DETERMINISTIC PAC-BAYESIAN GENERALIZATION BOUNDS FOR DEEP NETWORKS VIA GENERALIZING NOISE-RESILIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

The ability of overparameterized deep networks to generalize well has been linked to the fact that stochastic gradient descent (SGD) finds solutions that lie in flat, wide minima in the training loss – minima where the output of the network is resilient to small random noise added to its parameters. So far this observation has been used to provide generalization guarantees only for neural networks whose parameters are either *stochastic* or *compressed*. In this work, we present a general PAC-Bayesian framework that leverages this observation to provide a bound on the original network learned – a network that is deterministic and uncompressed. What enables us to do this is a key novelty in our approach: our framework allows us to show that if on training data, the interactions between the weight matrices satisfy certain conditions that imply a wide training loss minimum, these conditions themselves *generalize* to the interactions between the matrices on test data, thereby implying a wide test loss minimum. We then apply our general framework in a setup where we assume that the pre-activation values of the network are not too small (although we assume this only on the training data). In this setup, we provide a generalization guarantee for the original (deterministic, uncompressed) network, that does not scale with product of the spectral norms of the weight matrices – a guarantee that would not have been possible with prior approaches.

1 INTRODUCTION

Modern deep neural networks contain millions of parameters and are trained on relatively few samples. Conventional wisdom in machine learning suggests that such models should massively overfit on the training data, as these models have the capacity to memorize even a randomly labeled dataset of similar size (Zhang et al., 2017). Yet these models have achieved state-of-the-art generalization error on many real-world tasks. This observation has spurred an active line of research (Neyshabur et al., 2015; Soudry et al., 2018; Brutzkus et al., 2018) that has tried to understand what properties are possessed by stochastic gradient descent (SGD) training of deep networks that allows these networks to generalize well.

One particularly promising line of work (Neyshabur et al., 2017; Arora et al., 2018) has been bounds that utilize the *noise-resilience* of deep networks on training data (i.e., how much the training loss of the network changes with noise injected into the parameters, or roughly, how wide is the training loss minimum). While these have yielded generalization bounds that do not have a severe exponential dependence on depth (unlike other bounds that grow with the product of spectral norms of the weight matrices), these bounds are quite limited: they either apply to a *stochastic* version of the classifier (where the parameters are drawn from a distribution) or a *compressed* version of the classifier (where the parameters are modified and represented using fewer bits).

In this paper, we revisit the PAC-Bayesian analysis of deep networks in Neyshabur et al. (2017; 2018). We provide a general framework that allows one to use noise-resilience of the deep network on training data to provide a bound on the original deterministic and uncompressed network. We achieve this by arguing that if on the training data, the interaction between the 'activated weight matrices' (weight matrices where the weights incoming from/outgoing to inactive units are zeroed out)

satisfy certain conditions which results in a wide training loss minimum, these conditions themselves generalize to the weight matrix interactions on the test data.

After presenting this general PAC-Bayesian framework, we specialize it to the case of deep ReLU networks, showing that we can provide a generalization bound that accomplishes two goals simultaneously: i) it applies to the original network and ii) it does not scale exponentially with depth in terms of the products of the spectral norms of the weight matrices; instead our bound scales with more meaningful terms that capture the interactions between the weight matrices and do not have such a severe dependence on depth in practice. We note that all but one of these terms are indeed quite small on networks in practice. However, one particularly (empirically) large term that we use is the reciprocal of the magnitude of the network pre-activations on the training data (and so our bound would be small only in the scenario where the pre-activations are not too small). We emphasize that this drawback is more of a limitation in how we characterize noise-resilience through the specific conditions we chose for the ReLU network, rather than a drawback in our PAC-Bayesian framework itself. Our hope is that, since our technique is quite general and flexible, by carefully identifying the right set of conditions, in the future, one might be able to derive a similar generalization guarantee that is smaller in practice.

To the best of our knowledge, our approach of *generalizing* noise-resilience of deep networks from training data to test data in order to derive a bound on the *original* network that does not scale with products of spectral norms, has neither been considered nor accomplished so far, even in limited situations.

2 Background and related work

One of the most important aspects of the generalization puzzle that has been studied is that of the flatness/width of the training loss at the minimum found by SGD. The general understanding is that flatter minima are correlated with better generalization behavior, and this should somehow help explain the generalization behavior (Hochreiter & Schmidhuber, 1997; Hinton & van Camp, 1993; Keskar et al., 2017). Flatness of the training loss minimum is also correlated with the observation that on training data, adding noise to the parameters of the network results only in little change in the output of the network – or in other words, the network is noise-resilient.

Most generalization bounds that do not make use of noise-resilience of the network have generalization guarantees that in practice have strong exponential dependence on the depth of the network. More precisely, these bounds scale either with the product of the spectral norms of the weight matrices (Neyshabur et al., 2018; Bartlett et al., 2017) or their Frobenius norms (Golowich et al., 2018). In practice, the weight matrices have a spectral norm that is as large as 2 or 3, and an even larger Frobenius norm that scales with $\sqrt{H}$ where H is the width of the network i.e., maximum number of hidden units per layer. [1] Thus, the generalization bound scales as say, 2^D or $H^{D/2}$, where D is the depth of the network.

At a high level, the reason these bounds suffer from such an exponential dependence on depth is that they effectively perform a worst case approximation of how the weight matrices interact with each other. For example, the product of the spectral norms arises from a naive approximation of the Lipschitz constant of the neural network, which would hold only when the singular values of the weight matrices all align with each other. However, in practice, for most inputs to the network, the interactions between the activated weight matrices are not as adverse.

By using noise-resilience of the networks, prior approaches (Arora et al., 2018; Neyshabur et al., 2017) have been able to derive bounds that replace the above worst-case approximation with smaller terms that realistically capture these interactions. However, these works are limited in critical ways. Arora et al. (2018) use noise-resilience of the network to *modify* and "compress" the parameter representation of the network, and derive a generalization bound on the compressed network. While this bound enjoys a better dependence on depth because its applies to a compressed network, the

[1] To understand why these values are of this order in magnitude, consider the initial matrix that is randomly initialized with independent entries with variance $1/\sqrt{H}$. It can be shown that the spectral norm of this matrix, with high probability, lies near its expected value, near 2 and the Frobenius norm near its expected value which is $\sqrt{H}$. Since SGD is observed not to move too far away from the initialization regardless of H (Nagarajan & Kolter, 2017), these values are more or less preserved for the final weight matrices.

main drawback of this bound is that it does not apply on the original network. On the other hand, Neyshabur et al. (2017) take advantage of noise-resilience on training data by incorporating it within a PAC-Bayesian generalization bound (McAllester, 1999a). However, their final guarantee is only a bound on the expected test loss of a stochastic network.

In this work, we revisit the idea in Neyshabur et al. (2017), by pursuing the PAC-Bayesian framework (McAllester, 1999a) to answer this question. The standard PAC-Bayesian framework provides generalization bounds for the expected loss of a stochastic classifier, where the stochasticity typically corresponds to Gaussian noise injected into the parameters output by the learning algorithm. However, if the classifier is noise-resilient on both training and test data, one could extend the PAC-Bayesian bound to a standard generalization guarantee on the deterministic classifier.

Other works have used PAC-Bayesian bounds in different ways in the context of neural networks. Langford & Caruana (2001); Dziugaite & Roy (2017) optimize the stochasticity and the weights of the network in order to *numerically* compute good (i.e., non-vacuous) generalization bounds on the stochastic network. Neyshabur et al. (2018) derive generalization bounds on the original, deterministic network by working from the PAC-Bayesian bound on the stochastic network. However, as stated earlier, their work does not make use of noise resilience in the networks learned by SGD.

OUR CONTRIBUTIONS The key contribution in our work is a general PAC-Bayesian framework for deriving generalization bounds while leveraging the noise resilience of a deep network. While our approach is applied to deep networks, we note that it is quite general (as will be apparent from Section 3) and can be applied to other classifiers.

In our framework, we consider a set of conditions that when satisfied by the network, makes the output of the network noise-resilient at a particular input datapoint. For example, these conditions could characterize the interactions between the activated weight matrices at a particular input. To provide a generalization guarantee, we assume that the learning algorithm has found weights such that these conditions hold for the weight interactions in the network *on training data* (which effectively implies a wide training loss minimum). Then, as a key step, we *generalize* these conditions over to the weight interactions on test data (which effectively implies a wide test loss minimum) [2]. Thus, with the guarantee that the classifier is noise-resilient both on training and test data, we derive a generalization bound on the test loss of the original network.

Our framework relies on a novel technique to convert the PAC-Bayesian bound on a stochastic classifier to the generalization bound on the deterministic classifier. This technique is arguably more powerful than known PAC-Bayesian approaches, in leveraging noise-resilience, and is thus of independent theoretical interest.

Finally, we apply our framework to a specific set up of ReLU based feedforward networks. In particular, we first instantiate the above abstract framework with a set of specific conditions, and then use the above framework to derive a bound on the original network. While very similar conditions have already been identified in prior work (Arora et al., 2018; Neyshabur et al., 2017), our contribution here is in showing how these conditions generalize from training to test data. Crucially, like these works, our bound does not have severe exponential dependence on depth in terms of products of spectral norms.

We note that in reality, all but one of our conditions on the network do hold on training data as necessitated by the framework. The strong, non-realistic condition we make is that the pre-activation values of the network are sufficiently large, although only on training data; however, in practice a small proportion of the pre-activation values can be arbitrarily small. Our generalization bound scales inversely with the smallest absolute value of the pre-activations on the training data, and hence in practice, our bound would be large.

It is worth noting that Arora et al. (2018); Neyshabur et al. (2017) too require similar assumptions about pre-activation values; their assumptions are more realistic in that they effectively allow for a small proportion of units to have small pre-activation values. However, even under our stronger condition that no such units exist, it is not apparent how these approaches would yield a similar bound on the deterministic, uncompressed network without generalizing their conditions to test data.

[2] Note that we can not directly assume these conditions to hold on test data, as that would be 'cheating' from the perspective of a generalization guarantee.

We hope that in the future our work could be developed further to accommodate the more realistic conditions from Arora et al. (2018); Neyshabur et al. (2017).

3 A GENERAL PAC-BAYESIAN FRAMEWORK

In this section, we present our general PAC-Bayesian framework that uses noise-resilience of the network to convert a PAC-Bayesian generalization bound on the stochastic classifier to a generalization bound on the deterministic classifier.

NOTATION. Let $KL(\cdot\|\cdot)$ denote the KL-divergence, $\|\cdot\|$ denote the ℓ_2 norm of a vector, $\|\cdot\|_\infty$ the ℓ_∞ norm of a vector, and $\|\cdot\|_2, \|\cdot\|_F$ denote the spectral and Frobenius norm of a matrix, respectively. Consider a K-class learning task where the labeled datapoints $(\mathbf{x}, y)$ are drawn from an underlying distribution $\mathcal{D}$ over $\mathcal{X} \times \{1, 2, \cdots, K\}$ where $\mathcal{X} \in \mathbb{R}^N$ and $\forall \mathbf{x} \in \mathcal{X}$, $\|\mathbf{x}\| \leq B$. We consider a neural network of D layers (we will mostly care about $D > 2$) mapping from $\mathbb{R}^N \to \mathbb{R}^K$. The layers are fully connected with H units in each hidden layer, and with ReLU activations $\phi(\cdot)$ on all the hidden units and linear activations on the output units. The network is parameterized by $\mathcal{W} = (W_1, W_2, \cdots, W_D)$ such that the output of the network is computed as $f(\mathbf{x}; \mathcal{W}) = W_D \phi(W_{D-1} \cdots \phi(W_1 \mathbf{x}))$ and the output for the kth class as $f(\mathbf{x}; \mathcal{W})[k]$. We denote the value of the hth hidden unit on the dth layer before and after the activation by $g^d(\mathbf{x}; \mathcal{W})[h]$ and $f^d(\mathbf{x}; \mathcal{W})[h]$ respectively. We define $J^{d/d'}(\mathbf{x}; \mathcal{W}) := \partial g^d(\mathbf{x}; \mathcal{W}) / \partial g^{d'}(\mathbf{x}; \mathcal{W})$ to be the Jacobian of the pre-activations of layer d with respect to the pre-activations of layer d' for $d' \leq d$ (each row in this Jacobian corresponds to a unit in layer d). In short, we will call this, Jacobian d/d'. We will denote the parameters of a random initialization of the network by $\mathcal{Z} = (Z_1, Z_2, \cdots, Z_D)$. In our PAC-Bayesian analysis, we will use $\mathcal{U} = (U_1, U_2, \cdots, U_D) \sim \mathcal{N}(0, \sigma^2)$ to denote parameters whose entries are sampled independently from a Gaussian, and $\mathcal{W} + \mathcal{U}$ to denote the entrywise addition of the two sets of parameters. We use $\|\mathcal{W}\|_F^2$ to denote $\sum_{d=1}^{D} \|W_d\|_F^2$. Finally, given a training set S of m samples, we let $(\mathbf{x}, y) \sim S$ to denote uniform sampling from the set. See Appendix A for more notations.

TRADITIONAL PAC-BAYESIAN BOUNDS. The PAC-Bayesian framework (McAllester, 1999a;b) allows us to derive generalization bounds for a *stochastic* classifier. Specifically, let $\tilde{\mathcal{W}}$ be a random variable in the parameter space whose distribution is learned based on training data S. Let P be a prior distribution in the parameter space chosen independent of the training data. Let $\mathcal{L}(\tilde{\mathcal{W}}, \mathbf{x}, y)$ be the loss of the classifier $\tilde{\mathcal{W}}$ on a datapoint $(\mathbf{x}, y)$, where the loss between 0 and 1; in this paper, we take the loss to be the 0-1 classification error. The PAC-Bayesian framework yields the following generalization bound on the stochastic classifier that holds with probability $1 - \delta$ over the draw of the training set S of m samples:

$$\mathbb{E}_{\tilde{\mathcal{W}}}[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}(\tilde{\mathcal{W}}, \mathbf{x}, y)]] \leq \mathbb{E}_{\tilde{\mathcal{W}}}[\mathbb{E}_{(\mathbf{x},y)\sim S}[\mathcal{L}(\tilde{\mathcal{W}}, \mathbf{x}, y)]] + 2\sqrt{2(KL(\tilde{\mathcal{W}}\|P) + \ln m/\delta)/(m-1)}$$

Typically, and in the rest of this discussion, $\tilde{\mathcal{W}}$ is a Gaussian with covariance $\sigma^2 I$ for some $\sigma > 0$ centered at the weights $\mathcal{W}$ learned based on the training data. Furthermore, we will set P to be a Gaussian with covariance $\sigma^2 I$ centered at the random initialization of the network like in Dziugaite & Roy (2017), instead of at the origin, like in Neyshabur et al. (2018). This is because the resulting KL-divergence – which depends on the distance between the means of the prior and the posterior – is known to be smaller, and to save a $\sqrt{H}$ factor in the bound (Nagarajan & Kolter, 2017).

