructed here is *improper* since $\mathcal{H}_{\mathbf{w}}$ need not be a subset of $\mathcal{H}_f$.

The goal of this paper is to characterize how the structures of $\mathcal{H}_f$ and $\mathcal{U}$ impact our fault-tolerant PAC learnability, to find learning rules, and to derive upper and lower bounds on the sample complexity for selected natural classes.

3. Main Results

In this section, we provide tight characterizations on the fault-tolerant PAC learnability of finite VC-dimensional classes and various fault-types.

3.1. General Characterizations

We initiate our discussion with some basic properties of our *fault-tolerant* PAC learning framework by focusing on *binary-valued* hypothesis classes. For clarity of presentation, we consider the case when $\mathcal{Y} = \{0, 1\}$ and loss $\ell(y, y') = 1\{y \neq y'\}$ is the mis-classification loss. The following fact is easy to observe:

Fact 1. *For any probability distribution p over $\mathcal{W}$ with mis-classification loss ℓ, we have for any $(\mathbf{x}, y)$:*

$$\mathbb{E}_{\mathbf{w}'\sim p}[\ell(h_{\mathbf{w}'}(\mathbf{x}), y)] = |\mathbb{E}_{\mathbf{w}'\sim p}[h_{\mathbf{w}'}(\mathbf{x})] - y|.$$

Proof. We argue by cases: if $y = 0$ then $\ell(h_{\mathbf{w}'}(\mathbf{x}), y) = h_{\mathbf{w}'}(\mathbf{x})$ and the result trivially holds; if $y = 1$, then we have $\ell(h_{\mathbf{w}'}(\mathbf{x}), y) = 1 - h_{\mathbf{w}'}(\mathbf{x})$, and therefore the result follows by taking $\mathbb{E}_{\mathbf{w}'}$ on both sides. □

Warm-up: Random Fault-Types. We start with the simpler *random* fault types introduced in Example 1. Let $\mathcal{U}$ be a *random* fault-type, i.e., for any $\mathbf{w} \in \mathcal{W}$ we have $|\mathcal{U}(\mathbf{w})| = 1$. Define for any $\mathbf{w} \in \mathcal{W}$ the function:

$$g_{\mathbf{w}}(\mathbf{x}) = \mathbb{E}_{\mathbf{w}' \sim p_{\mathbf{w}}}[h_{\mathbf{w}'}(\mathbf{x})],$$

where $p_{\mathbf{w}}$ is the single distribution in $\mathcal{U}(\mathbf{w})$. We have by Fact 1 that the *fault-tolerant* PAC learnability of the pair $(\mathcal{H}_f, \mathcal{U})$ under mis-classification loss is equivalent to the (regular) PAC learning of $\mathcal{G} = \{g_{\mathbf{w}}(\mathbf{x}) : \mathbf{w} \in \mathcal{W}\}$ under *absolute loss*. Therefore, we have:

Proposition 1. *Let $\mathcal{H}_f$ be a binary-valued class, $\mathcal{U}$ be any fault-type such that $|\mathcal{U}(\mathbf{w})| = 1$ for all $\mathbf{w} \in \mathcal{W}$, and $\mathcal{G} \subset [0,1]^{\mathcal{X}}$ be as defined above. Then $(\mathcal{H}_f, \mathcal{U})$ is fault-tolerant PAC learnable under mis-classification loss if and only if $\mathcal{G}$ is PAC learnable under absolute loss. Moreover, these two learning problems have the same sample complexity.*

Note that, although Proposition 1 provides a complete characterization of fault-tolerant PAC learnability by reducing it to PAC learning of class $\mathcal{G}$, the structure of $\mathcal{G}$ may still be complicated. Fortunately, by definition of function $g_{\mathbf{w}}$, we have the class $\mathcal{G}$ is essentially a subset of the *convex hull* of $\mathcal{H}_f$.

To proceed, we first recall the definition of Rademacher complexity of a hypothesis class $\mathcal{H}$ with horizon n as:

$$\mathsf{Rad}_n(\mathcal{H}) = \sup_{\mathbf{x}^n \in \mathcal{X}^n} \mathbb{E}_{\epsilon^n}\left[\sup_{h \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i h(\mathbf{x}_i)\right],$$

where ϵ^n is sampled from the uniform distribution over $\{-1, +1\}^n$. The following key lemma relates the Rademacher complexities between $\mathcal{H}_f$ and $\mathcal{G}$:

Lemma 1. *We have $\mathsf{Rad}_n(\mathcal{G}) \leq \mathsf{Rad}_n(\mathcal{H}_f)$.*

Proof. By (Shalev-Shwartz & Ben-David, 2014, Lem. 26.7) we know that the Rademacher complexity of a class equals the Rademacher complexity of its convex hull. The lemma follows since $\mathcal{G}$ is within the convex hull of $\mathcal{H}_f$. □

We now arrive at our first main result that completely characterizes the sample complexity of fault-tolerant learning of binary-valued hypothesis with a *random* fault-type.

Theorem 1. *Let $\mathcal{H}_f$ be a binary valued class with finite VC-dimension $\mathsf{VC}(\mathcal{H}_f)$. Then, for* any *fault-type $\mathcal{U}$ with $|\mathcal{U}(\mathbf{w})| = 1$ for all $\mathbf{w} \in \mathcal{W}$, we have the sample complexity*

$$\mathcal{M}(\epsilon, \delta; \mathcal{H}_f, \mathcal{U}) \leq O\left(\frac{\mathsf{VC}(\mathcal{H}_f) + \log(1/\delta)}{\epsilon^2}\right).$$

Moreover, this bound is tight when $\mathcal{U}$ introduces no faults.

Proof. By Proposition 1, we have the sample complexity $\mathcal{M}(\epsilon, \delta; \mathcal{H}_f, \mathcal{U})$ equals the sample complexity of PAC learning $\mathcal{G}$ under absolute loss. Using a simple symmetrization argument as (Shalev-Shwartz & Ben-David, 2014, Thm. 26.5) and 1-Lipschitz property of absolute loss, we have the generalization error ϵ of $\mathcal{G}$ is upper bounded by $O(\mathsf{Rad}_n(\mathcal{G}) + \sqrt{\log(1/\delta)/n})$ for a sample of size n via ERM rule. The theorem then follows by Lemma 1, the fact $\mathsf{Rad}_n(\mathcal{H}_f) \leq O(\sqrt{\mathsf{VC}(\mathcal{H}_f)/n})$ (Wainwright, 2019, Example 5.24), and rewriting n w.r.t. ϵ, δ. □

Remark 1. *Note that, Theorem 1 shows that, statistically, introducing random faults does not make learning harder. Moreover, optimal sample complexity is achievable by the fault-tolerant ERM rule as in (6).*