3.1 OUR FRAMEWORK

To extend the above PAC-Bayesian bound to a standard generalization bound on a deterministic classifier $\mathcal{W}$, we need to replace the training and the test loss of the stochastic classifier with that of the original, deterministic classifier. However, in doing so, we will have to introduce extra terms in the upper bound to account for the perturbation suffered by the train and test loss under the Gaussian perturbation of the parameters. To tightly bound these two terms, we need that the network is noise-resilient on training and test data respectively. Our hope is that if the learning algorithm has found weights such that the network is noise-resilient on the training data, we can then generalize this noise-resilience over to test data as well, allowing us to better bound the excess terms.

Below, we lay out the key components of our framework, and constraints on how these should be designed; following this we present our main result and some intuition about our proof technique.

INPUT-DEPENDENT PROPERTIES OF WEIGHTS In our framework, we characterize noise-resilience through a set of say, R (abstract) conditions on how the activated weight matrices interact with each other for a given input. To define these noise-resilience-related conditions, we will consider R sets of the form $\{\rho_r^{(1)}(\mathcal{W}, \mathbf{x}, y), \rho_r^{(2)}(\mathcal{W}, \mathbf{x}, y), \cdots,\}$ for $r = 1, 2, 3, \cdots, R$; here $\rho_r^{(l)}(\mathcal{W}, \mathbf{x}, y)$ computes a scalar value which corresponds to an *input-dependent property* of the weights (and l is an index for the elements within a set). The reader can think of each set here as the set of properties of a particular layer in the network (the reason for grouping them in this manner will be apparent soon). For example, in the next section, for ReLU networks, we will consider D sets, each consisting of the pre-activations of each layer $\{g^d(\mathbf{x}; \mathcal{W})[1], \cdots, g^d(\mathbf{x}; \mathcal{W})[H]\}$ for $d = 1, 2, \cdots D$. (Although, as we will see, we will actually consider even more sets of properties amounting to $R = \mathcal{O}(D^2)$ sets in total.)

Next, we define conditions that bound the properties. To do this, for each set, we define a function $T_r(\mathcal{W}, \mathbf{x}, y)$ that aggregates the values of the properties in that set; then, a *condition* of the form $T_r(\mathcal{W}, \mathbf{x}, y) > 0$ would imply a bound on the properties in the set. For example, on ReLU networks, we will define $T_d(\mathcal{W}, \mathbf{x}, y) := \min_{h=1}^{H} |g^d(\mathbf{x}; \mathcal{W})[h]| - \Delta$ (for some $\Delta > 0$) which aggregates the pre-activation values of the dth layer; the condition $T_d(\mathcal{W}, \mathbf{x}, y) > 0$ imposed on all the training data, would imply that these pre-activation values on the training set have magnitude at least Δ.

For convenience, we instantiate the Rth set of properties to be the set of K classifier outputs $\{f^D(\mathbf{x}; \mathcal{W})[1], \cdots, f^D(\mathbf{x}; \mathcal{W})[K]\}$ and define $T_R(\mathcal{W}, \mathbf{x}, y) := (f(\mathbf{x}; \mathcal{W})[y] - \max_{j \neq y} f(\mathbf{x}; \mathcal{W})[j])/2$. Note that the sign of $T_R(\mathcal{W}, \mathbf{x}, y)$ corresponds to the 0-1 error.

CONSTRAINT 1: AGGREGATION OF THE PROPERTIES In this framework, we impose the following constraint on how these aggregator functions are designed:

$$|T_r(\mathcal{W}, \mathbf{x}, y) - T_r(\mathcal{W}', \mathbf{x}, y)| \leq \max_l |\rho_r^{(l)}(\mathcal{W}, \mathbf{x}, y) - \rho_r^{(l)}(\mathcal{W}', \mathbf{x}, y)| \tag{1}$$

This limits us to functions that are linear in terms of the extreme values/absolute values from the corresponding set of properties, like the example function $T_d(\mathcal{W}, \mathbf{x}, y) = \min_{h=1}^{H} |g^d(\mathbf{x}; \mathcal{W})[h]| - \Delta$.

CONSTRAINT 2: ORDERING OF THE SETS OF PROPERTIES Next, we impose a crucial constraint on how the properties are grouped and indexed, that will help us show that these properties themselves are noise-resilient. Roughly speaking, we want that for a given input, *the properties in the r-th set are noise-resilient, whenever the first $r-1$ sets of properties satisfy the bounds imposed on them through the conditions $T_q(\mathcal{W}, \mathbf{x}, y) > 0$ for $q = 1, 2, \cdots r-1$.* For example, in the case of ReLU networks, we will show that the perturbation in the pre-activation values of the dth layer i.e., $\max_h |g^d(\mathbf{x}; \mathcal{W})[h] - g^d(\mathbf{x}; \mathcal{W} + \mathcal{U})[h]|$, is small (i.e., doesn't scale with the product of the spectral norms of the weights) w.h.p over $\mathcal{U} \sim \mathcal{N}(0, \sigma^2 I)$ as long as the absolute pre-activation values in the layers below $d-1$ are large (and a few other norm-bounds on the lower layer weights are satisfied).

We formalize the above requirement by defining a set of expressions $\Delta_1(\sigma), \cdots, \Delta_R(\sigma)$ that bound the perturbation in the properties as follows. For any $r \leq R$, let $\mathbf{x} \in \mathcal{X}$ be such that $T_q(\mathcal{W}, \mathbf{x}, y) > 0$ for all the preceding $q < r$. Then, the following should hold:

$$\begin{aligned} Pr_{\mathcal{U} \sim \mathcal{N}(0, \sigma^2 I)} \Big[\max_l \ |\rho_r^{(l)}(\mathcal{W} + \mathcal{U}, \mathbf{x}, y) - \rho_r^{(l)}(\mathcal{W}, \mathbf{x}, y)| > \frac{\Delta_r(\sigma)}{2} \ \wedge \\ \forall q < r, \max_l \ |\rho_q^{(l)}(\mathcal{W} + \mathcal{U}, \mathbf{x}, y) - \rho_q^{(l)}(\mathcal{W}, \mathbf{x}, y)| < \frac{\Delta_q(\sigma)}{2} \Big] \leq \frac{1}{R\sqrt{m}}. \end{aligned} \tag{2}$$

In other words, when the first $r-1$ conditions $T_q(\mathcal{W}, \mathbf{x}, y) > 0$ are satisfied at an $\mathbf{x}$, the event that i) the properties involved in the rth condition suffers a large perturbation and that ii) the properties involved in the first $r-1$ functions do *not* suffer much perturbation – which implies that the first $r-1$ conditions are still approximately satisfied under perturbation – together hold, must have a small probability. (When we instantiate the framework, we have to derive closed form expressions for $\Delta_r(\sigma)$ satisfying the above inequality, like we will do in Lemma E.1 for ReLU networks.)

THEOREM STATEMENT In this setup, we have the following 'margin-based' generalization guarantee on the original network. That is, we bound the 0-1 test error of the network by a margin-based error on the training data. Our generalization guarantee – which scales linearly with the number of conditions R – holds under the setting that the training algorithm always finds weights such that on the training data, the condition that $T_r(\mathcal{W}, \mathbf{x}, y)$ is sufficiently large is satisfied for all $r = 1, \cdots, R-1$.

Theorem 3.1. *Let $\gamma_{class} > 0$. Let $\Delta_1^\star, \Delta_2^\star, \cdots, \Delta_R^\star$ be a set of positive constants, where $\Delta_R^\star = \gamma_{class}/2$. Let σ^* be the maximum value of σ such that for all $r = 1, \cdots, R$, $\Delta_r(\sigma) \leq \Delta_r^\star$. Then, for any $\delta > 0$, with probability $1-\delta$ over the draw of samples S from $\mathcal{D}^m$, for any $\mathcal{W}$ we have that, if $\mathcal{W}$ satisfies $T_r(\mathcal{W}, \mathbf{x}, y) > \Delta_r^\star$ for all $r < R$ and for all $(\mathbf{x}, y) \in S$, then*[3]

$$Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[f(\mathbf{x};\mathcal{W})[y] < \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j]\right] \leq Pr_{(\mathbf{x},y)\sim S}\left[f(\mathbf{x};\mathcal{W})[y] < \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j] + \gamma_{class}\right] + 2\tilde{\mathcal{O}}\left(R\sqrt{\frac{2KL(\mathcal{N}(\mathcal{W},(\sigma^\star)^2 I)\|P)}{m-1}}\right)$$

The crux of our proof (in Appendix D) lies in generalizing the conditions $T_r(\mathcal{W}, \mathbf{x}, y) > \Delta_r^\star$ satisfied on the training data to test data *one after the other*. Crucially, after we generalize the first $r-1$ conditions from training data to test data, we will have from Equation 2 that the rth set of properties are noise-resilient on both training and test data. Given the noise-resilience on the rth set of properties on test data, we can generalize the rth condition to test data too.

We emphasize a key, fundamental tool that we develop for use in this theorem. Specifically, in Theorem C.1 we present a new technique to convert a generic PAC-Bayesian bound on a stochastic classifier, to a generalization bound on the deterministic classifier. We argue in Section C.1 this technique is more powerful than prior approaches (Neyshabur et al., 2018; Langford & Shawe-Taylor, 2002; McAllester, 2003) in leveraging the noise-resilience of a generic classifier.

The high level argument is that, to convert the PAC-Bayesian bound, previous works relied on a looser output perturbation bound, one that holds on all possible inputs, with high probability over all perturbations i.e., a bound on $\max_{\mathbf{x}} \|f(\mathbf{x};\mathcal{W}) - f(\mathbf{x};\mathcal{W}+\mathcal{U})\|_\infty$ w.h.p over draws of $\mathcal{U}$. In contrast, our technique relies on a subtly different but significantly tighter bound; specifically, we rely on a bound on the output perturbation that holds with high probability *given an input* i.e., a bound on $\|f(\mathbf{x};\mathcal{W}) - f(\mathbf{x};\mathcal{W}+\mathcal{U})\|_\infty$ w.h.p over draws of $\mathcal{U}$ for each $\mathbf{x}$. When we do instantiate our framework as in the next section, this subtle difference is critical in being able to bound the output perturbation without suffering from a factor proportional to the product of the spectral norms of the weight matrices (which is the case in Neyshabur et al. (2018)).

4 APPLICATION OF OUR FRAMEWORK TO RELU NETWORKS

In this section, we apply our framework to feedforward ReLU networks and derive a generalization bound on the original network that does not scale with the product of spectral norms of the weight matrices. Informally, we consider a setting where the learning algorithm satisfies the following conditions on the training data that make it noise-resilient on training data: a) the Jacobian of any layer with respect to a lower layer, has rows with a small ℓ_2 norm, b) the ℓ_2 norm of the hidden layers are all small and c) the pre-activation values are all sufficiently large in magnitude. We cast these conditions in terms of the functions T_r in the general framework. These properties are quite similar to those already explored in Arora et al. (2018); Neyshabur et al. (2017); we provide more intuition about these properties, and how we cast them in our framework in Appendix E.1.

We prove in Lemma E.1 in Appendix E a guarantee equivalent to the abstract inequality in Equation 2. Essentially, we show that under random perturbations of the parameters, the perturbation in the output of the network and the perturbation in the input-dependent properties involved in (a), (b), (c) themselves can all be bounded in terms of each other. The highlight of these perturbation bounds is that none of them grow with the spectral norms of the network.

Having set up the conditions of the framework as required, we then instantiate the bound provided by the framework as described below. Our generalization bound scales with the bounds on the properties

[3] We use $\tilde{\mathcal{O}}(\cdot)$ to hide logarithmic factors.

in (a) and (b) above as satisfied on the training data, and with the reciprocal of the property in (c) i.e., the smallest absolute value of the pre-activations on the training data. Additionally, our bound has an explicit quadratic dependence on the depth of the network, which arises from the fact that we generalize $\mathcal{O}(D^2)$ conditions. Most importantly, our bound does not have a dependence on the product of the spectral norms of the weight matrices.

Theorem. F.1 ***(shorter version**; see Appendix F for the complete statement) Let $\zeta^\star_{d/d'}, \alpha^\star_d, \gamma^\star_d$ be a set of positive constants. For any margin $\gamma_{class} > 0$, and any $\delta > 0$, with probability $1-\delta$ over the draw of samples from $\mathcal{D}^m$, for any $\mathcal{W}$, if $\mathcal{W}$ satisfies the following three conditions for every training data point $(\mathbf{x}, y) \in S$:*

(a) Sufficiently small maximum row ℓ_2 norm of Jacobian d/d', $\max_h \|J^{d/d'}(\mathbf{x};\mathcal{W})[h]\| \leq \zeta^\star_{d/d'}$ for all d', d such that $1 \leq d' \leq d \leq D$.

(b) Sufficiently small ℓ_2 norm of layer d's output: $\|f^d(\mathbf{x};\mathcal{W})\| \leq \alpha^\star_d$ for all d such that $0 \leq d < D$.