Example 3. *Let $f(\mathbf{w}, \mathbf{x}) = 1\{\langle \mathbf{w}, \mathbf{x} \rangle \geq 0\}$ with $\mathbf{w}, \mathbf{x} \in \mathbb{R}^d$ be the linear threshold function. Consider the fault-type that flips each coordinate $\mathbf{w}[i]$ with $-\mathbf{w}[i]$ independently with certain probability $\sigma > 0$. Although the flip faults can alter the original functions in a complicated way, Theorem 1 shows that such a pairs can be fault-tolerant PAC learned with sample complexity $O(\frac{d+\log(1/\delta)}{\epsilon^2})$.*

Characterization of general fault-types. We now deal with the general fault-type $\mathcal{U}$ with $|\mathcal{U}(\mathbf{w})| \geq 1$ for all $\mathbf{w} \in \mathcal{W}$. Recall that $\mathcal{U}(\mathbf{w})$ is a set of distributions over $\mathcal{W}$. This can be interpreted as, for instance, all possible random perturbations without knowing the exact perturbing distribution a-priori. If the distributions in $\mathcal{U}(\mathbf{w})$ are *singleton* distributions, this case reduces to the full *adversary* case as in Example 2.

Now, for any $\mathcal{H}_f, \mathcal{U}$, and $\mathbf{w} \in \mathcal{W}$ we define two functions:

$$g^{\min}_{\mathbf{w}}(\mathbf{x}) = \inf_{p \in \mathcal{U}(\mathbf{w})} \mathbb{E}_{\mathbf{w}' \sim p}[h_{\mathbf{w}'}(\mathbf{x})],$$

and

$$g^{\max}_{\mathbf{w}}(\mathbf{x}) = \sup_{p \in \mathcal{U}(\mathbf{w})} \mathbb{E}_{\mathbf{w}' \sim p}[h_{\mathbf{w}'}(\mathbf{x})].$$

We observe the following fact:

Fact 2. *For the mis-classification loss ℓ, we have:*

$$\ell_{\mathcal{U}}(h_{\mathbf{w}}; \mathbf{x}, y) = \begin{cases} g^{\max}_{\mathbf{w}}(\mathbf{x}), & \text{if } y = 0 \\ 1 - g^{\min}_{\mathbf{w}}(\mathbf{x}), & \text{if } y = 1 \end{cases}$$

where $\ell_{\mathcal{U}}(h_{\mathbf{w}}; \mathbf{x}, y)$ is the faulty-loss as in (2).

Proof. This follows directly from Fact 1 and definition in (2) by arguing on cases of y. □

Let $g_{\mathbf{w}}(\mathbf{x}) = (g^{\min}_{\mathbf{w}}(\mathbf{x}), g^{\max}_{\mathbf{w}}(\mathbf{x}))$ be a function mapping $\mathcal{X} \to [0,1]^2$ and $\mathcal{G} = \{g_{\mathbf{w}} : \mathbf{w} \in \mathcal{W}\}$. We know that

the fault-tolerant PAC learnability of $(\mathcal{H}_f,\mathcal{U})$ under misclassification loss is equivalent to the PAC learning of $\mathcal{G}$ under the following loss:

$$\tilde{\ell}(g_{\mathbf{w}};\mathbf{x},y) = (1-y)g^{\max}_{\mathbf{w}}(\mathbf{x}) + y(1-g^{\min}_{\mathbf{w}}(\mathbf{x})).$$

Observe, however, that functions $g^{\min}_{\mathbf{w}}(\mathbf{x})$ and $g^{\max}_{\mathbf{w}}(\mathbf{x})$ need *not* be convex combinations of functions in $\mathcal{H}_f$. Therefore, class $\mathcal{G}$ can have complicated structures. Specifically, for *adversarial* fault-type $\mathcal{U}$, function $g_{\mathbf{w}}$ can be compactly expressed as $g_{\mathbf{w}} : \mathcal{X} \to \{*,0,1\}$ such that[3] $g_{\mathbf{w}}(\mathbf{x}) = 0$ if $\forall \mathbf{w}' \in \mathcal{U}(\mathbf{w})$, $h_{\mathbf{w}'}(\mathbf{x}) = 0$; $g_{\mathbf{w}}(\mathbf{x}) = 1$ if $\forall \mathbf{w}' \in \mathcal{U}(\mathbf{w})$, $h_{\mathbf{w}'}(\mathbf{x}) = 1$; and $g_{\mathbf{w}}(\mathbf{x}) = *$ otherwise. Moreover, the loss $\tilde{\ell}(*,y) = 1$ for all $y \in \{0,1\}$ and $\tilde{\ell}(y',y) = 1\{y' \neq y\}$ for all $y', y \neq *$.

The following lemma demonstrates that even for a class $\mathcal{H}_f$ with VC-dimension 1, the sample complexity of fault-tolerant learning $(\mathcal{H}_f,\mathcal{U})$ can be *arbitrarily* large:

Lemma 2. *There exists a class $\mathcal{H}_f$ of VC-dimension 1, such that for any $B > 0$, there exists a fault-type $\mathcal{U}$ with sample complexity $\mathcal{M}(\frac{1}{8},\frac{1}{7};\mathcal{H}_f,\mathcal{U}) \geq \Omega(B)$.*

Proof. Let $\mathcal{X},\mathcal{W} := \mathbb{N}$ the set of natural numbers. For any $\mathbf{w} \in \mathcal{W}$, we define a function $h_{\mathbf{w}}(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{w}$ and $h_{\mathbf{w}}(\mathbf{x}) = 0$ otherwise. Denote $\mathcal{H}_f = \{h_{\mathbf{w}}(\mathbf{x}) : \mathbf{w} \in \mathcal{W}\}$. It is easy to observe that $\mathsf{VC}(\mathcal{H}_f) = 1$.

We now construct, for any natural number $B > 0$, an *adversarial* fault-type $\mathcal{U}$ that attains our claimed lower bound $\Omega(B)$ for any learning rule. Let $A_1,\cdots,A_N$ be an enumeration of all subsets $A_i \subset \{1,\cdots,3B\}$ of size $|A_i| = B$, where $N = \binom{3B}{B}$. For $\mathbf{w} \in \{1,\cdots,N\}$, we define $\mathcal{U}(\mathbf{w})$ as the class of all *singleton* distributions over $A_{\mathbf{w}}$ (i.e., $\mathcal{U}(\mathbf{w})$ is effectively $A_{\mathbf{w}}$). For any other $\mathbf{w}$, we set $\mathcal{U}(\mathbf{w}) = \mathcal{U}(1)$. Note that, $|\mathcal{U}(\mathbf{w})| = B$ for all $\mathbf{w} \in \mathcal{W}$. Moreover, for $\mathbf{w} \in \{1,\cdots,N\}$ and $y = 0$, we have $\ell_{\mathcal{U}}(h_{\mathbf{w}};\mathbf{x},y) = 1$ if and only if $\mathbf{x} \in A_{\mathbf{w}}$ (by definition of $\mathcal{H}_f,\mathcal{U}$ and Fact 2).