(c) Sufficiently large pre-activations on layer d: $\min_h |g^d(\mathbf{x};\mathcal{W})[h]| \geq \gamma^\star_d$ for d such that $1 \leq d < D$.

then, for such $\mathcal{W}$, we have that:

$$Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[f(\mathbf{x};\mathcal{W})[y] < \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j]\right] \leq Pr_{(\mathbf{x},y)\sim S}\left[f(\mathbf{x};\mathcal{W})[y] < \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j] + \gamma_{class}\right] + \tilde{\mathcal{O}}\left(D^2\sqrt{\|\mathcal{W}-\mathcal{Z}\|_F^2/((m-1)(\sigma^\star)^2)}\right)$$

Here $1/\sigma^\star$ equals $\tilde{\mathcal{O}}(\sqrt{H}\max\{\mathcal{B}_{jac}, \mathcal{B}_{out}, \mathcal{B}_{hidden\text{-}preact}, \mathcal{B}_{output\text{-}preact}\})$, where[4]:

$$\mathcal{B}_{jac} := \max_{1\leq d'<d\leq D} \frac{\sum_{d''=d'+1}^{d} \zeta^\star_{d/d''}\zeta^\star_{d''-1/d'}}{\zeta^\star_{d/d'}}, \mathcal{B}_{out} := \max_{d\leq D-1} \frac{\sum_{d'=1}^{d} \zeta^\star_{d/d'}\alpha^\star_{d'-1}}{\alpha^\star_d}$$

$$\mathcal{B}_{hidden\text{-}preact} := \max_{d\leq D-1} \frac{\sum_{d'=1}^{d} \zeta^\star_{d/d'}\alpha^\star_{d'-1}}{\sqrt{H}\gamma^\star_d}, \mathcal{B}_{output\text{-}preact} := \frac{2\sum_{d'=1}^{D} \zeta^\star_{D/d'}\alpha^\star_{d'-1}}{\sqrt{H}\gamma_{class}}$$

In Figure 1, we demonstrate that in practice the row ℓ_2 norms of the Jacobian and the ℓ_2 norms of the hidden layer outputs are orders of magnitude smaller than products of the spectral norms. In Figure 2, we train networks of varying depth and demonstrate that the terms $\mathcal{B}_{\text{jac}}, \mathcal{B}_{\text{out}}, \mathcal{B}_{\text{output-preact}}$ also do not exhibit such severe exponential dependence on depth. (In Appendix F, we demonstrate that these terms decay with width, although our emphasis is not on the width-dependence.)

However, the bottleneck in our bound is $\mathcal{B}_{\text{hidden-preact}}$, which scales inversely with the magnitude of the smallest absolute pre-activation value of the network. In practice, this term can be arbitrarily large, even though it does not depend on the product of spectral norms/depth (Figure 2). This is because some hidden units can have arbitrarily small absolute pre-activation values – although this is true only for a small proportion of these units. To give an idea of the typical magnitude of the pre-activation values, in Figure 2, we plot a modified term called 'median-based'-$\mathcal{B}_{\text{hidden-preact}}$, where we replace $\gamma^\star_d$ in the denominator with a different constant that bounds the median absolute pre-activation value for each training datapoint; 'median-based'-$\mathcal{B}_{\text{hidden-preact}}$ turns out to be a significantly smaller quantity.

As noted before, we emphasize that the drawback highlighted above is a limitation in how we characterize noise-resilience through our conditions rather than a drawback in our general PAC-Bayesian framework itself. Improving this dependence is an important direction for future work. For example, one way to do this, would be to modify our analysis to allow perturbations large enough to flip a small proportion of the activation states; one could potentially formulate such realistic conditions by drawing inspiration from the conditions in Neyshabur et al. (2017); Arora et al. (2018).

However, we note that even though these prior approaches made more realistic assumptions about the magnitudes of the pre-activation values, the key limitation in these approaches is that even under our non-realistic assumption, their approaches would yield bounds only on stochastic/compressed networks. *Generalizing noise-resilience from training data to test data* is crucial to extending these bounds to the original network, which we accomplish.

[4]Note that α_0 is an upper bound on the 'zeroth' layer i.e., on $\|\mathbf{x}\|$, which we assumed to be B.

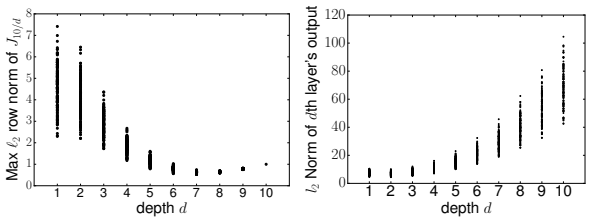

Figure 1: In all these plots, we train a network with $D = 11$, $H = 1280$ to minimize cross-entropy loss on the MNIST dataset. **Left**: Each black point corresponds to the maximum row ℓ_2 norm of the Jacobian $10/d$. Observe that for any d, these quantities are nowhere near as large as a naive upper bound that would roughly scale as $\prod_{d'=d}^{10} \|W_d\|_2 = 2^{10-d}$. **Right**: Each black point corresponds to a particular training example $\mathbf{x}$, and has y-value equal to the ℓ_2 norm of the output of layer d for that datapoint. A naive upper bound on this value would be $\|\mathbf{x}\| \prod_{d'=1}^{d} \|W_d\|_2 \approx 10 \cdot 2^d$, which would be at least 100 times larger than the observed value for $d = 10$.

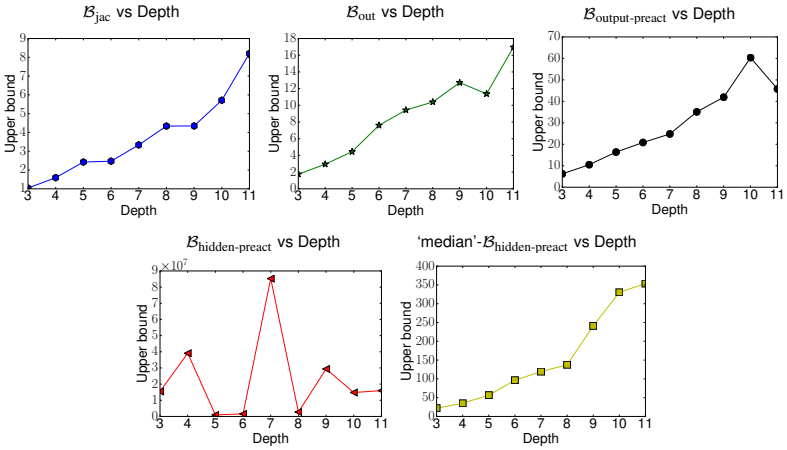

Figure 2: In the above figure, we plot the values of $\mathcal{B}_{\text{jac}}$ (**top left**) $\mathcal{B}_{\text{out}}$ (**top middle**) $\mathcal{B}_{\text{output-preact}}$ (**top right**) $\mathcal{B}_{\text{hidden-preact}}$ (**bottom left**) and 'median-based'-$\mathcal{B}_{\text{hidden-preact}}$ (**bottom right**, plotted only for comparison) computed on networks of width 40 with varying depth, trained on the MNIST dataset. While it is hard to quantify how $\mathcal{B}_{\text{hidden-preact}}$ varies, all other values do not increase with depth as rapidly as the product of the spectral norms (i.e., for a additive increase of 1 in the depth, the bound does **not** increase by a multiplicative factor of 2.).

5 Summary and Future Work

In this work, we introduced a novel PAC-Bayesian framework for leveraging the noise-resilience of neural networks on training data, to derive a generalization bound on the original uncompressed, deterministic network. The main philosophy of our approach is to first generalize the noise-resilience from training data to test data; this allows us to convert a PAC-Bayesian bound on a stochastic network to a standard margin-based generalization bound. We also build a fundamental technique for converting PAC-Bayesian bounds, that is more powerful than prior approaches. We apply our approach to ReLU based networks and derive a bound that scales with terms that capture the interactions between the weight matrices better than the product of spectral norms.

Our work brings up two concrete directions for future work. The most important direction is that of removing the dependence on our strong assumption that the magnitude of the pre-activation values of the network are not too small on training data. Another interesting direction would be to tighten the dependence on D^2 in our bound in Theorem F.1. More generally, our framework for generalizing noise-resilience is a promising tool for tapping the noise-resilience of the network to provide generalization guarantees on the original network; we hope that by further studying and applying this tool, we will be better equipped to answer the generalization puzzle in the future.

REFERENCES

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *The 35th International Conference on Machine Learning, ICML*, 2018.

Peter L. Bartlett, Dylan J. Foster, and Matus J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 6241–6250, 2017.

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. *International Conference on Learning Representations*, 2018.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *Computational Learning Theory, COLT 2018*, 2018.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, COLT*, 1993.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations*, 2017.

John Langford and Rich Caruana. (not) bounding the true error. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001]*, pp. 809–816, 2001.

John Langford and John Shawe-Taylor. Pac-bayes & margins. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002*, pp. 423–430, 2002.

David McAllester. Simplified pac-bayesian margin bounds. In *Learning Theory and Kernel Machines*, pp. 203–215. Springer Berlin Heidelberg, 2003.

David A. McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999a.

David A. McAllester. Pac-bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory, COLT 1999*, pp. 164–170, 1999b.

Vaishnavh Nagarajan and J. Zico Kolter. Generalization in deep networks: The role of distance from initialization. *Deep Learning: Bridging Theory and Practice Workshop in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *International Conference on Learning Representations Workshop Track*, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *Advances in Neural Information Processing Systems* to appear, 2017.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations*, 2018.

Daniel Soudry, Elad Hoffer, and Nathan Srebro. The implicit bias of gradient descent on separable data. *International Conference on Learning Representations*, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations*, 2017.

APPENDIX

A NOTATIONS (CONTINUED)

We will use upper-case symbols to denote matrices, and lower-case bold-face symbols to denote vectors. In order to make the mathematical statements/derivations easier to read, if we want to emphasize a term, say x, we write, $\boldsymbol{x}$.

Recall that we consider a neural netork of depth D (i.e., $D-1$ hidden layers and one output layer) mapping from $\mathbb{R}^N \to \mathbb{R}^K$, where K is the number of class labels in the learning task. The layers are fully connected with H units in each hidden layer, and with ReLU activations $\phi(\cdot)$ on all the hidden units and linear activations on the output units. We denote the parameters of the network using the symbol $\mathcal{W}$, which in turn denotes a set of weight matrices $W_1, W_2, \cdots, W_D$. Here, $W_1 \in \mathbb{R}^{H \times N}$, and $W_D \in \mathbb{R}^{K \times H}$ and for all other layers $d \neq 1, D$, $W_d \in \mathbb{R}^{H \times H}$. We will use the notation $\mathcal{W}_d$ to denote the first d weight matrices. We denote the vector of weights input to the hth unit on the dth layer (which corresponds to the hth row in W_d) as $\mathbf{w}_h^d$.

For any input $\mathbf{x} \in \mathbb{R}^N$, we denote the function computed by the network on that input as $f(\mathbf{x}; \mathcal{W}) = W_D \phi(W_{D-1} \cdots \phi(W_1 \mathbf{x}))$. For any $d = 1, \cdots, D-1$, we denote the output of the dth hidden layer after the activation by $f^d(\mathbf{x}; \mathcal{W})$. We denote the corresponding pre-activation values for that layer by $g^d(\mathbf{x}; \mathcal{W})$. We denote the value of the hth hidden unit on the dth layer after and before the activation by $f^d(\mathbf{x}; \mathcal{W})[h]$ and $g^d(\mathbf{x}; \mathcal{W})[h]$ respectively. Note that for the output layer $d = D$, these two values are equal as we assume only a linear activation. For $d = 0$, we define $f^0(\mathbf{x}; \mathcal{W}) = \mathbf{x}$. As a result, we have the following recursions:

$$
\begin{aligned}
f^d(\mathbf{x}; \mathcal{W}) &= \phi\left(g^d(\mathbf{x}; \mathcal{W})\right), d = 1, 2, \cdots, D-1 \\
f^D(\mathbf{x}; \mathcal{W}) &= g^D(\mathbf{x}; \mathcal{W}) = f(\mathbf{x}; \mathcal{W}), \\
g^d(\mathbf{x}; \mathcal{W}) &= W_d f^{d-1}(\mathbf{x}; \mathcal{W}), \forall d = 1, 2, \cdots, D \\
g^d(\mathbf{x}; \mathcal{W})[h] &= \mathbf{w}_h^d \cdot f^{d-1}(\mathbf{x}; \mathcal{W})[h], \forall d = 1, 2, \cdots, D
\end{aligned}
$$

For layers d', d such that $d' \leq d$, let us define $J^{d/d'}(\mathbf{x}; \mathcal{W})$ to be the Jacobian corresponding to the pre-activation values of layer d with respect to the pre-activation values of layer d' on an input $\mathbf{x}$. That is,

$$
J^{d/d'}(\mathbf{x}; \mathcal{W}) = \frac{\partial g^d(\mathbf{x}; \mathcal{W})}{\partial g^{d'}(\mathbf{x}; \mathcal{W})}
$$

In other words, this corresponds to the product of the 'activated' portion of the matrices $W_{d'+1}, W_{d'+2}, \cdots, W_d$, where the weights corresponding to inactive inputs are zeroed out. In short, we will call this 'Jacobian d/d''. Note that each row in this Jacobian corresponds to a unit on the dth layer, and each column corresponds to a unit on the d'th layer.

We will denote the parameters of a random initialization of the network by $\mathcal{Z} = (Z_1, Z_2, \cdots, Z_d)$. Let $\mathcal{D}$ be an underlying distribution over $\mathbb{R}^N \times \{1, 2, \cdots, K\}$ from which the data is drawn.

In our PAC-Bayesian analysis, we will use $\mathcal{U}$ to denote a set of D weight matrices $U_1, U_2, \cdots, U_D$ whose entries are sampled independently from a Gaussian. Furthermore, we will use $\mathcal{U}_d$ to denote only the first d of the randomly sampled weight matrices, and $\mathcal{W} + \mathcal{U}_d$ to denote a network where the d random matrices are added to the first d weight matrices in $\mathcal{W}$. Note that $\mathcal{W} + \mathcal{U}_0 = \mathcal{W}$. Thus, $f(\mathbf{x}; \mathcal{W} + \mathcal{U}_d)$ is the output of a network where the first d weight matrices have been perturbed. In our analysis, we will also need to study a perturbed network where the hidden units are frozen to be at the activation state they were at before the perturbation; we will use the notation $\mathcal{W}[+\mathcal{U}_d]$ to denote the weights of such a network.

For our statements regarding probability of events, we will use $\wedge$, $\vee$, and $\neg$ to denote the intersection, union and complement of events (to disambiguate from the set operators).

B USEFUL LEMMAS

In this section, we present some results. The first two results below will be useful for our noise resilience analysis.

HOEFFDING BOUND

Lemma B.1. *For $i = 1, 2, \cdots, n$, let X_i be independent random variables sampled from a Gaussian with mean μ_i and variance σ_i^2. Then for all $t \geq 0$, we have:*

$$Pr\left[\sum_{i=1}^{n}(X_i - \mu_i) \geq t\right] \leq \exp\left(-\frac{t^2}{2\sum_{i=1}^{n}\sigma_i^2}\right).$$

Or alternatively, for $\delta \in (0, 1]$

$$Pr\left[\sum_{i=1}^{n}(X_i - \mu_i) \geq \sqrt{2\sum_{i=1}^{n}\sigma_i^2 \ln\frac{1}{\delta}}\right] \leq \delta$$

Note that an identical inequality holds good symmetrically for the event $\sum_{i=1}^{n} X_i - \mu_i \leq -t$, and so the probability that the event $|\sum_{i=1}^{n} X_i - \mu_i| > t$ holds, is at most twice the failure probability in the above inequalities.