We now construct the distribution $\mathcal{D}$ using a probabilistic argument, resembling those used in (Montasser et al., 2019). For any $A \subset \{1,\cdots,3B\}$, we define $\mathcal{D}_A$ as the distribution with $\mathbf{x}$ uniform over A and with y being constant 0. Let μ be the distribution uniform over all subsets of $\{1,\cdots,3B\}$ of size $2B$ and $\mathcal{A}$ be any learning rule. We have:

$$\begin{aligned}
&\mathbb{E}_{A\sim\mu}\left[\mathbb{E}_{S_B\sim\mathcal{D}_A}\left[R_{\mathcal{U}}(\mathcal{A}(S_B);\mathcal{D}_A) - \inf_{h_{\mathbf{w}}} R_{\mathcal{U}}(h_{\mathbf{w}};\mathcal{D}_A)\right]\right] \\
&\overset{(a)}{\geq} \mathbb{E}_{A\sim\mu}\left[\mathbb{E}_{S_B\sim\mathcal{D}_A}\left[R_{\mathcal{U}}(\mathcal{A}(S_B);\mathcal{D}_A)\right]\right] \\
&\overset{(b)}{=} \frac{1}{2B}\mathbb{E}_{A_S}\left[\mathbb{E}_{A|A_S}\left[\sum_{\mathbf{x}\in A}\ell_{\mathcal{U}}(\hat{h};\mathbf{x},0)\right]\right] \\
&\overset{(c)}{=} \frac{1}{2B}\mathbb{E}_{A_S}\left[C + \mathbb{E}_{A|A_S}\left[\sum_{\mathbf{x}\in A\setminus A_S}\ell_{\mathcal{U}}(\hat{h};\mathbf{x},0)\right]\right] \\
&\overset{(d)}{\geq} \frac{1}{2B}\mathbb{E}_{A_S}[C + (B-C)/2] \geq \frac{1}{4},
\end{aligned}$$

where (a) follows from the fact that the hypothesis with index corresponds to subset $\{1,\cdots,3B\}\setminus A$ attains 0 fault-tolerant risk on $\mathcal{D}_A$ (by construction of $\mathcal{U}$ and $y = 0$); (b) follows by switching the order of expectation and A_S corresponds to the features in S_B; (c) follows by setting C to be the loss incurred by $\hat{h} := \mathcal{A}(S_B)$ on A_S; (d) follows from the fact that the loss incurred by $\hat{h}$ equals $|A_{\mathbf{w}} \cap A|$, where $\mathbf{w}$ is the index for $\hat{h} \in \mathcal{H}_f$ (taking $A_{\mathbf{w}}$ being A_1 if $\mathbf{w} \geq N$); by conditioning on A_S, the set $A\setminus A_S$ is uniformly distributed over $\{1,\cdots,3B\}\setminus A_S$ with size $2B - |A_S|$, we have the expected size of $|A_{\mathbf{w}} \cap A|$ equals $C + (|A_{\mathbf{w}}| - C)\frac{2B-|A_S|}{3B-|A_S|} \geq C + (|A_{\mathbf{w}}| - C)\frac{1}{2} = C + (B-C)/2$ since $|A_S| \leq B$ and $|A_{\mathbf{w}}| = B$.

Therefore, for any learning rule $\mathcal{A}$, there must exists a distribution $\mathcal{D}_A$ such that the *expected* fault-tolerant regret is lower bounded by $\frac{1}{4}$ with sample size $\leq B$. The lemma then follows by a standard argument for converting the expected lower bound to high probability lower bound as in (Shalev-Shwartz & Ben-David, 2014, Thm. 5.1). □

Note that, Lemma 2 demonstrates that one cannot hope to obtain a *uniform* sample complexity bound that is independent of the fault type $\mathcal{U}$ even for classes with finite VC-dimension 1. This is in contrast to the random faults case established in Theorem 1. Therefore, one must introduce certain constrains on $\mathcal{U}$ in order to obtain meaningful results. Inspired by Theorem 1, a natural constraint would be to bound the size of $\mathcal{U}(\mathbf{w})$. Doing so allows us to obtain our second main result of this paper:

Theorem 2. *Let $\mathcal{H}_f$ be a binary valued class of finite VC-dimension. Then, for any fault-type $\mathcal{U}$ with $|\mathcal{U}(\mathbf{w})| \leq B$ for all $\mathbf{w} \in \mathcal{W}$, we have:*

$$\mathcal{M}(\epsilon,\delta;\mathcal{H}_f,\mathcal{U}) \leq O\left(\frac{B^2\cdot\mathsf{VC}(\mathcal{H}_f) + \log(1/\delta)}{\epsilon^2}\right).$$

If, in addition, $\mathcal{U}$ is adversarial *fault-type, then:*

$$\mathcal{M}(\epsilon,\delta;\mathcal{H}_f,\mathcal{U}) \leq O\left(\frac{B\cdot\mathsf{VC}(\mathcal{H}_f)\log(1/\epsilon) + \log(1/\delta)}{\epsilon^2}\right),$$

and the linear dependency of on B cannot be improved.

Proof. The lower bound on the linear dependency of B follows directly from Lemma 2. We now prove the upper bound. By Fact 2 and the discussion that follows, we have the fault-tolerant sample complexity is equivalent to the sample complexity of PAC learning $\mathcal{G}$ under loss $\tilde{\ell}$. Using

[3] Recall that for *adversarial* fault-type $\mathcal{U}$, the set $\mathcal{U}(\mathbf{w})$ can be viewed as a subset of $\mathcal{W}$.

a standard symmetrization argument as (Shalev-Shwartz & Ben-David, 2014, Thm. 26.5), we have the generalization error (for *fault-tolerant* ERM rule) with sample size n is upper bounded by (w.p. $\geq 1-\delta$):

$$O(\mathsf{Rad}_n(\tilde{\ell}\circ\mathcal{G})+\sqrt{\log(1/\delta)/n}),$$

where (for ϵ^n uniform over $\{-1,+1\}^n$):

$$\mathsf{Rad}_n(\tilde{\ell}\circ\mathcal{G})=\sup_{\mathbf{x}^n,y^n}\mathbb{E}_{\epsilon^n}\left[\sup_{g_{\mathbf{w}}\in\mathcal{G}}\frac{1}{n}\sum_{i=1}^n\epsilon_i\tilde{\ell}(g_{\mathbf{w}};\mathbf{x}_i,y_i)\right].$$