PRODUCT OF AN ENTRYWISE GAUSSIAN MATRIX AND A VECTOR

Lemma B.2. *Let U be a $H_1 \times H_2$ matrix where each entry is sampled from $\mathcal{N}(0, \sigma^2)$. Let $\mathbf{x}$ be an arbitrary vector in $\mathbb{R}^{H_2}$. Then, $U\mathbf{x} \sim \mathcal{N}(0, \|\mathbf{x}\|_2^2\sigma^2 I)$.*

Proof. $U\mathbf{x}$ is a random vector sampled from a multivariate Gaussian with mean $\mathbb{E}[U\mathbf{x}] = 0$ and co-variance $\mathbb{E}[U\mathbf{x}\mathbf{x}^T U^T]$. The (i, j)th entry in this covariance matrix is $\mathbb{E}[(\mathbf{u}_i^T\mathbf{x})(\mathbf{u}_j^T\mathbf{x})]$ where $\mathbf{u}_i$ and $\mathbf{u}_j$ are the ith and jth row in U. When $i = j$, $\mathbb{E}[(\mathbf{u}_i^T\mathbf{x})(\mathbf{u}_j^T\mathbf{x})] = \mathbb{E}[\|\mathbf{u}_i^T\mathbf{x}\|^2] = \sum_{h=1}^{H_2}\mathbb{E}[u_{ih}^2]x_h^2 = \sigma^2\|\mathbf{x}\|_2^2$. When $i \neq j$, since $\mathbf{u}_i$ and $\mathbf{u}_j$ are independent random variables, we will have $\mathbb{E}[(\mathbf{u}_i^T\mathbf{x})(\mathbf{u}_j^T\mathbf{x})] = \sum_{h=1}^{H_2}\mathbb{E}[u_{ih}x_h]\sum_{h=1}^{H_2}\mathbb{E}[u_{jh}x_h] = 0$.

□

KL DIVERGENCE OF GAUSSIANS. We will use the following KL divergence equality to bound the generalization error in our PAC-Bayesian analyses.

Lemma B.3. *Let P be the spherical Gaussian $\mathcal{N}(\boldsymbol{\mu}_1, \sigma^2 I)$ and Q be the spherical Gaussian $\mathcal{N}(\boldsymbol{\mu}_2, \sigma^2 I)$. Then, the KL-divergence between Q and P is:*

$$KL(Q\|P) = \frac{\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2}{2\sigma^2}$$

C PAC-BAYESIAN THEOREM

In this section, we will present our main PAC-Bayesian theorem that will guide our result about generalization. Concretely, our result extends the generalization bound provided by conventional PAC-Bayesian analysis McAllester (2003) – which is a generalization bound on the expected loss of a *distribution* of classifiers i.e., a stochastic classifier – to a generalization bound on a deterministic classifier. The way we reduce the PAC-Bayesian bound to a standard generalization bound, is different from the one pursued in previous works (Neyshabur et al., 2018; Langford & Shawe-Taylor, 2002; McAllester, 2003).

The generalization bound that we state below is a bit more general than standard generalization bounds. Typically, generalization bounds are on the classification error; however, as discussed in the main paper we will be dealing with generalizing multiple different conditions on the interactions between the weights of the network from the training data to test data.

So to state a bound that is general enough, we consider a set of generic functions $T_r(\mathcal{W}, \mathbf{x}, \mathbf{y})$ for $r = 1, 2, \cdots R$ (as initially considered in the main paper in Section 3). Recall that each of these functions compute a scalar value that corresponds to some *input-dependent property* of the network with parameters $\mathcal{W}$ for the datapoint $(\mathbf{x}, y)$. As an example, this property could simply be the margin of the function on the yth class i.e., $f(\mathbf{x}; \mathcal{W})[y] - \max_{j \neq y} f(\mathbf{x}; \mathcal{W})[j]$, though we consider many other input-dependent properties of the network when we apply this theorem.

Theorem C.1. *Let P be a prior distribution over the parameter space that is chosen independent of the training dataset. Let $\mathcal{U}$ be a random variable sampled entrywise from $\mathcal{N}(0, \sigma^2)$. Let $T_r(\cdot, \cdot, \cdot)$ and $\Delta_r > 0$ for $r = 1, 2, \cdots R$, be a set of functions and their corresponding margins. We define the network $\mathcal{W}$ to be noise-resilient with respect to all these functions, at a given data point $(\mathbf{x}, y)$ if:*

$$Pr_{\mathcal{U} \sim \mathcal{N}(0,\sigma^2)} \left[\exists r \,:\, |T_r(\mathcal{W}, \mathbf{x}, y) - T_r(\mathcal{W} + \mathcal{U}, \mathbf{x}, y)| > \frac{\Delta_r}{2} \right] \leq \frac{1}{\sqrt{m}}. \tag{3}$$

Let $\mu_{\mathcal{D}}(\{(T_r, \Delta_r)\}_{r=1}^{R}, \mathcal{W})$ denote the probability over the random draw of a point $(\mathbf{x}, y)$ drawn from $\mathcal{D}$, that the network with weights $\mathcal{W}$ is **not** *noise-resilient at $(\mathbf{x}, y)$ according to Equation 3. That is, let $\mu_{\mathcal{D}}(\{(T_r, \Delta_r)\}_{r=1}^{R}, \mathcal{W}) :=$*

$$Pr_{(\mathbf{x},y) \sim \mathcal{D}} \left[Pr_{\mathcal{U} \sim \mathcal{N}(0,\sigma^2)} \left[\exists r \,:\, |T_r(\mathcal{W}, \mathbf{x}, y) - T_r(\mathcal{W} + \mathcal{U}, \mathbf{x}, y)| > \frac{\Delta_r}{2} \right] > \frac{1}{\sqrt{m}} \right]$$

Similarly, let $\hat{\mu}_S(\{(T_r, \Delta_r)\}_{r=1}^{R}, \mathcal{W})$ denote the fraction of data points $(\mathbf{x}, y)$ in a dataset S for which the network is **not** *noise-resilient according to Equation 3. Then for any δ, with probability $1 - \delta$ over the draws of a sample set $S = \{(\mathbf{x}_i, y_i) \sim \mathcal{D} \,|i = 1, 2, \cdots, m\}$, for any $\mathcal{W}$ we have:*

$$Pr_{(\mathbf{x},y) \sim \mathcal{D}} \left[\exists r \,:\, T_r(\mathcal{W}, \mathbf{x}, y) < 0 \right] \leq \frac{1}{m} \sum_{(\mathbf{x},y) \in S} \mathbf{1} \left[\exists r \,:\, T_r(\mathcal{W}, \mathbf{x}, y) < \Delta_r \right] + \hat{\mu}_S(\{(T_r, \Delta_r)\}_{r=1}^{R}, \mathcal{W})$$
$$+ \mu_{\mathcal{D}}(\{(T_r, \Delta_r)\}_{r=1}^{R}, \mathcal{W}) + 2\sqrt{\frac{2KL(\mathcal{N}(\mathcal{W}, \sigma^2 I) \| P) + \ln \frac{2m}{\delta}}{m-1}} + \frac{2}{\sqrt{m} - 1}.$$

The reader maybe curious about how one would bound the term $\mu_{\mathcal{D}}$ in the above bound, as this term corresponds to noise-resilience with respect to *test* data. This is precisely what we bound later when we generalize the noise-resilience-related conditions satisfied on train data over to test data.

C.1 Key advantage of our bound.

The above approach differs from previous approaches used by Neyshabur et al. (2018); Langford & Shawe-Taylor (2002); McAllester (2003) in how strong a noise-resilience we require of the classifier to provide the generalization guarantee. The stronger the noise-resilience requirement, the more price we have to pay when we jump from the PAC-Bayesian guarantee on the stochastic classifier to a guarantee on the deterministic classifier. We argue that *our noise-resilience requirement is a much milder condition and therefore promises tighter guarantees*.

More concretely, to arrive at a reasonable generalization guarantee in our setup, we would need that $\mu_{\mathcal{D}}$ and $\hat{\mu}_S$ are both only as large as $\mathcal{O}(1/\sqrt{m})$. In other words, we would want the following for $(\mathbf{x}, y) \sim \mathcal{D}$ and for $(\mathbf{x}, y) \sim S$:

$$\Pr_{(\mathbf{x},y)} \left[\Pr_{\mathcal{U} \sim \mathcal{N}(0,\sigma^2)} \left[\exists r \,:\, |T_r(\mathcal{W}, \mathbf{x}, y) - T_r(\mathcal{W} + \mathcal{U}, \mathbf{x}, y)| > \frac{\Delta_r}{2} \right] > \frac{1}{\sqrt{m}} \right] = \mathcal{O}(1/\sqrt{m}).$$

Previous works require a noise resilience condition of the form that with high probability a particular perturbation does not perturb the classifier output on *any input*. For example, the noise-resilience condition used in Neyshabur et al. (2018) written in terms of our notations, would be:

$$\Pr_{\mathcal{U} \sim \mathcal{N}(0,\sigma^2)} \left[\exists \mathbf{x} \,:\, \exists r \,:\, |T_r(\mathcal{W}, \mathbf{x}, y) - T_r(\mathcal{W} + \mathcal{U}, \mathbf{x}, y)| > \frac{\Delta_r}{2} \right] \leq \frac{1}{2}.$$

The main difference between the above two formulations is in what makes a particular perturbation (un)favorable for the classifier. In our case, we deem a perturbation unfavorable only after fixing the datapoint. However, in the earlier works, a perturbation is deemed unfavorable if it perturbs the classifier output sufficiently on *some* datapoint from the domain of the distribution. While this difference is subtle, the earlier approach would lead to a much more pessimistic analysis of these perturbations. In our analysis, this weakened noise resilience condition will be critical in analyzing the Gaussian perturbations more carefully than in Neyshabur et al. (2018) i.e., we can bound the perturbation in the classifier output more tightly by analyzing the Gaussian perturbation for a fixed input point.

Note that one way our noise resilience condition would seem stronger in that on a given datapoint we want less than $1/\sqrt{m}$ mass of the perturbations to be unfavorable for us, while in previous bounds, there can be as much as $1/2$ probability mass of perturbations that are unfavorable. In our analysis, this will only weaken our generalization bound by a $\ln \sqrt{m}$ factor in comparison to previous bounds (while we save other significant factors).

C.2 Proof for Theorem C.1

Proof. The starting point of our proof is a standard PAC-Bayesian theorem McAllester (2003) which bounds the generalization error of a stochastic classifier. Let P be a data-independent prior over the parameter space. Let $\mathcal{L}(\mathcal{W}, \mathbf{x}, y)$ be any loss function that takes as input the network parameter, and a datapoint $\mathbf{x}$ and its true label y and outputs a value in $[0, 1]$. Then, we have that, with probability $1 - \delta$ over the draw of $S \sim \mathcal{D}^m$, for every distribution Q over the parameter space, the following holds:

$$\mathbb{E}_{\tilde{\mathcal{W}} \sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}(\tilde{\mathcal{W}}, \mathbf{x}, y)\right]\right] \leq \mathbb{E}_{\tilde{\mathcal{W}} \sim Q}\left[\frac{1}{m}\sum_{(\mathbf{x},y)\in S}\mathcal{L}(\tilde{\mathcal{W}}, \mathbf{x}, y)\right] + 2\sqrt{\frac{2KL(Q\|P) + \ln\frac{2m}{\delta}}{m-1}} \quad (4)$$

In other words, the statement tells us that except for a δ proportion of bad draws of m samples, the test loss of the stochastic classifier $\tilde{\mathcal{W}} \sim Q$ would be close to its train loss. This holds for every possible distribution Q, which allows us to cleverly choose Q based on S. As is the convention, we choose Q to be the distribution of the stochastic classifier picked from $\mathcal{N}(\mathcal{W}, \sigma^2 I)$ i.e., a Gaussian perturbation of the deterministic classifier $\mathcal{W}$.

Relating test loss of stochastic classifier to deterministic classifier. Now our task is to bound the loss for the deterministic classifier $\mathcal{W}$, $\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists r \mid \mathcal{T}_r(\mathcal{W}, \mathbf{x}, y) < 0\right]$. To this end, let us define the following margin-based variation of this loss for some $c \geq 0$:

$$\mathcal{L}_c(\mathcal{W}, \mathbf{x}, y) = \begin{cases} 1 & \exists r \,:\, T_r(\mathcal{W}, \mathbf{x}, y) < c\Delta_r \\ 0 & \text{otherwise,} \end{cases}$$

and so we have $\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists r \mid \mathcal{T}_r(\mathcal{W}, \mathbf{x}, y) < 0\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}_0(\mathcal{W}, \mathbf{x}, y)\right]$. First, we will bound the expected $\mathcal{L}_0$ of a deterministic classifier by the expected $\mathcal{L}_{1/2}$ of the stochastic classifier; then we will bound the test $\mathcal{L}_{1/2}$ of the stochastic classifier using the PAC-Bayesian bound.