We now fix any $\mathbf{x}^n$ and y^n. Note that if $y_i=0$ then $\tilde{\ell}(g_{\mathbf{w}};\mathbf{x}_i,y_i)=g_{\mathbf{w}}^{\max}$, else we have $\tilde{\ell}(g_{\mathbf{w}};\mathbf{x}_i,y_i)=1-g_{\mathbf{w}}^{\min}$. Therefore we have:

$$\sum_{i=1}^n\epsilon_i\tilde{\ell}(g_{\mathbf{w}};\mathbf{x}_i,y_i)=\sum_{\{i:y_i=0\}}\epsilon_i g_{\mathbf{w}}^{\max}(\mathbf{x}_i)+\sum_{\{i:y_i=1\}}\epsilon_i(1-g_{\mathbf{w}}^{\min}(\mathbf{x}_i)).$$

Taking the operator $\mathbb{E}_{\epsilon^n}\sup_{g_{\mathbf{w}}}$ on both sides and noticing that the constant 1 vanishes, we have:

$$\mathbb{E}_{\epsilon^n}\left[\sup_{g_{\mathbf{w}}}\sum_{i=1}^n\epsilon_i\tilde{\ell}(g_{\mathbf{w}};\mathbf{x}_i,y_i)\right]\leq\mathbb{E}_{\epsilon^n}\left[\sup_{\mathbf{w}}\sum_{\{i:y_i=0\}}\epsilon_i g_{\mathbf{w}}^{\max}(\mathbf{x}_i)\right]+\mathbb{E}_{\epsilon^n}\left[\sup_{\mathbf{w}}\sum_{\{i:y_i=1\}}\epsilon_i g_{\mathbf{w}}^{\min}(\mathbf{x}_i)\right],\tag{7}$$

where the inequality follows by $\sup(A+B)\leq\sup A+\sup B$. We now upper bound the first term in the RHS of (7) (the second term can be bounded similarly). For any $\mathbf{w}\in\mathcal{W}$, we denote $\mathcal{H}_{\mathbf{w}}=\{h_p(\mathbf{x})=\mathbb{E}_{\mathbf{w}'\sim p}[h_{\mathbf{w}'}(\mathbf{x})]:p\in\mathcal{U}(\mathbf{w})\}$. We have $g_{\mathbf{w}}^{\max}(\mathbf{x})=\sup_{h\in\mathcal{H}_{\mathbf{w}}}h(\mathbf{x})$ and $|\mathcal{H}_{\mathbf{w}}|\leq B$ for all $\mathbf{w}\in\mathcal{W}$.

To proceed, we establish the following key inequality. For *any* function $F:\mathcal{W}\to\mathbb{R}$ and i with $y_i=0$, we have:

$$\mathbb{E}_{\epsilon_i}\left[\sup_{\mathbf{w}}\epsilon_i g_{\mathbf{w}}^{\max}(\mathbf{x}_i)+F(\mathbf{w})\right]\leq\mathbb{E}_{\epsilon'^B}\left[\sup_{\mathbf{w}}\sum_{h_j\in\mathcal{H}_{\mathbf{w}}}\epsilon'_j h_j(\mathbf{x}_i)+F(\mathbf{w})\right],\tag{8}$$

where ϵ'^B is a fresh sample uniform over $\{-1,+1\}^B$. To see this, we observe that $g_{\mathbf{w}}^{\max}(\mathbf{x})=\sup_{h\in\mathcal{H}_{\mathbf{w}}}h(\mathbf{x})$. Therefore, there exists $h_{j^*}\in\mathcal{H}_{\mathbf{w}}$ such that $g_{\mathbf{w}}^{\max}(\mathbf{x}_i)=h_{j^*}(\mathbf{x}_i)$. We now fix ϵ_i and couple $\epsilon_{j^*}=\epsilon_i$. Let $\hat{\mathbf{w}}$ attain the $\sup_{\mathbf{w}}$ term in the LHS of (8). We have:

$$\sum_{h_j\in\mathcal{H}_{\hat{\mathbf{w}}}}\epsilon'_j h_j(\mathbf{x}_i)+F(\hat{\mathbf{w}})\leq\sup_{\mathbf{w}}\sum_{h_j\in\mathcal{H}_{\mathbf{w}}}\epsilon'_j h_j(\mathbf{x}_i)+F(\mathbf{w}).$$

Taking expectation over ϵ'_js for $j\neq j^*$, we have the LHS equals $\epsilon_{j^*}h_{j^*}(\mathbf{x}_i)+F(\hat{\mathbf{w}})$. By definition of $\hat{\mathbf{w}}$, we have $\mathbb{E}_{\epsilon_{j^*}}[\epsilon_{j^*}h_{j^*}(\mathbf{x}_i)+F(\hat{\mathbf{w}})]$ equals the LHS of (8). Therefore, the inequality (8) follows.

Repeatably using (8), we have:

$$\begin{aligned}\mathbb{E}_{\epsilon^n}\left[\sup_{\mathbf{w}}\sum_{\{i:y_i=0\}}\epsilon_i g_{\mathbf{w}}^{\max}(\mathbf{x}_i)\right]\leq&\ \mathbb{E}_{\epsilon^{Bn}}\left[\sup_{\mathbf{w}}\sum_{h_j\in\mathcal{H}_{\mathbf{w}}}\sum_{i=1}^n\epsilon_{(j-1)B+i}h_j(\mathbf{x}_i)\right]\\ \leq&\ \mathbb{E}_{\epsilon^{Bn}}\left[\sup_{h^B\in\mathsf{co}(\mathcal{H}_f)}\sum_{j=1}^B\sum_{i=1}^n\epsilon_{(j-1)B+i}h_j(\mathbf{x}_i)\right]\\ \leq&\ \mathbb{E}_{\epsilon^{Bn}}\left[\sum_{j=1}^B\sup_{h_j\in\mathsf{co}(\mathcal{H}_f)}\sum_{i=1}^n\epsilon_{(j-1)B+i}h_j(\mathbf{x}_i)\right]\\ \leq&\ Bn\mathsf{Rad}_n(\mathsf{co}(\mathcal{H}_f))=Bn\mathsf{Rad}_n(\mathcal{H}_f),\end{aligned}$$

where $\mathsf{co}(\mathcal{H}_f)$ denotes the *convex hull* of $\mathcal{H}_f$ and the last equality follows by (Shalev-Shwartz & Ben-David, 2014, Lem. 26.7). Since $\mathsf{Rad}_n(\mathcal{H}_f)\leq O(\sqrt{\mathsf{VC}(\mathcal{H}_f)/n})$, we have the generalization error upper bounded by $O(B\sqrt{\mathsf{VC}(\mathcal{H}_f)/n})=O(\sqrt{B^2\mathsf{VC}(\mathcal{H}_f)/n})$. The first upper bound then follows by rewrite n w.r.t. ϵ,δ.

To prove the second upper bound, we observe that $\mathcal{H}_{\mathbf{w}}\subset\mathcal{H}_f$ for *adversarial* fault-types. By Sauer's lemma (Shalev-Shwartz & Ben-David, 2014), we have the number of such $\mathcal{H}_{\mathbf{w}}$s restricted on any $\mathbf{x}^n$ is upper bounded by $n^{B\cdot\mathsf{VC}(\mathcal{H}_f)}$. Since $g_{\mathbf{w}}^{\max}$ and $g_{\mathbf{w}}^{\min}$ are completely determined by $\mathcal{H}_{\mathbf{w}}$, we have the size of restriction of $\mathcal{G}$ on $\mathbf{x}^n$ is upper bounded by $n^{B\cdot\mathsf{VC}(\mathcal{H}_f)}$. Applying the Massart's lemma (Shalev-Shwartz & Ben-David, 2014), we arrive at an $O(\sqrt{B\cdot\mathsf{VC}(\mathcal{H}_f)\log n/n})$ upper bound for $\mathsf{Rad}_n(\tilde{\ell}\circ\mathcal{G})$. This completes the proof. □

3.2. Neural Networks with Deletion Faults

In this section, we focus on an important hypothesis class: feed-forward neural networks with the threshold activation function. For simplicity, we assume that there are L layers, each with N neurons (except for the last layer, which has 1 neuron), and adjacent layers are fully connected[4]. We consider specifically the *threshold* activation function $\sigma(x)=1\{x\geq 0\}$. Let $\mathcal{X}=\mathbb{R}^d$ be the input space, and E be the set of all edges (weights), where $|E|=dN+(L-2)N^2+N$. The index space is therefore $\mathcal{W}=\mathbb{R}^{|E|}$, which assigns the weight of each coordinate of $\mathbf{w}$ to an edge in E. Note that, for any $\mathbf{w}\in\mathcal{W}$, the architecture defines a hypothesis $h_{\mathbf{w}}:\mathcal{X}\to\{0,1\}$. The hypothesis class we consider is therefore $\mathcal{H}_f=\{h_{\mathbf{w}}:\mathbf{w}\in\mathcal{W}\}$. It

[4]General architectures can be analyzed similarly.

is well known that $\mathsf{VC}(\mathcal{H}_f) \leq O(|E|\log|E|)$ (Bartlett & Maass, 2003).

We now consider the following *deletion faults*. Let $\mathcal{S}$ be a set of subsets of E, the fault happens by (adversarially) selecting a subset $A \in \mathcal{S}$ and setting $\mathbf{w}[i] = 0$ for all $i \in A$. In other words, the adversary effectively "deletes" the edges in A. We denote $\mathbf{w}_A$ as the vector obtained by setting all coordinates of $\mathbf{w}$ in A being 0. Formally, for any given $\mathcal{S}$, the (adversarial) deletion fault-type is a function $\mathcal{U}_\mathcal{S}$ such that $\mathcal{U}_\mathcal{S}(\mathbf{w})$ is the class of all *singleton* distributions over $\{\mathbf{w}_A : A \in \mathcal{S}\}$. For notational convenience, we can simply write $\mathcal{U}_\mathcal{S} = \{\mathbf{w}_A : A \in \mathcal{S}\}$.

The following corollary follows directly from Theorem 2:

Corollary 1. *Let $\mathcal{H}_f$ and $\mathcal{U}_\mathcal{S}$ be defined as above, we have*

$$\mathcal{M}(\epsilon, \delta; \mathcal{H}_f, \mathcal{U}) \leq \tilde{O}\left(\frac{|\mathcal{S}||E| + \log(1/\delta)}{\epsilon^2}\right),$$

where $\tilde{O}$ hide poly-logarithmic factors on $|E|, \epsilon$.

Note that, if we allow the adversary to delete k (arbitrary) edges, then $|\mathcal{S}| = \binom{|E|}{k} \sim |E|^k$. Therefore, even for reasonably large k, the sample complexity provided in Corollary 1 can be exponentially large. Our main result of this section is the following theorem that establishes a sample complexity *independent* of $|\mathcal{S}|$.

Theorem 3. *Let $\mathcal{H}_f$ and $\mathcal{U}_\mathcal{S}$ be as defined above, we have the sample complexity $\mathcal{M}(\epsilon, \delta; \mathcal{H}_f, \mathcal{U}_\mathcal{S})$ upper bounded by:*

$$\tilde{O}\left(\frac{dLN^3 + L^2N^4 + d^2N^2 + \log(1/\delta)}{\epsilon^2}\right),$$

where $\tilde{O}$ hides poly-logarithmic factors on ϵ.

Note that, the hypothesis class we constructed in the proof of Lemma 2 can be instantiated by the neural network as well. Therefore, the hard fault-type constructed therein also applies to the hypothesis $\mathcal{H}_f$ in Theorem 3; i.e., the sample complexity must be scale linearly w.r.t. the fault-type size in the worst case. Perhaps surprisingly, Theorem 3 demonstrates that for certain *natural* fault-types, the sample complexity can be significantly improved.

Analysis for single neuron. Before we provide a formal proof of Theorem 3. We first consider the simpler case with a *single* neuron; i.e., the linear threshold function class $f(\mathbf{w}, \mathbf{x}) = 1\{\langle \mathbf{w}, \mathbf{x}\rangle \geq 0\}$. Let $\mathcal{U}_\mathcal{S}$ be an arbitrary deletion fault-type. We have by the proof of Theorem 2 that the generalization error (of *fault-tolerant* ERM rule) with sample size n is upper bounded by:

$$O(\mathsf{Rad}_n(\tilde{\ell} \circ \mathcal{G}) + \sqrt{\log(1/\delta)/n}),$$

where $\mathcal{G}$ and $\tilde{\ell}$ are as in Fact 2 and the discussion that follows. Note that by Massart's lemma (Shalev-Shwartz & Ben-David, 2014), we have for any given $\mathbf{x}^n$:

$$\mathsf{Rad}_n(\tilde{\ell} \circ \mathcal{G}) \leq O(\sqrt{\log|\mathcal{G}_{|\mathbf{x}^n}|/n}), \tag{9}$$

where $\mathcal{G}_{|\mathbf{x}^n}$ is the class of $\mathcal{G}$ restricted on $\mathbf{x}^n$. Therefore, it is sufficient to bound the size of $\mathcal{G}_{|\mathbf{x}^n}$.

By Sauer's lemma, the restriction of $\mathcal{H}_f$ on any $\mathbf{x}^n$ has size upper bounded by n^d. However, it is possible that for two hypotheses $h_{\mathbf{w}_1}$ and $h_{\mathbf{w}_2}$ with same restriction on $\mathbf{x}^n$, the restrictions of $g_{\mathbf{w}_1}$ and $g_{\mathbf{w}_2}$ are *different*. Therefore, to bound the size of $\mathcal{G}_{|\mathbf{x}^n}$, we will have to leverage the *structural* properties of $\mathcal{G}$.