We will split the expected loss of the deterministic classifier into an expectation over datapoints for which it is noise-resilient with respect to Gaussian noise and an expectation over the rest. To write this out, we define, for a datapoint $(\mathbf{x}, y)$, $\mathfrak{N}(\mathcal{W}, \mathbf{x}, y)$ to be the event that $\mathcal{W}$ is noise-resilient at $(\mathbf{x}, y)$ as defined in Equation 3 in the theorem statement:

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\right] &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\middle|\ \mathfrak{N}(\mathcal{W},\mathbf{x},y)\right]\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right] \\ &+ \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\middle|\ \neg\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right]}_{\leq 1}\underbrace{\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\neg\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right]}_{\mu_{\mathcal{D}}(\{(T_r,\Delta_r)\}_{r=1}^{R},\mathcal{W})} \\ &\leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\middle|\ \mathfrak{N}(\mathcal{W},\mathbf{x},y)\right]\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right] \\ &+ \mu_{\mathcal{D}}(\{(T_r,\Delta_r)\}_{r=1}^{R},\mathcal{W}) \qquad (5)\end{aligned}$$

To further continue the upper bound on the left hand side, we turn our attention to the stochastic classifier's loss on the noise-resilient part of the distribution $\mathcal{D}$ (we will lower bound this term in terms of the first term on the right hand side above). For simplicity of notations, we will write $\mathcal{D}'$ to denote the distribution $\mathcal{D}$ conditioned on $\mathfrak{N}(\mathcal{W},\mathbf{x},y)$. Also, let $\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)$ be the favorable event that for a given data point $(\mathbf{x},y)$ and a draw of the stochastic classifier, $\tilde{\mathcal{W}}$, it is the case that for every r, $|T_r(\mathcal{W},\mathbf{x},y) - T_r(\tilde{\mathcal{W}},\mathbf{x},y)| \leq \Delta_r/2$. Then, the stochastic classifier's loss $\mathcal{L}_{1/2}$ on $\mathcal{D}'$ is:

$$\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right]$$

splitting the inner expectation over the favorable and unfavorable perturbations, and using linearity of expectations,

$$\begin{aligned}&= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\middle|\ \mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \\ &+ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\middle|\ \neg\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\neg\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right]\end{aligned}$$

to lower bound this, we simply ignore the second term (which is positive)

$$\geq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\middle|\ \mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right].$$

Next, we use the following fact: if $\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y) = 0$, then for all r, $T_r(\tilde{\mathcal{W}},x,y) \geq \Delta_r/2$ and if $\tilde{\mathcal{W}}$ is a favorable perturbation of $\mathcal{W}$, then for all r, $T_r(\mathcal{W},x,y) \geq T_r(\tilde{\mathcal{W}},\mathbf{x},y) - \Delta_r/2 > 0$ i.e., $\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y) = 0$ implies $\mathcal{L}_0(\mathcal{W},\mathbf{x},y) = 0$. Hence if $\tilde{\mathcal{W}}$ is a favorable perturbation then, $\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y) \geq \mathcal{L}_0(\mathcal{W},\mathbf{x},y)$. Therefore, we can lower bound the above expression by replacing the stochastic classifier with the deterministic classifier (and thus ridding ourselves of the expectation over Q):

$$\geq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right].$$

Since the favorable perturbations for a fixed datapoint drawn from $\mathcal{D}'$ have sufficiently high probability (that is, $\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right] \geq 1 - 1/\sqrt{m}$), we have:

$$\geq \left(1 - \frac{1}{\sqrt{m}}\right)\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\right].$$

Thus, we have a lower bound on the stochastic classifier's loss that is in terms of the deterministic classifier's loss on the noise-resilient datapoints. Rearranging it, we get an upper bound on the latter:

$$
\begin{aligned}
\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\right] &\leq \frac{1}{\left(1-\frac{1}{\sqrt{m}}\right)}\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \\
&\leq \left(1+\frac{1}{\sqrt{m}-1}\right)\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \\
&\leq \mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \\
&+ \frac{1}{\sqrt{m}-1}\underbrace{\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right]}_{\leq 1} \\
&\leq \mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] + \frac{1}{\sqrt{m}-1}
\end{aligned}
$$

Thus, we have an upper bound on the expected loss of the deterministic classifier $\mathcal{W}$ on the noise-resilient part of the distribution. Plugging this back in the first term of the upper bound on the deterministic classifier's loss on the whole distribution $\mathcal{D}$ in Equation 5 we get :

$$
\begin{aligned}
\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}_0(\mathcal{W},\mathbf{x},y)\right] \leq &\left(\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] + \frac{1}{\sqrt{m}-1}\right)\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right] + \\
&\mu_{\mathcal{D}}(\{(T_r,\Delta_r)\}_{r=1}^{R},\mathcal{W})
\end{aligned}
$$

rearranging, we get:

$$
\begin{aligned}
&\leq \left(\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}'}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right]\right)\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right] + \\
&\mu_{\mathcal{D}}(\{(T_r,\Delta_r)\}_{r=1}^{R},\mathcal{W}) + \frac{1}{\sqrt{m}-1}\underbrace{\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right]}_{\leq 1}
\end{aligned}
$$

rewriting the expectation over $\mathcal{D}'$ explicitly as an expectation over $\mathcal{D}$ conditioned on $\mathfrak{N}(\mathcal{W},\mathbf{x},y)$, we get:

$$
\begin{aligned}
&\leq \left(\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\,\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\middle|\ \mathfrak{N}(\mathcal{W},\mathbf{x},y)\right]\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathfrak{N}(\mathcal{W},\mathbf{x},y)\right]\right]\right) + \\
&\mu_{\mathcal{D}}(\{(T_r,\Delta_r)\}_{r=1}^{R},\mathcal{W}) + \frac{1}{\sqrt{m}-1}
\end{aligned}
$$

the first term above is essentially an expectation of a loss over the distribution $\mathcal{D}$ with the loss set to be zero over the non-noise-resilient datapoints and set to be $\mathcal{L}_{1/2}$ over the noise-resilient datapoints; thus we can upper bound it with the expectation of the $\mathcal{L}_{1/2}$ loss over the whole distribution $\mathcal{D}$:

$$
\leq \mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] + \cdot\mu_{\mathcal{D}}(\{(T_r,\Delta_r)\}_{r=1}^{R},\mathcal{W}) + \frac{1}{\sqrt{m}-1} \tag{6}
$$

Now observe that we can upper bound the first term here using the PAC-Bayesian bound by plugging in $\mathcal{L}_{1/2}$ for the generic $\mathcal{L}$ in Equation 4; however, the bound would still be in terms of the stochastic classifier's train error. To get the generalization bound we seek, which involves the deterministic classifier's train error, we need to take one final step mirroring these tricks on the train loss.

RELATING THE STOCHASTIC CLASSIFIER'S TRAIN LOSS TO DETERMINISTIC CLASSIFIER'S TRAIN LOSS. Our analysis here is almost identical to the previous analysis. Instead of working with the distribution $\mathcal{D}$ and $\mathcal{D}'$ we will work with the training data set S and a subset of it S' for which noise resilience property is satisfied by $\mathcal{W}$. Below, to make the presentation neater, we use $(\mathbf{x},y)\sim S$ to denote uniform sampling from S.

First, we upper bound the stochastic classifier's train loss ($\mathcal{L}_{1/2}$) as follows:

$$\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim S}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] = \mathbb{E}_{(\mathbf{x},y)\sim S}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right]$$

splitting over the noise-resilient points S' ($(\mathbf{x},y) \in S$ for which $\mathfrak{N}(\mathcal{W},\mathbf{x},y)$ holds) like in Equation 5, we can upper bound as:

$$\begin{aligned} &\leq \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right]\Pr_{(\mathbf{x},y)\sim S}\left[(\mathbf{x},y)\in S'\right] \\ &+ \hat{\mu}_S(\{(T_r,\Delta_r)\}_{r=1}^{R},\mathcal{W}) \end{aligned} \tag{7}$$

We can upper bound the first term by first splitting it over the favorable and unfavorable perturbations like we did before:

$$\begin{aligned} &\mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \\ &= \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\middle|\ \mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \\ &+ \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\middle|\ \neg\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\neg\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \end{aligned}$$

To upper bound this, we apply a similar argument. First, if $\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y) = 1$, then $\exists r$ such that $T_r(\tilde{\mathcal{W}},x,y) < \Delta_r/2$ and if $\tilde{\mathcal{W}}$ is a favorable perturbation then for that value of r, $T_r(\mathcal{W},x,y) < T_r(\tilde{\mathcal{W}},\mathbf{x},y) + \Delta_r/2 < \Delta_r$. Thus if $\tilde{\mathcal{W}}$ is a favorable perturbation then, $\mathcal{L}_1(\mathcal{W},\mathbf{x},y) = 1$ whenever $\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y) = 1$ i.e., $\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y) \leq \mathcal{L}_1(\mathcal{W},\mathbf{x},y)$. Next, we use the fact that the unfavorable perturbations for a fixed datapoint drawn from S' have sufficiently low probability i.e., $\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\neg\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right] \leq 1/\sqrt{m}$. Then, we get the following upper bound on the above equations, by replacing the stochastic classifier with the deterministic classifier (and thus ignoring the expectation over Q):

$$\begin{aligned} &\leq \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_1(\mathcal{W},\mathbf{x},y)\right|\ \mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\underbrace{\Pr_{\tilde{\mathcal{W}}\sim Q}\left[\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]}_{\leq 1}\right] \\ &+ \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\underbrace{\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\middle|\ \neg\mathfrak{U}(\tilde{\mathcal{W}},\mathbf{x},y)\right]}_{\leq 1}\frac{1}{\sqrt{m}}\right] \\ &\leq \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathcal{L}_1(\mathcal{W},\mathbf{x},y)\right] + \frac{1}{\sqrt{m}} \end{aligned}$$

Plugging this back in the first term of Equation 7, we get:

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathcal{W}}\sim Q}\left[\mathbb{E}_{(\mathbf{x},y)\sim S}\left[\mathcal{L}_{1/2}(\tilde{\mathcal{W}},\mathbf{x},y)\right]\right] \leq & \left(\mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathcal{L}_1(\mathcal{W},\mathbf{x},y)\right]+\frac{1}{\sqrt{m}}\right)\Pr_{(\mathbf{x},y)\sim S}\left[(\mathbf{x},y)\in S'\right] \\
& +\hat{\mu}_S(\{(T_r,\Delta_r)\}_{r=1}^R,\mathcal{W}) \\
\leq & \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathcal{L}_1(\mathcal{W},\mathbf{x},y)\right]\Pr_{(\mathbf{x},y)\sim S}\left[(\mathbf{x},y)\in S'\right] \\
& +\hat{\mu}_S(\{(T_r,\Delta_r)\}_{r=1}^R,\mathcal{W})+\frac{1}{\sqrt{m}}\underbrace{\Pr_{(\mathbf{x},y)\sim S}\left[(\mathbf{x},y)\in S'\right]}_{\leq 1} \\
\leq & \mathbb{E}_{(\mathbf{x},y)\sim S'}\left[\mathcal{L}_1(\mathcal{W},\mathbf{x},y)\right]\Pr_{(\mathbf{x},y)\sim S}\left[(\mathbf{x},y)\in S'\right]+\frac{1}{\sqrt{m}} \\
& +\hat{\mu}_S(\{(T_r,\Delta_r)\}_{r=1}^R,\mathcal{W})
\end{aligned}$$

since the first term is effectively the expectation of a loss over the whole distribution with the loss set to be zero on the non-noise-resilient points and set to $\mathcal{L}_1$ over the rest, we can upper bound it by setting the loss to be $\mathcal{L}_1$ over the whole distribution:

$$\leq \mathbb{E}_{(\mathbf{x},y)\sim S}\left[\mathcal{L}_1(\mathcal{W},\mathbf{x},y)\right]+\hat{\mu}_S(\{(T_r,\Delta_r)\}_{r=1}^R,\mathcal{W})+\frac{1}{\sqrt{m}}$$

Applying the above upper bound and the bound in Equation 6 into the PAC-Bayesian result of Equation 4 yields our result (Note that combining these equations would produce the term $\frac{1}{\sqrt{m}}+\frac{1}{\sqrt{m}-1}$ which is at most $\frac{2}{\sqrt{m}-1}$, which we reflect in the final bound.).

□

D Proof for Theorem 3.1

In this section, we present the proof for the (abstract) generalization guarantee presented in Section 3. Our proof is based on the following recursive inequality that holds for all $r < R$ (we will prove a similar, but slightly different inequality for $r = R$):

$$\begin{aligned}
\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists q\leq r\ T_q(\mathcal{W},\mathbf{x},y)<0\right] \leq & \Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists q<r\ T_q(\mathcal{W},\mathbf{x},y)<0\right] \\
& +\underbrace{\tilde{\mathcal{O}}\left(\sqrt{\frac{2KL(\mathcal{N}(\mathcal{W},\sigma^2 I)\|P)}{m-1}}\right)}_{\text{generalization error for condition } r}
\end{aligned} \tag{8}$$

That is, we bound the probability mass of test points such that any one of the first r condition fails, in terms of the probability mass of points where one of the first $r-1$ conditions fail and an error that corresponds to how much error there can be in generalizing the rth condition from the training data. First, we consider the base case when $r = 1$, and apply the PAC-Bayes-based guarantee from Theorem C.1 on the singleton set of functions consisting only of T_1. First we have from our assumption that on all the training data, the condition $T_1(\mathcal{W},\mathbf{x},y) > \Delta_1^\star$ is satisfied. Thus, the first term in the upper bound in Theorem C.1 is zero. Next, we can show that the terms $\hat{\mu}_S$ and $\hat{\mu}_\mathcal{D}$ would be zero too. This follows from the fact that Equation 2 holds in this framework. Applying this equation for $r = 1$, for $\sigma = \sigma^\star$, we get that for all $\forall \mathbf{x} \in \mathcal{X}$ the following inequality holds:

$$\Pr_{\mathcal{U}}\left[\max_l |\rho_1^{(l)}(\mathcal{W}+\mathcal{U},\mathbf{x},y)-\rho_1^{(l)}(\mathcal{W},\mathbf{x},y)|>\frac{\Delta_1(\sigma^\star)}{2}\right]\leq\frac{1}{R\sqrt{m}}.$$

Since, $\sigma^\star$ was chosen such that $\Delta_1(\sigma^\star) \leq \Delta_1^\star$, we effectively have:

$$\Pr_{\mathcal{U}}\left[\max_l |\rho_1^{(l)}(\mathcal{W}+\mathcal{U},\mathbf{x},y)-\rho_1^{(l)}(\mathcal{W},\mathbf{x},y)|>\frac{\Delta_1^\star}{2}\right]\leq\frac{1}{R\sqrt{m}}.$$

Then, using Equation 1, we have that

$$\Pr_{\mathcal{U}}\left[|T_1(\mathcal{W}+\mathcal{U},\mathbf{x},y)-T_1(\mathcal{W},\mathbf{x},y)|>\frac{\Delta_1^\star}{2}\right]\leq\frac{1}{R\sqrt{m}}.$$

Effectively this establishes that the noise-resilience requirement of Equation 3 holds on all possible inputs, for T_1, thus proving our claim that the terms $\hat{\mu}_S$ and $\hat{\mu}_{\mathcal{D}}$ would be zero. Thus, we will get that

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[T_1(\mathcal{W},\mathbf{x},y)<0\right]\leq\tilde{\mathcal{O}}\left(\sqrt{\frac{2KL(\mathcal{N}(\mathcal{W},\sigma^2 I)\|P)}{m-1}}\right)$$

which proves the recursion statement for the base case.

To prove the recursion for some arbitrary $r<R$, we again apply the PAC-Bayes-based guarantee from Theorem C.1, but on the set of first r functions. Again, we will have that the first term in the guarantee would be zero, since the corresponding conditions are satisfied on the training data. Now, to bound the proportion of bad points $\hat{\mu}_S$ and $\hat{\mu}_{\mathcal{D}}$, we claim that the network is noise-resilient according to Equation 3 for any input that satisfies the $r-1$ conditions, $T_q(\mathcal{W},\mathbf{x},y)>0$ for $q=1,2,\cdots,r-1$. Since these conditions are assumed to be satisfied by a margin on the training data, it immediately implies that $\hat{\mu}_S(\{(T_q,\Delta_q)\}_{q=1}^r,\mathcal{W})=0$. On the test data, we can bound $\mu_{\mathcal{D}}(\{(T_q,\Delta_q)\}_{q=1}^r,\mathcal{W})$ in terms of $\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists q<r\ T_q(\mathcal{W},\mathbf{x},y)<0\right]$, thus giving rise to the recursion in Equation 8.

Now, to prove our claim, consider an input $\mathbf{x}$ such that $T_q(\mathcal{W},\mathbf{x},y)>0$ for $q=1,2,\cdots,r-1$. We then have the following:

$$\begin{aligned}&\Pr_{\mathcal{U}}\left[\exists q\leq r\ :\ |T_q(\mathcal{W},\mathbf{x},y)-T_q(\mathcal{W}+\mathcal{U},\mathbf{x},y)|>\frac{\Delta_q^\star}{2}\right]\\&\leq\Pr_{\mathcal{U}}\left[\exists q\leq r\ :\ |T_q(\mathcal{W},\mathbf{x},y)-T_q(\mathcal{W}+\mathcal{U},\mathbf{x},y)|>\frac{\Delta_q(\sigma^\star)}{2}\right]\end{aligned}$$

because $\Delta_q(\sigma^\star)\leq\Delta_q^\star$

$$\leq\Pr_{\mathcal{U}}\left[\exists q\leq r\ :\max_l\ |\rho_q^{(l)}(\mathcal{W}+\mathcal{U},\mathbf{x},y)-\rho_q^{(l)}(\mathcal{W},\mathbf{x},y)|>\frac{\Delta_q(\sigma^\star)}{2}\right]$$

the above inequality follows from the constraint we imposed on T_r in Equation 1

$$\begin{aligned}&\leq\sum_{q=1}^r Pr\Big[\max_l\ |\rho_q^{(l)}(\mathcal{W}+\mathcal{U},\mathbf{x},y)-\rho_q^{(l)}(\mathcal{W},\mathbf{x},y)|>\frac{\Delta_q(\sigma^\star)}{2}\wedge\\&\qquad\forall q'<q,\ \max_l\left|\rho_{q'}^{(l)}(\mathcal{W},\mathbf{x},y)-\rho_{q'}^{(l)}(\mathcal{W}+\mathcal{U},\mathbf{x},y)\right|<\frac{\Delta_{q'}(\sigma^\star)}{2}\Big]\\&\leq\sum_{q=1}^r\frac{1}{R\sqrt{m}}\leq\frac{1}{\sqrt{m}}\end{aligned}$$

Thus, we have that $\mathbf{x}$ satisfies the noise-resilience condition from Equation 3 if it also satisfies $T_q(\mathcal{W},\mathbf{x},y)>0$ for $q=1,2,\cdots,r-1$. This proves the recursion in Equation 8.

Finally, we can apply a similar argument for T_R with a small change since the first term in the guarantee from Theorem C.1 is not explicitly assumed to be zero; we will get an inequality in terms of the number of training points that are not classified correctly by a margin, giving rise to the margin-based bound:

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists q \leq R\ T_q(\mathcal{W},\mathbf{x},y) < 0\right] \leq \frac{1}{m}\sum_{(\mathbf{x},y)\in S}\mathbf{1}[T_R(\mathcal{W},\mathbf{x},y) < \Delta_R] + \Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists q < R\ T_q(\mathcal{W},\mathbf{x},y) < 0\right] + \tilde{\mathcal{O}}\left(\sqrt{\frac{2KL(\mathcal{N}(\mathcal{W},\sigma^2 I)\|P)}{m-1}}\right)$$

By using the fact that the test error is upper bounded by the left hand side in the above equation, applying the recursion on the right hand side R times, we get our final result.

E PERTURBATION BOUNDS

E.1 OVERVIEW OF THE BOUNDS.

In this section, we will quantify the noise resilience of a network in different aspects. Each of our bounds has the following structure: we fix an input point $(\mathbf{x}, y)$, and then say that with high probability over a Gaussian perturbation of the network's parameters, a particular *input-dependent property* of the network (say the output of the network, or the pre-activation value of a particular unit h at a particular layer d, or say the Frobenius norm sof its active weight matrices), changes only by a small magnitude proportional to the variance σ^2 of the Gaussian perturbation of the parameters.

A key feature of our bounds is that they do not involve the product of the spectral norm of the weight matrices and hence save us an exponential factor in the final generalization bound. Instead, the bound in the perturbation of a particular property will be in terms of i) the magnitude of the some 'preceding' properties (typically, these are properties of the lower layers) of the network, and ii) how those preceding properties themselves respond to perturbations. For example, an upper bound in the perturbation of the dth layer's output would involve the ℓ_2 norm of the lower layers $d' < d$, and how much they would blow up under these perturbations. More concretely, the bound on the perturbation in the norm of rows of the Jacobian d/d' depends on i) the magnitude of the norms related to all the lower layer Jacobians, and also the 'higher' Jacobians of layer d i.e., d/d'' for $d'' > d'$ and ii) the perturbation suffered by the higher Jacobians of layer d. Second, the bound on the perturbation of both the ℓ_2 norm and the pre-activations of layer d depend on i) the maximum row ℓ_2 norm of the Jacobians d/d' for each $d' < d$ and the ℓ_2 norms of all lower layers $d' < d$ ii) the perturbation suffered by the ℓ_2 norms of all lower layers.

Intuitively, when the Jacobian of layer d with respect to some d' has small ℓ_2 row norms, it diminishes the effect of noise injected into the matrix immediately after layer d/immediately before layer d'. Similarly, when the ℓ_2 norm of the output of the dth layer is small, it diminishes the effect of noise injected into the subsequent matrix. Finally, assuming sufficiently large pre-activation values prevents hidden units from crossing non-linearities with high probability; this then allows us to propagate the Gaussian noise 'linearly' through the network.

E.2 SOME NOTATIONS.

To formulate our lemma statement succinctly, we design a notation wherein we define a set of 'tolerance parameters' which we will use to denote the extent of perturbation suffered by a particular property of the network.

Let $\hat{\mathscr{C}}$ denote a 'set' (more on what we mean by a set below) of positive tolerance values, consisting of the following elements:

1. $\hat{\zeta}_{d/d'}$ for each layer $d = 1, 2, \cdots, D$, and $d' = 1, \cdots, d$ (a tolerance value for the ℓ_2 norm of each row of the Jacobians at layer d)
2. $\hat{\alpha}_d$, for each layer $d = 1, \cdots, D-1$ (a tolerance value for the ℓ_2 norm of the output of layer d)
3. $\hat{\gamma}_d$ for each layer $d = 1, \cdots, D$. (a tolerance value for the magnitude of the pre-activations of layer d)

Notes about (abuse of) notation:

- We call $\hat{\mathscr{C}}$ a 'set' to denote a group of related constants into a single symbol. Each element in this set has a particular semantic associated with it, unlike the standard notation of a set, and so when we refer to, say $\hat{\zeta}_{d/d'} \in \hat{\mathscr{C}}$, we are indexing into the set to pick a particular element.
- We will use the subscript $\hat{\mathscr{C}}_d$ to index into a subset of only those tolerance values corresponding to layers from 1 until d.

Next we define two events. The first event formulates the scenario that for a given input, a particular perturbation of the weights until layer d brought about very little change in the properties of these layers (within some tolerance levels). The second event formulates the scenario that the perturbation did not flip the activation states of the network.

Definition E.1. *Given an input* $\mathbf{x}$*, and an arbitrary set of constants* $\hat{\mathscr{C}}'$*, for any perturbation* $\mathcal{U}$ *of* $\mathcal{W}$*, we denote by* $\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \hat{\mathscr{C}}', \mathbf{x})$ *the event that:*

- *for each* $\hat{\zeta}'_{d/d'} \in \hat{\mathscr{C}}'$*, the maximum perturbation in the* ℓ_2 *norm of a row of the Jacobian* d/d' *is bounded as* $\max_h \left| \left\| J^{d/d'}(\mathbf{x};\mathcal{W})[h] \right\| - \left\| J^{d/d'}(\mathbf{x};\mathcal{W}+\mathcal{U})[h] \right\| \right| \leq \hat{\zeta}'_{d/d'}$.
- *for each* $\hat{\alpha}'_d \in \hat{\mathscr{C}}'$*, the perturbation in the* ℓ_2 *norm of layer* d *activations is bounded as* $\left| \left\| f^d(\mathbf{x};\mathcal{W}) \right\| - \left\| f^d(\mathbf{x};\mathcal{W}+\mathcal{U}) \right\| \right| \leq \hat{\alpha}'_d$.
- *for each* $\hat{\gamma}'_d \in \hat{\mathscr{C}}'$*, the maximum perturbation in the preactivation of hidden units on layer* d *is bounded as* $\max_h \left| f^d(\mathbf{x};\mathcal{W})[h] - f^d(\mathbf{x};\mathcal{W}+\mathcal{U})[h] \right| \leq \hat{\gamma}'_d$.

NOTE: A subtle point in the above definition (which we take advantage of, when we state our results) is that if we supply only a subset of $\hat{\mathscr{C}}$ (say $\hat{\mathscr{C}}_d$ instead of the whole of $\hat{\mathscr{C}}$) to the above event, $\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \cdot, \mathbf{x})$, then it would denote the event that the perturbations suffered by only that subset of properties is within the respective tolerance values.

Next, we define the event that the perturbations do not affect the activation states of the network.

Definition E.2. *For any perturbation* $\mathcal{U}$ *of the matrices* $\mathcal{W}$*, let* $\textsc{Unchanged-Acts}_d(\mathcal{W}+\mathcal{U}, \mathbf{x})$ *denote the event that none of the activation states of the first* d *layers change on perturbation.*

E.3 MAIN LEMMA.

Our results here are styled similar to the equations required by Equation 2 presented in the main paper. For a given input point and for a particular property of the network, roughly, we bound the the probability that a perturbation affects the value of the property while none of the preceding preceding properties themselves are perturbed beyond a certain tolerance level.

Lemma E.1. *Let* $\hat{\mathscr{C}}$ *be a set of constants. For any* $\hat{\delta} > 0$*, define* $\hat{\mathscr{C}}'$ *in terms of* $\hat{\mathscr{C}}$*, for all* $d = 1, 2, \cdots, D$ *as follows:*

$$\hat{\zeta}'_{d/d'} := \sigma \sum_{d''=d'+1}^{d} \max_h \left\| J^{d/d''}(\mathbf{x};\mathcal{W})[h] \right\| \left(\left\| J^{d''-1/d'}(\mathbf{x};\mathcal{W}) \right\|_F + \hat{\zeta}_{d''-1/d'}\sqrt{H} \right) \sqrt{4 \ln \frac{DH}{\hat{\delta}}} \quad \forall d' = d-1, \cdots, 1$$

$$\hat{\alpha}'_d := \sigma \sum_{d'=1}^{d} \left\| J^{d/d'}(\mathbf{x};\mathcal{W}) \right\|_F \left(\left\| f^{d'-1}(\mathbf{x};\mathcal{W}) \right\| + \hat{\alpha}_{d'-1} \right) \sqrt{2 \ln \frac{2DH}{\hat{\delta}}}$$

$$\hat{\gamma}'_d := \sigma \sum_{d'=1}^{d} \max_h \left\| J^{d/d'}(\mathbf{x};\mathcal{W})[h] \right\| \left(\left\| f^{d'-1}(\mathbf{x};\mathcal{W}) \right\| + \hat{\alpha}_{d'-1} \right) \sqrt{2 \ln \frac{2DH}{\hat{\delta}}}$$

$$\hat{\zeta}'_{d/d} := 0$$

$$\hat{\alpha}'_0 = 0$$

Let $\mathcal{U}_d$ be sampled entrywise from $\mathcal{N}(0, \sigma^2)$ for any d. Then, the following statements hold good:

1. Bound on perturbation of ℓ_2 norm on the rows of the Jacobians d/d'. *For all $d' = d, d-1, \cdots, 1$:*

$$Pr_{\mathcal{U}}\Big[\neg\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \{\hat{\zeta}'_{d/d'}\}, \mathbf{x}) \wedge$$
$$\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \hat{\mathscr{C}}_{d-1} \bigcup \{\hat{\zeta}_{d/d''}\}_{d''=d'+1}^{d}, \mathbf{x}) \wedge \textsc{Unchanged-Acts}_{d-1}(\mathcal{W}+\mathcal{U}, \mathbf{x})\Big] \leq \hat{\delta}$$

2. Bound on perturbation of of ℓ_2 norm of the output of layer d. *For all $d = 1, 2, \cdots, D$,*

$$Pr_{\mathcal{U}}\Big[\neg\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \{\hat{\alpha}'_{d}\}, \mathbf{x}) \wedge$$
$$\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \hat{\mathscr{C}}_{d-1} \bigcup \{\hat{\zeta}_{d/d'}\}_{d'=1}^{d}, \mathbf{x}) \wedge \textsc{Unchanged-Acts}_{d-1}(\mathcal{W}+\mathcal{U}, \mathbf{x})\Big] \leq \hat{\delta}$$

3. Bound on perturbation of pre-activations at layer d. *For all $d = 1, 2, \cdots, D$,*

$$Pr_{\mathcal{U}}\Big[\neg\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \{\hat{\gamma}'_{d}\}, \mathbf{x}) \wedge$$
$$\textsc{Pert-Bound}(\mathcal{W}+\mathcal{U}, \hat{\mathscr{C}}_{d-1} \bigcup \{\hat{\zeta}_{d/d'}\}_{d'=1}^{d} \bigcup \{\hat{\alpha}_{d}\}, \mathbf{x}) \wedge \textsc{Unchanged-Acts}_{d-1}(\mathcal{W}+\mathcal{U}, \mathbf{x})\Big] \leq \hat{\delta}$$

Proof. For the most part of this discussion, we will consider a perturbed network where all the hidden units are frozen to be at the same activation state as they were at, before the perturbation (we denote the weights of such a network by $\mathcal{W}[+\mathcal{U}]$ and its output at the dth layer by $f^d(\mathbf{x}; \mathcal{W}[+\mathcal{U}])$). By having the activations states frozen, the Gaussian perturbations propagate as Gaussian perturbations and we can enjoy the well-established properties of the Gaussian even after they propagate.