Our key observation is that, by property of linear function, the deletion on $\mathbf{w}$ is *equivalent* to the deletion on $\mathbf{x}$! Specifically, we have:

$$\langle \mathbf{w}_A, \mathbf{x}\rangle = \langle \mathbf{w}, \mathbf{x}_A\rangle,$$

for any $A \subset \{1, \cdots, d\}$. Therefore, for any $\mathbf{w}$, the restriction of $\mathcal{H}_\mathbf{w} = \{h_{\mathbf{w}'} : \mathbf{w}' \in \mathcal{U}_\mathcal{S}(\mathbf{w})\} = \{h_{\mathbf{w}_A} : A \in \mathcal{S}\}$ on $\mathbf{x}^n$ can be *uniquely* recovered by the values of $h_\mathbf{w}$ on $\mathcal{L} \stackrel{\text{def}}{=} \{\mathbf{x}_A : \mathbf{x} \in \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}, A \in \mathcal{S}\}$. Since $|\mathcal{S}| \leq 2^d$, we have $|\mathcal{L}| \leq n2^d$. Invoking Sauer's lemma, we have the number of possible values of functions in $\mathcal{H}_f$ on $\mathcal{L}$ is upper bounded by $(n2^d)^d = n^d 2^{d^2}$. Note that any $g_\mathbf{w} \in \mathcal{G}$ is completely determined by $\mathcal{H}_\mathbf{w}$. This implies that $\mathcal{G}_{|\mathbf{x}^n} \leq n^d 2^{d^2}$, and therefore (9) is upper bounded by $O(\sqrt{d^2 \log n/n} + \sqrt{\log(1/\delta)/n})$. Rewriting n w.r.t. ϵ, δ, we arrive at the sample complexity:

$$O\left(\frac{d^2\log(1/\epsilon) + \log(1/\delta)}{\epsilon^2}\right).$$

This proves Theorem 3 for $L = N = 1$.

Proof of Theorem 3. We now provide the complete proof of Theorem 3. By the same argument as for single neuron, we know that the sample complexity is reduced to the upper bound of Rademacher complexity as in (9), which is further reduced to the upper bound on the size of $\mathcal{G}_{|\mathbf{x}^n}$. For any fixed $\mathbf{x}^n$, we have:

$$\mathcal{G}_{|\mathbf{x}^n} = \{(g_\mathbf{w}(\mathbf{x}_1), \cdots, g_\mathbf{w}(\mathbf{x}_n)) : \mathbf{w} \in \mathcal{W}\},$$

where $g_\mathbf{w}(\mathbf{x}) = (g^{\min}_\mathbf{w}(\mathbf{x}), g^{\max}_\mathbf{w}(\mathbf{x}))$ with $g^{\min}_\mathbf{w}(\mathbf{x}) = \inf_{A\in\mathcal{S}}\{h_{\mathbf{w}_A}(\mathbf{x})\}$ and $g^{\max}_\mathbf{w}(\mathbf{x}) = \sup_{A\in\mathcal{S}}\{h_{\mathbf{w}_A}(\mathbf{x})\}$. Note that, for any $\mathbf{w}$, $(g_\mathbf{w}(\mathbf{x}_1), \cdots, g_\mathbf{w}(\mathbf{x}_n))$ is completely determined by $\mathcal{V}_\mathbf{w} = \{(h_{\mathbf{w}_A}(\mathbf{x}_1), \cdots, h_{\mathbf{w}_A}(\mathbf{x}_n)) : A \in \mathcal{S}\}$. Therefore, we have:

$$|\mathcal{G}_{|\mathbf{x}^n}| \leq |\{\mathcal{V}_\mathbf{w} : \mathbf{w} \in \mathcal{W}\}|. \tag{10}$$

Now, our key idea is to *recursively* construct a sequence of *sets* $\mathcal{L}_1, \cdots, \mathcal{L}_L$ such that there exists an *injection* map $F : \{\mathcal{V}_\mathbf{w} : \mathbf{w} \in \mathcal{W}\} \to \mathcal{L}_L$, i.e., $|\{\mathcal{V}_\mathbf{w} : \mathbf{w} \in \mathcal{W}\}| \leq |\mathcal{L}_L|$.

To this send, we denote by $h^l_{\mathbf{w}}$ the transition function specified by neurons in layer $l \in [L]$ with weight $\mathbf{w}$ [5]. Note that for $l = 1$, we have $h^1_{\mathbf{w}}$ maps $\mathbb{R}^{dN} \to \{0,1\}^N$; for $2 \le l \le L-1$, $h^l_{\mathbf{w}}$ maps $\{0,1\}^{N^2} \to \{0,1\}^N$; and for $l = L$, $h^L_{\mathbf{w}}$ maps $\{0,1\}^N \to \{0,1\}$. Let $\mathcal{L}_0 = \{(\mathbf{x}^1_A, \cdots, \mathbf{x}^N_A) : \mathbf{x} \in \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}, A \in \mathcal{S}\}$, where $\mathbf{x}^i_A$ denotes for the deletion of $\mathbf{x}$ by applying the deletion type of A restricted on the ith neuron of the first layer. Denote $\mathcal{W}_\ell$ to be the weight space at layer $\ell \in [L]$. We *recursively* define

1. $\mathcal{L}_1 = \{(h^1_{\mathbf{w}}(\mathbf{v}))_{\mathbf{v}\in\mathcal{L}_0} : \mathbf{w} \in \mathcal{W}_1\}$;
2. Assume for $\ell \le L-1$, we have constructed the set $\mathcal{L}_l$, which satisfies the form that any element $\mathbf{v} \in \mathcal{L}_l$ is an *array* with each coordinate $\mathbf{v}[i] \in \{0,1\}^N$. For any $\mathbf{v} \in \mathcal{L}_l$, we define an expanded array $\mathbf{v}' = \{(\mathbf{v}[i]^1_A, \cdots, \mathbf{v}[i]^N_A) : i \in \mathsf{len}(\mathbf{v}), A \in \mathcal{S}\}$, where $\mathbf{v}[i]^j_A$ denotes the deletion of $\mathbf{v}[i]$ by applying the deletion type of A restricted on the jth neuron of the $l+1$st layer. Denote $\mathcal{L}'_l = \{\mathbf{v}' : \mathbf{v} \in \mathcal{L}_l\}$. We define:

$$\mathcal{L}_{l+1} = \left\{(h^{l+1}_{\mathbf{w}}(\mathbf{v}'[i]))_{i\in\mathsf{len}(\mathbf{v}')} : \mathbf{v}' \in \mathcal{L}'_l, \mathbf{w} \in \mathcal{W}_{\ell+1}\right\}. \tag{11}$$

We now observe the following fact about the $\mathcal{L}_l$s:

Fact 3. *We have $|\mathcal{L}_1| \le n^{dN} 2^{d^2N^2}$ and for any $\ell \le L-1$, we have $|\mathcal{L}_{l+1}| \le |\mathcal{L}_l| n^{N^2} 2^{N^3 d + \ell N^4}$.*

Proof. Note that $|\mathcal{L}_0| \le n2^{dN}$. We have $\mathcal{L}_1$ is essentially the restriction of all transition functions in the first layer on $\mathcal{L}_0$. Since the VC-dimension of linear threshold functions with input dimension d is upper bounded by d and there are N neurons in each layer, we have by Sauer's lemma that $|\mathcal{L}_1| \le (n2^{dN})^{dN} \le n^{dN} 2^{d^2N^2}$.