NOTE. We will prove the parts of the above lemma in an order different from how it is stated; specifically, we will study the perturbation of the Jacobian's row-wise ℓ_2 norm the last, as it is slightly more involved. However, the reason for stating the lemma in that order is to keep it consistent with the order in which our generalization analysis applies parts of it later.

PERTURBATION BOUND ON THE ℓ_2 NORM OF LAYER d. We bound the change in the ℓ_2 norm of the dth layer's output by applying a triangle inequality[5] after splitting it into a sum of vectors. Each summand here (which we define as $\mathbf{v}_{d'}$ for each $d' \leq d$) is the difference in the dth layer output on introducing noise in weight matrix d' after having introduced noise into all the first $d'-1$ weight matrices.

$$\left|\left\|f^d(\mathbf{x}; \mathcal{W}[+\mathcal{U}_d])\right\| - \left\|f^d(\mathbf{x}; \mathcal{W})\right\|\right| \leq \left\|f^d(\mathbf{x}; \mathcal{W}[+\mathcal{U}_d]) - f^d(\mathbf{x}; \mathcal{W})\right\|$$

because the activations are ReLU, we can replace this with the perturbation of the pre-activation

$$\begin{aligned}
&\leq \left\|g^d(\mathbf{x}; \mathcal{W}[+\mathcal{U}_d]) - g^d(\mathbf{x}; \mathcal{W})\right\| \\
&\leq \left\|\sum_{d'=1}^{d}\left(\underbrace{g^d(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d'}]) - g^d(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d'-1}])}_{:=\mathbf{v}_{d'}}\right)\right\| \\
&\leq \sum_{d'=1}^{d}\|\mathbf{v}_{d'}\| = \sum_{d'=1}^{d}\sqrt{\sum_{h} v_{d',h}^2}
\end{aligned} \tag{9}$$

Here, $v_{d',h}$ is the perturbation in the preactivation of hidden unit h on layer d', brought about by perturbation of the d'th weight matrix in a network where the first $d'-1$ weight matrices have already been perturbed.

[5]Specifically, for two vectors $\mathbf{a}, \mathbf{b}$, we have from triangle inequality that $\|\mathbf{b}\| \leq \|\mathbf{a}\| + \|\mathbf{b}-\mathbf{a}\|$ and $\|\mathbf{a}\| \leq \|\mathbf{b}\| + \|\mathbf{a}-\mathbf{b}\|$. As a result of this, we have: $-\|\mathbf{a}-\mathbf{b}\| \leq \|\mathbf{a}\| - \|\mathbf{b}\| \leq \|\mathbf{a}-\mathbf{b}\|$. We use this inequality in our proof.

Now, for each h, we bound $v_{d',h}$ in Equation 9. Since the activations have been frozen we can rewrite each $v_{d',h}$ as the product of the hth row of the unperturbed network's Jacobian d/d' , followed by only the perturbation matrix $U_{d'}$, and then the output of the layer $d'-1$. Concretely, we have[6] [7]:

$$v_{d',h} = \overbrace{J^{d/d'}(\mathbf{x};\mathcal{W})[h]}^{1\times H_{d'}} \underbrace{\overbrace{U_{d'}}^{H_{d'}\times H_{d'-1}} \overbrace{f^{d'-1}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d'-1}])}^{H_{d'-1}\times 1}}_{\text{spherical Gaussian}}$$

What do these random variables $v_{d',h}$ look like? Conditioned on $\mathcal{U}_{d'-1}$, the second part of our expansion of $v_{d',h}$, namely, $U_{d'} f^{d'-1}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d'-1}])$ is a multivariate spherical Gaussian (see Lemma B.2) of the form $\mathcal{N}(0, \sigma^2 \| f^{d'-1}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d'-1}]) \|^2 I)$. As a result, conditioned on $\mathcal{U}_{d'-1}$, $v_{d',h}$ is a univariate Gaussian $\mathcal{N}(0, \sigma^2 \| J^{d/d'}(\mathbf{x};\mathcal{W})[h] \|^2 \| f^{d'-1}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d'-1}]) \|^2)$.

Then, we can apply a standard Gaussian tail bound (see Lemma B.1) to conclude that with probability $1-\hat{\delta}/DH$ over the draws of $U_{d'}$ (conditioned on any $\mathcal{U}_{d'-1}$), $v_{d',h}$ is bounded as:

$$|v_{d',h}| \leq \sigma \left\| J^{d/d'}(\mathbf{x};\mathcal{W})[h] \right\| \| f^{d'-1}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d'-1}]) \| \sqrt{2 \ln \frac{2DH}{\hat{\delta}}}. \tag{10}$$

Then, by a union bound over all the hidden units on layer d, and for each d', we have that with probability $1-\hat{\delta}$, Equation 9 is upper bounded as:

$$\sum_{d'} \sqrt{\sum_{h} v_{d',h}^2} \leq \sum_{d'=1}^{d} \sigma \left\| J^{d/d'}(\mathbf{x};\mathcal{W}) \right\|_F \| f^{d'-1}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d'-1}]) \| \sqrt{2 \ln \frac{2DH}{\hat{\delta}}}. \tag{11}$$

Finally, we prove the probability bound in the lemma statement. To simplify notations, let us denote $\hat{\mathscr{C}}_{d-1} \bigcup \{\hat{\zeta}_{d/d'}\}_{d'=1}^{d}$ by $\hat{\mathscr{C}}_{\text{prev}}$. Furthermore, we will drop redundant symbols in the arguments of the events we have defined. Then, recall that we want to we upper bound the following probability (we ignore the arguments $\mathcal{W}+\mathcal{U}$ and $\mathbf{x}$ for brevity):

$$\Pr\left[(\neg \textsc{Pert-Bound}(\{\hat{\alpha}'_d\})) \wedge \textsc{Pert-Bound}(\hat{\mathscr{C}}_{\text{prev}}) \wedge \textsc{Unchanged-Acts}_{d-1} \right]$$

Recall that Equation 11 is a bound on the perturbation of the ℓ_2 norm of the dth layer's output when the activation states are explicitly frozen. If the perturbation we randomly draw happens to satisfy $\textsc{Unchanged-Acts}_{d-1}$ then this bound holds good even in the case where the activation states are not explicitly frozen. Furthermore, when $\textsc{Pert-Bound}(\hat{\mathscr{C}}_{\text{prev}})$ holds, the bound in Equation 11 can be upper-bounded by $\hat{\alpha}'_d$ as defined in the lemma statement, because under $\textsc{Pert-Bound}(\hat{\mathscr{C}}_{\text{prev}})$, the middle term in Equation 11 can be upper bounded using triangle inequality as $\| f^{d'-1}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d'-1}]) \| \leq \| f^{d'-1}(\mathbf{x};\mathcal{W}) \| + \hat{\alpha}_{d'-1}$. Hence, the event above happens only for the perturbations for which Equation 11 fails and hence we have that the above is upper bounded by $\hat{\delta}$.

PERTURBATION BOUND ON THE PREACTIVATION VALUES OF LAYER d. Following the same analysis as above, the bound we are seeking here is essentially $\max_h \sum_{d'=1}^{d} |v_{d',h}|$. The bound follows similarly from Equation 10.

[6] Below, we have used H_d to denote the number of units on the dth layer (and this equals H for the hidden units and K for the output layer).

[7] Note that the succinct formula below holds good even for the corner case $d'=d$, where the first Jacobian-row term becomes a vector with zeros on all but the hth entry and therefore only the hth row of the perturbation matrix $U_{d'}$ will participate in the expression of $v_{d',h}$.

PERTURBATION BOUND ON THE ℓ_2 NORM OF THE ROWS OF THE JACOBIAN d/d'. We split this term like we did in the previous section, and apply triangle equality as follows:

$$
\begin{aligned}
&\max_h \left| \left\| J^{d/d'}(\mathbf{x}; \mathcal{W})[h] \right\| - \left\| J^{d/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_d])[h] \right\| \right| \\
&\leq \max_h \left\| J^{d/d'}(\mathbf{x}; \mathcal{W})[h] - J^{d/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_d])[h] \right\| \\
&\leq \left\| \sum_{d''=1}^{d} \left(\underbrace{J^{d/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d''}])[h] - J^{d/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d''-1}])[h]}_{:=\mathbf{y}_h^{d''}} \right) \right\| \\
&\leq \max_h \sum_{d''=1}^{d} \left\| \mathbf{y}_h^{d''} \right\| = \max_h \sum_{d''=1}^{d} \sqrt{\sum_{h'} \left(y_{d'',h,h'} \right)^2}
\end{aligned}
\tag{12}
$$

Here, we have defined $\mathbf{y}_h^{d''}$ to be the vector that corresponds to the difference in the hth row of the Jacobian d/d' brought about by perturbing the d''th weight matrix, given that the first $d''-1$ matrices have already been perturbed. We use h to iterate over the units in the dth layer and h' to iterate over the units in the d'th layer.

Now, under the frozen activation states, when we perturb the weight matrices from 1 uptil d', since these matrices are not involved in the Jacobian d/d', fortunately, the Jacobian d/d' is not perturbed (as the set of active weights in d/d' are the same when we perturb $\mathcal{W}$ as $\mathcal{W}[+\mathcal{U}_{d'}]$). So, we will only need to bound $y_{d'',h,h'}$ for $d'' > d'$.

What does the distribution of $y_{d'',h,h'}$ look like for $d'' > d'$? We can expand[8] $y_{d'',h,h'}$ as the product of i) the hth row of the Jacobian d/d'' ii) the perturbation matrix $\mathcal{U}_{d''}$ and iii) the h'th column of the Jacobian $d'/d''-1$ for the perturbed network:

$$
y_{d'',h,h'} = \overbrace{J^{d/d''}(\mathbf{x}; \mathcal{W})[h]}^{1 \times H_{d''}} \underbrace{\overbrace{U_{d''}}^{H_{d''} \times H_{d''-1}} \overbrace{J^{d''-1/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d''-1}])[:, h']}^{H_{d''-1} \times 1}}_{\text{spherical Gaussian}}
$$

Conditioned on $\mathcal{U}_{d''-1}$, the second part of this expansion, namely, $U_{d''} J^{d''-1/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d''-1}])[:, h']$ is a multivariate spherical Gaussian (see Lemma B.2) of the form $\mathcal{N}(0, \sigma^2 \| J^{d''-1/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d''-1}])[:, h'] \|^2 I)$. As a result, conditioned on $\mathcal{U}_{d''-1}$, $y_{d'',h,h'}$ is a univariate Gaussian $\mathcal{N}(0, \sigma^2 \| J^{d/d''}(\mathbf{x}; \mathcal{W})[h] \|^2 \| J^{d''-1/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d''-1}])[:, h'] \|^2)$.

Then, by applying a standard Gaussian tail bound we have that with probability $1 - \frac{\hat{\delta}}{DH^2}$ over the draws of $U_{d''}$ conditioned on $\mathcal{U}_{d''-1}$, each of these quantities is bounded as:

$$
|y_{d'',h,h'}| \leq \sigma \| J^{d/d''}(\mathbf{x}; \mathcal{W})[h] \| \| J^{d''-1/d'}(\mathbf{x}; \mathcal{W}[+\mathcal{U}_{d''-1}])[:, h'] \| \sqrt{2 \ln \frac{DH^2}{\hat{\delta}}}
$$

By a union bound, we then get that with probability $1 - \hat{\delta}$ over the draws of $\mathcal{U}_d$, we can upper bound Equation 12 as:

[8] Again, note that the below succinct formula works even for corner cases like $d'' = d'$ or $d'' = d$.

$$\max_h \sum_{d'=1}^{D} \sqrt{\sum_{h'} \left(y_{d'',h,h'}\right)^2} \leq$$
$$\sum_{d''=d'+1}^{D} \sigma \max_h \left\| J^{d/d''}(\mathbf{x};\mathcal{W})[h] \right\| \left\| J^{d''-1/d'}(\mathbf{x};\mathcal{W}[+\mathcal{U}_{d''-1}]) \right\|_F \sqrt{4 \ln \frac{DH}{\hat{\delta}}}$$

Finally we get the result of the lemma by a similar argument as in the case of the perturbation bound on the output of each layer.

□

F PROOF FOR THEOREM F.1

In this section, we will prove our the generalization bound for the specific case of ReLU based deep networks, that we proved in our main paper.

F.1 NOTATIONS.

To make the presentation of our theorem cleaner, we will set up some notations. First, we define the following positive constants which will correspond to norm bounds on different properties of the network *on the training data*:

1. $\zeta^{\star}_{d/d'}$ for each layer $d = 1, 2, \cdots, D$, and $d' = 1, \cdots, d$ (we will use this to bound ℓ_2 norms of rows in the Jacobians of the network for a training input)
2. $\alpha^{\star}_{d}$, for each hidden layer $d = 1, \cdots, D-1$, (we will use this to bound ℓ_2 norms of the outputs of the layers of the network on a training input)
3. $\gamma^{\star}_{d}$ for each layer $d = 1, \cdots, D$, (we will use this to bound magnitudes of the preactivations values of the network on a training input).

We will also define $\alpha^{\star}_0$ to be an ℓ_2-norm bound on the input $\mathbf{x}$. As we did in the previous section, to simplify our discussion, we will group these constants into a 'set' $\mathscr{C}^{\star}$.

Given these training set related constants, we also define $\mathscr{C}^{\dagger}$ to be the following constants corresponding to weaker norm-bounds related to the *test data*:

1. $\zeta^{\dagger}_{d/d'} := \mathbf{2}\zeta^{\star}_{d/d'}$ for each layer $d = 1, 2, \cdots, D$, and $d' = 1, \cdots, d$ (we will use this to bound ℓ_2 norms of rows in the Jacobians of the network for a test input)
2. $\alpha^{\dagger}_{d} := \mathbf{2}\alpha^{\star}_{d}$, for each hidden layer $d = 1, \cdots, D-1$, (we will use this to bound ℓ_2 norms of the outputs of the layers of the network on a test input)
3. $\gamma^{\dagger}_{d} := \gamma^{\star}_{d}/\mathbf{2}$ for each layer $d = 1, \cdots, D$, (we will use this to bound magnitudes of the preactivations values of the network on a test input).

Note about (abuse of) notation: We reiterate a point about our notation which we also made in Appendix E. We call $\mathscr{C}^{\star}$ and $\mathscr{C}^{\dagger}$ a 'set' to denote a group of related constants by a single symbol. Each element in this set has a particular semantic associated with it, unlike the standard notation of a set.