To prove the second upper bound, we denote L_l as the maximum length of arrays in $\mathcal{L}_l$. We have $L_1 \le n2^{dN}$ and $L_{l+1} \le L_l 2^{N^2}$. This follows by our construction that $\mathsf{len}(\mathbf{v}') \le \mathsf{len}(\mathbf{v}) 2^{N^2}$, since there can be at most 2^{N^2} deletion types on each layer. Therefore, we have $L_l \le n2^{dN} 2^{lN^2}$. Note that:

$$\mathcal{L}_{l+1} = \bigcup_{\mathbf{v}'\in\mathcal{L}'_l} \{(h^{l+1}_{\mathbf{w}}(\mathbf{v}'[i]))_{i\in\mathsf{len}(\mathbf{v}')} : \mathbf{w} \in \mathcal{W}_{\ell+1}\},$$

where each set in the union can be viewed as the restriction of all transition function of layer $l+1$ on $\mathbf{v}'$. We have, by Sauer's lemma again that:

$$|\mathcal{L}_{l+1}| \le |\mathcal{L}'_l| L^{N^2}_{l+1} \le |\mathcal{L}_l| n^{N^2} 2^{N^3 d + \ell N^4},$$

where we have used the fact that $|\mathcal{L}_l| = |\mathcal{L}'_l|$. □

[5] Here, we assume that each neuron has different inputs.

We now prove the main property of our construction:

Fact 4. *There exists a injection function $F : \{\mathcal{V}_{\mathbf{w}} : \mathbf{w} \in \mathcal{W}\} \to \mathcal{L}_L$, i.e., $|\{\mathcal{V}_{\mathbf{w}} : \mathbf{w} \in \mathcal{W}\}| \le |\mathcal{L}_L|$.*

Proof. For any weight $\mathbf{w}$, we show that the set $\mathcal{V}_{\mathbf{w}}$ can be reconstructed from an element in $\mathcal{L}_L$ using a *universal* recovering rule. To see this, we define $\mathbf{v}^l_{\mathbf{w}}$ to be the element in $\mathcal{L}_l$ that corresponds to $\mathbf{w}$ (note that $\mathbf{w}$ uniquely identifies an element in $\mathcal{L}_\ell$ by restriction the weight on the first ℓ layers, see construction in (11)). We claim that $\mathbf{v}^L_{\mathbf{w}} \in \mathcal{L}_L$ is the desired element that reconstructs $\mathcal{V}_{\mathbf{w}}$. Let $A \in \mathcal{S}$, we now describe a strategy that recovers $(h_{\mathbf{w}_A}(\mathbf{x}_1), \cdots, h_{\mathbf{w}_A}(\mathbf{x}_n))$. For any $\mathbf{x}_i$ and $l \le L-1$, we identify an index i_l in $\mathbf{v}^l_{\mathbf{w}}$. Initially, we set i_1 to be the index in $\mathcal{L}_0$ that corresponds to deletion type A and $\mathbf{x}_i$. Assume i_l has been constructed. Since $\mathbf{v}^{l+1}_{\mathbf{w}}$ is constructed from $\mathbf{v}^l_{\mathbf{w}}$ by expanding on deletion types in $\mathcal{S}$, we define i_{l+1} to be the index in $\mathbf{v}^{l+1}_{\mathbf{w}}$ that corresponds to $\mathbf{v}^l_{\mathbf{w}}[i_l]$ and deletion type A (see the definition of $\mathbf{v}'$ in our construction). It is easy to verify that $\mathbf{v}^L_{\mathbf{w}}[i_L]$ is exactly the same as $h_{\mathbf{w}_A}(\mathbf{x}_i)$. Note that our recovering rule depends only on the deletion type A and is *independent* of $\mathbf{w}$. Therefore, any two distinct $\mathcal{V}_{\mathbf{w}_1}$ and $\mathcal{V}_{\mathbf{w}_2}$ cannot be recovered from the same element in $\mathcal{L}_L$. The function defined by mapping $\mathcal{V}_{\mathbf{w}}$ to any elements in $\mathcal{L}_L$ that recovers it is the desired injection map. □

Now, by Fact 3, we have

$$|\mathcal{L}_L| \le n^{dN + LN^2} 2^{dLN^3 + L^2N^4 + d^2N^2}.$$

Therefore, by Fact 4, (9) and (10), the sample complexity is upper bounded by

$$\tilde{O}\left(\frac{dLN^3 + L^2N^4 + d^2N^2 + \log(1/\delta)}{\epsilon^2}\right),$$

where $\tilde{O}$ hides poly-logarithmic factor on ϵ. This completes the proof of Theorem 3. □

4. Conclusion and Discussion

In this paper, we introduce a novel theoretical framework for analyzing fault tolerance in machine learning models, offering tight characterizations of fault-tolerant PAC learnability for both random and adversarial faults. We show that while random faults do not increase learning complexity, adversarial faults scale linearly with the fault space size. Specifically, for neural networks with deletion faults, we demonstrate that sample complexity can be, surprisingly, independent of the number of deletions, scaling instead with network parameters, indicating that certain natural fault-types can be well-tolerated. Our framework paves the way for significant future research, including extending analysis to other activation functions like ReLU in neural networks and exploring additional fault types, such as neuron activation faults or

precision errors. This work lays a rigorous foundation for developing more reliable and trustworthy machine learning models for diverse applications.

Computational aspects. While our work primarily focuses on the statistical sample complexity, it would also be interesting to investigate computationally efficient methods for computing our *fault-tolerant* ERM rule as in (6). A heuristic approach would work as follows: (i) we first estimate the *faulty-loss* by making (sampling) oracle calls to the fault-type $\mathcal{U}$, which implements a *value-oracle* for the faulty-loss; (ii) the value-oracle will then be implemented via, e.g., the *discrete differences*, to compute the *gradients* of faulty-loss w.r.t. $\mathbf{w}$; (iii) a standard SGD will then be used to compute the *fault-tolerant* ERM. We believe it would be an interesting future direction to investigate how the structures of the class $\mathcal{G}$ (see Proposition 1) impact the convergence rate of such an SGD algorithm.

Other variations. Besides the random and adversarial faulty scenarios we investigate in this paper, there can be many other formulations under the concept of *fault-tolerant* PAC learning. For instance, one may assume that the adversary alters the function only at the beginning, instead of altering it with each sample, as in our case. In such a scenario, the sample complexity would be controlled by that of $\bigcup_{h\in\mathcal{H}} \mathcal{U}(h)$ via a standard uniform convergence argument. We believe investigating other types of formulations would also be interesting for future research.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful suggestions. This work was partially supported by the NSF Center for Science of Information (CSoI) Grant CCF-0939370, and also by NSF Grants CCF-2006440, CCF-2007238, and CCF-2211423.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abraham, J. A. and Fuchs, W. K. Fault and error models for vlsi. *Proceedings of the IEEE*, 74(5):639–654, 1986.

Al Shahrani, A. M., Alomar, M. A., Alqahtani, K. N., Basingab, M. S., Sharma, B., and Rizwan, A. Machine learning-enabled smart industrial automation systems using internet of things. *Sensors*, 23(1):324, 2022.

Arad, B. S. and El-Amawy, A. On fault tolerant training of feedforward neural networks. *Neural Networks*, 10(3): 539–553, 1997.

Bartlett, P. L. and Maass, W. Vapnik-chervonenkis dimension of neural nets. *The handbook of brain theory and neural networks*, pp. 1188–1192, 2003.

Chin, C.-T., Mehrotra, K., Mohan, C. K., and Rankat, S. Training techniques to obtain fault-tolerant neural networks. In *Proceedings of IEEE 24th international symposium on fault-tolerant computing*, pp. 360–369. IEEE, 1994.

Chu, L.-C. and Wah, B. W. Fault tolerant neural networks with hybrid redundancy. In *1990 IJCNN international joint conference on neural networks*, pp. 639–649. IEEE, 1990.

Deng, J., Fang, Y., Du, Z., Wang, Y., Li, H., Temam, O., Ienne, P., Novo, D., Li, X., Chen, Y., et al. Retraining-based timing error mitigation for hardware neural networks. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 593–596. IEEE, 2015.

Deodhare, D., Vidyasagar, M., and Keethi, S. S. Synthesis of fault-tolerant feedforward neural networks using minimax optimization. *IEEE Transactions on Neural Networks*, 9 (5):891–900, 1998.

Dodd, P. E. and Massengill, L. W. Basic mechanisms and modeling of single-event upset in digital microelectronics. *IEEE Transactions on nuclear Science*, 50(3):583–602, 2003.

Edwards, P. J. and Murray, A. F. Penalty terms for fault tolerance. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 2, pp. 943–947. IEEE, 1997.

Eghbal, A., Yaghini, P. M., Bagherzadeh, N., and Khayambashi, M. Analytical fault tolerance assessment and metrics for tsv-based 3d network-on-chip. *IEEE Transactions on computers*, 64(12):3591–3604, 2015.

Emmerson, M. D. and Damper, R. I. Determining and improving the fault tolerance of multilayer perceptrons in a pattern-recognition application. *IEEE transactions on neural networks*, 4(5):788–793, 1993.

Hashmi, A., Berry, H., Temam, O., and Lipasti, M. Automatic abstraction and fault tolerance in cortical microachitectures. *ACM SIGARCH computer architecture news*, 39(3):1–10, 2011.

Hengstler, M., Enkel, E., and Duelli, S. Applied artificial intelligence and trust—the case of autonomous vehicles and

medical assistance devices. *Technological Forecasting and Social Change*, 105:105–120, 2016.

Izzo, D., Märtens, M., and Pan, B. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics*, 3:287–299, 2019.

Kershaw, J., Yu, R., Zhang, Y., and Wang, P. Hybrid machine learning-enabled adaptive welding speed control. *Journal of Manufacturing Processes*, 71:374–383, 2021.

Khan, F. Safety and integrity management of operations in harsh environments, 2020.

Khunasaraphan, C., Vanapipat, K., and Lursinsap, C. Weight shifting techniques for self-recovery neural networks. *IEEE transactions on neural networks*, 5(4):651–658, 1994.

Liu, Y., Wei, L., Luo, B., and Xu, Q. Fault injection attack on deep neural network. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 131–138. IEEE, 2017.

Lu, Y. *Adaptive Intelligent Systems for Extreme Environments*. PhD thesis, University of Essex, 2023.

Ma, Y., Wang, Z., Yang, H., and Yang, L. Artificial intelligence applications in the development of autonomous vehicles: A survey. *IEEE/CAA Journal of Automatica Sinica*, 7(2):315–329, 2020.

Montasser, O., Hanneke, S., and Srebro, N. Vc classes are adversarially robustly learnable, but only improperly. In Beygelzimer, A. and Hsu, D. (eds.), *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 2512–2530. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/montasser19a.html.

Nguyen, T.-H., Imran, M., Choi, J., and Yang, J.-S. Craft: Criticality-aware fault-tolerance enhancement techniques for emerging memories-based deep neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

Phatak, D. S. and Koren, I. Complete and partial fault tolerance of feedforward neural nets. *IEEE transactions on neural networks*, 6(2):446–456, 1995.

Piuri, V. Analysis of fault tolerance in artificial neural networks. *Journal of Parallel and Distributed Computing*, 61(1):18–48, 2001.

Protzel, P., Palumbo, D., and Arras, M. Performance and fault-tolerance of neural networks for optimization. *IEEE Transactions on Neural Networks*, 4(4):600–614, 1993. doi: 10.1109/72.238315.

Rakin, A. S., He, Z., and Fan, D. Bit-flip attack: Crushing neural network with progressive bit search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1211–1220, 2019.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Sosnowski, J. Transient fault tolerance in digital systems. *IEEE Micro*, 14(1):24–35, 1994.

Syed, R. T., Ulbricht, M., Piotrowski, K., and Krstic, M. A survey on fault-tolerant methodologies for deep neural networks. *Pomiary Automatyka Robotyka*, 27(2):89–98, 2023.

Tipaldi, M., Feruglio, L., Denis, P., and D'Angelo, G. On applying ai-driven flight data analysis for operational spacecraft model-based diagnostics. *Annual Reviews in Control*, 49:197–211, 2020.

Torres-Huitzil, C. and Girau, B. Fault and error tolerance in neural networks: A review. *IEEE Access*, 5:17322–17341, 2017. doi: 10.1109/ACCESS.2017.2742698.

Verma, S., Kaur, S., Garg, S., Sharma, A. K., and Alrashoud, M. Agric: Artificial intelligence-based green routing for industrial cyber-physical system pertaining to extreme environment. *IEEE Internet of Things Journal*, 2023.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wei, N., Yang, S., and Tong, S. A modified learning algorithm for improving the fault tolerance of bp networks. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pp. 247–252. IEEE, 1996.

Zhou, Z.-H. and Chen, S.-F. Evolving fault-tolerant neural networks. *Neural Computing & Applications*, 11:156–160, 2003.