Now, for any given set of constants $\mathscr{C}$, for a particular weight configuration $\mathcal{W}$, and for a given input $\mathbf{x}$, we define the following event which holds when the network satisfies certain norm-bounds defined by the constants $\mathscr{C}$ (that are favorable for noise-resilience).

Definition F.1. *For a set of constants $\mathscr{C}$, for network parameters $\mathcal{W}$ and for any input $\mathbf{x}$, we define* NORM-BOUND$(\mathcal{W}, \mathscr{C}, \mathbf{x})$ *to be the event that all the following hold good:*

1. *for all $\zeta_{d/d'} \in \mathscr{C}$, $\max_h \left\| J^{d/d'}(\mathbf{x};\mathcal{W})[h] \right\| < \zeta_{d/d'}$ (Rows of Jacobian do not have too large an ℓ_2 norm).*

2. *for all* $\alpha_d \in \mathscr{C}$, $\left\| f^d(\mathbf{x};\mathcal{W}) \right\| < \alpha_d$ *(Output of the layer does not have too large an* ℓ_2 *norm).*

3. *for all* $\gamma_d \in \mathscr{C}$, $\min_h \left| f^d(\mathbf{x};\mathcal{W})[h] \right| > \gamma_d$. *(Pre-activation values are not too small).*

NOTE: (Similar to a note under Definition E.1) A subtle point in the above definition (which we will make use of, to state our theorems) is that if we supply only a subset of $\mathscr{C}$ to the above event, NORM-BOUND$(\mathcal{W}, \cdot, \mathbf{x})$ then it would denote the event that only those subset of properties satsify the respective norm bounds.

F.2 MAIN RESULT

Below, we present our main result for this section, a generalization bound on a class of networks that satisfy certain norm bounds on the *training data*. We provide a more intuitive presentation of these bounds after the proof in Appendix F.4.

Theorem F.1. *Consider a set of positive training-set related constants* $\mathscr{C}^\star$. *Then, for any* $\delta > 0$, *with probability* $1-\delta$ *over the draw of samples from* $\mathcal{D}^m$, *for any* $\mathcal{W}$ *such that, for every* $(\mathbf{x}, y) \in S$, NORM-BOUND$(\mathcal{W}, \mathscr{C}^\star, \mathbf{x})$ *holds, we have that:*

$$
\begin{aligned}
Pr_{(x,y)\sim\mathcal{D}}\left[f(\mathbf{x};\mathcal{W})[y] < \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j] \right] \leq \frac{1}{m} \sum_{(\mathbf{x},y)\in S} \mathbf{1}\left[f(\mathbf{x};\mathcal{W})[y] < \gamma_{class} + \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j] \right] \\
+ 6D^2 \sqrt{\frac{1}{m-1} \cdot \left(2\sum_{d=1}^{D} \|W_d\|^2 \frac{1}{(\sigma^\star)^2} + \ln \frac{6D^2 m}{\delta} \right)} \\
+ \frac{6D^2}{\sqrt{m}-1}
\end{aligned}
$$

where $\frac{1}{\sigma^\star}$ *is* $40\sqrt{H}\sqrt{\ln\left(3D^3H\sqrt{m}\right)} \times \max\{\mathcal{B}_{jac}, \mathcal{B}_{out}, \mathcal{B}_{hidden\text{-}preact}, \mathcal{B}_{output\text{-}preact}\}$; *here we define:*

$$
\begin{aligned}
\mathcal{B}_{jac} &:= \max_{1\leq d'<d\leq D} \frac{\sum_{d''=d'+1}^{d} \zeta^\star_{d/d''} \zeta^\star_{d''-1/d'}}{\zeta^\star_{d/d'}}, \\
\mathcal{B}_{out} &:= \max_{d\leq D-1} \frac{\sum_{d'=1}^{d} \zeta^\star_{d/d'} \alpha^\star_{d'-1}}{\alpha^\star_d} \\
\mathcal{B}_{hidden\text{-}preact} &:= \max_{d\leq D-1} \frac{\sum_{d'=1}^{d} \zeta^\star_{d/d'} \alpha^\star_{d'-1}}{\sqrt{H}\gamma^\star_d} \\
\mathcal{B}_{output\text{-}preact} &:= \frac{2\sum_{d'=1}^{D} \zeta^\star_{D/d'} \alpha^\star_{d'-1}}{\sqrt{H}\gamma_{class}}
\end{aligned}
$$

REMARK Observe that the above generalization bound applies to a specific class of $\mathcal{W}$ that satisfies certain norm bounds on the training data. One could extend the bound to all values of $\mathcal{W}$ for theoretical purposes, by 'covering' all possible values of the constants defined in the norm bounds (see Neyshabur et al. (2018) for example) at the cost of extra factors. However, from a practical point of view, the statement in this form is sufficient because we are interested in a generalization bound for a small class of networks that SGD restricts itself to, and not all possible networks. In practice, to apply this bound, one could run SGD on many random draws of the training dataset and identify training-set related constants $\mathscr{C}^\star$ such that the network learned from all these random draws satisfy the norm bounds defined by $\mathscr{C}^\star$.

F.3 PROOF FOR THEOREM F.1

Proof. To apply the framework, we will have to first define the abstract functions T_r and quantities $\Delta_r^\star$ used in Theorem 3.1 . These functions will be composed of the following properties corresponding to different values of d, d' and h (the ρ functions described in the main framework): $\left\|J^{d/d'}(\mathbf{x};\mathcal{W})[h]\right\|$, $\left\|f^d(\mathbf{x};\mathcal{W})\right\|$ and $f^d(\mathbf{x};\mathcal{W})[h]$.

Definition F.2. *We define the following set of functions $\mathscr{T}^\star$, and the corresponding set of margin values $\mathscr{C}^\vartriangle$:*

$$T_{d/d',jac}(\mathcal{W},\mathbf{x},y) := 2\zeta^\star_{d/d'} - \max_h \left\|J^{d/d'}(\mathbf{x};\mathcal{W})[h]\right\|, \quad \zeta^\vartriangle_{d/d'} := \zeta^\star_{d/d'} \qquad \textit{for } d = 1,\cdots,D \;\; d' = 1,\cdots,d$$

$$T_{d,output}(\mathcal{W},\mathbf{x},y) := 2\alpha^\star_d - \left\|f^d(\mathbf{x};\mathcal{W})\right\|, \qquad \alpha^\vartriangle_d := \alpha^\star_d \qquad \textit{for } d = 1,2,\cdots,D-1$$

$$T_{d,preact}(\mathcal{W},\mathbf{x},y) := \min_h \left|f^d(\mathbf{x};\mathcal{W})[h]\right| - \frac{\gamma^\star_d}{2}, \qquad \gamma^\vartriangle_d := \frac{\gamma^\star_d}{2} \qquad \textit{for } d = 1,2,\cdots,D-1$$

and for the output layer D:

$$T_{D,preact}(\mathcal{W},\mathbf{x},y) := 0.5(f(\mathbf{x};\mathcal{W})[y] - \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j]) \qquad \gamma^\vartriangle_D := \frac{\gamma_{class}}{2}$$

We will use the notation $\mathscr{C}^\vartriangle/2$ to denote the values in the margin set divided by 2.

Note that the above functions satisfy the constraint Equation 1 as required by our abstract framework.

ON THE CHOICE OF THE ABOVE FUNCTIONS AND MARGIN VALUES. Below, we justify our choice of these functions and margins, by describing how inequalities of the form $T_r(\mathcal{W},\mathbf{x},y) < \Delta + r^\star$ or $T_r(\mathcal{W},\mathbf{x},y) < 0$ convert to norm-bounds on the input-dependent properties of the network.

When we apply the above quantities of T and $\Delta^\star$ in Theorem C.1, the guarantees will be in terms of the test loss $\Pr_{(x,y)\sim\mathcal{D}}[T(\mathcal{W},\mathbf{x}) < 0]$ and train loss $\Pr_{(x,y)\sim S}[T(\mathcal{W},\mathbf{x}) < \Delta]$ which would be equivalent to the norm-bounds shown in the table below. Observe that the second column (the 0-margin error column) corresponds to the event that a datapoint does not satisfy the weaker norm-bound defined by test-set constants $\mathscr{C}^\dagger$ (which was defined in the beginning of this section). On the other hand, the last column (the Δ-margin error column) corresponds to the event that a datapoint does not satisfy the norm-bounds defined by the train-set constants $\mathscr{C}^\star$.

Table 1: Margin-based test and train errors.

T, Δ	$T(\mathcal{W},\mathbf{x}) < 0$	$T(\mathcal{W},\mathbf{x}) < \Delta$
$T_{d/d',\text{jac}}, \zeta^\vartriangle_{d/d'}$	$\max_h \left\|J^{d/d'}(\mathbf{x};\mathcal{W})[h]\right\| > 2\zeta^\star_{d/d'} = \zeta^\dagger_{d/d'}$	$\max_h \left\|J^{d/d'}(\mathbf{x};\mathcal{W})[h]\right\| > \zeta^\star_{d/d'}$
$T_{d,\text{output}}, \alpha^\vartriangle_d$	$\left\|f^d(\mathbf{x};\mathcal{W})\right\| > 2\alpha^\star_d = \alpha^\dagger_d$	$\left\|f^d(\mathbf{x};\mathcal{W})\right\| > \alpha^\star_d$
$T_{d,\text{preact}}, \gamma^\vartriangle_d$ (for $d < D$)	$\min_h \left\|f^d(\mathbf{x};\mathcal{W})[h]\right\| < \gamma^\star_d/2 = \gamma^\dagger_d$	$\min_h \left\|f^d(\mathbf{x};\mathcal{W})[h]\right\| < \gamma^\star_d$
$T_{D,\text{preact}}, \gamma^\vartriangle_D$	$f(\mathbf{x};\mathcal{W})[y] < \max_{j\neq y} f(\mathbf{x};\mathcal{W})[j]$	$f(\mathbf{x};\mathcal{W})[y] <$ $\max_{j\neq y} f(\mathbf{x};\mathcal{W})[j] + \gamma_{\text{class}}$

Now to apply the abstract generalization bound in Theorem 3.1, recall that we need to come up with an ordering of the functions above such that we can realize an inequality of the form given in Equation 2. To this end, we induce an ordering based on the perturbation bounds presented in Lemma E.1.

The order in which we traverse the properties is as follows. We will go from layer 1 uptil D; within each layer we will first deal with the Jacobians d/d, $d/d-1$, so on until $d/1$ in that order[9] . Then we will deal with the ℓ_2 norm of the output of the layer and the pre-activation values (the order between these two will not actually matter). Note that $|\mathscr{T}^\star| \leq 3D^2$ since there are at most $D+2$ properties in each layer.

Definition F.3. ***Sequence 1:***

$$\begin{array}{lccc} & T_{1/1,jac}, & T_{1,output}, & T_{1,preact}, \\ \textit{followed by} & T_{2/1,jac}, & T_{2,output}, & T_{2,preact}, \\ & \vdots & \vdots & \vdots \\ \textit{followed by} & T_{D/D,jac}, \cdots, T_{D/1,jac}, & T_{D,output} & T_{D,preact}. \end{array}$$

The next step in our proof is to show that inequalities of the form presented in Equation 2 hold good. To do this, we instantiate Lemma E.1 with $\sigma = \sigma^\star$ as in Theorem F.1, $\hat{\delta} = \frac{1}{3D^2\sqrt{m}}$ and $\hat{\mathscr{C}} = \mathscr{C}^\vartriangle/2$. Then, it can be verified that the values of the tolerance parameters $\hat{\mathscr{C}}'$ in Lemma E.1 can be upper bounded by the corresponding value in $\mathscr{C}^\vartriangle/2$. Succinctly, we say:

$$\hat{\mathscr{C}}' \leq \mathscr{C}^\vartriangle/2 \tag{13}$$

Given this, we will focus on showing that an inequality of the form Equation 2 holds for $T_{d/d',\text{jac}}$. A similar approach would apply for the other properties. For brevity, we will define $\mathscr{C}^\dagger_{\text{prev}} = \mathscr{C}^\dagger_{d-1} \bigcup \{\zeta^\dagger_{d/d''}\}^d_{d''=d'+1}$ and $\hat{\mathscr{C}}_{\text{prev}} = \hat{\mathscr{C}}_{d-1} \bigcup \{\hat{\zeta}_{d/d''}\}^d_{d''=d'+1}$

First, we note that the precondition for Equation 2 translates to $\textsf{NORM-BOUND}(\mathcal{W}+\mathcal{U}, \mathscr{C}^\dagger_{\text{prev}}, \mathbf{x})$ (as shown in Table 1). Given this condition, we first argue that if $\textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \hat{\mathscr{C}}_{\text{prev}}, \mathbf{x})$ holds, then so does $\textsf{UNCHANGED-ACTS}_{d-1}(\mathcal{W}+\mathcal{U}, \mathbf{x})$. This is because, the event $\textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \hat{\mathscr{C}}_{\text{prev}}, \mathbf{x})$ implies that the pre-activation values of layer $d-1$ suffer a perturbation of at most $\hat{\gamma}_{d-1}$, which from Equation 13 is at most $\gamma^\vartriangle_{d-1}/2 = \gamma^\star_{d-1}/4$ i.e., $\max_h |f^{d-1}(\mathbf{x}; \mathcal{W}+\mathcal{U})[h] - f^{d-1}(\mathbf{x}; \mathcal{W})[h]| \leq \gamma^\star_{d-1}/4$. However, since $\textsf{NORM-BOUND}(\mathcal{W}, \mathscr{C}^\dagger_{\text{prev}}, \mathbf{x})$ holds, we have that the preactivation values of this layer have a magnitude of at least $\gamma^\dagger_{d-1} = \gamma^\star_{d-1}/2$ before perturbation i.e., $\min_h |f^{d-1}(\mathbf{x}; \mathcal{W})[h]| \geq \gamma^\star_{d-1}/2$. From these two equations, we have that the hidden units even at layer $d-1$ of the network do not change their activation state (i.e., the sign of the pre-activation does not change) under this perturbation. We can similarly argue for the layers below $d-1$, thus proving that $\textsf{UNCHANGED-ACTS}_{d-1}(\mathcal{W}+\mathcal{U}, \mathbf{x})$ holds under $\textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \hat{\mathscr{C}}_{\text{prev}}, \mathbf{x})$.

Then, observe that the guarantee from Lemma E.1 boils down to:

$$Pr_{\mathcal{U}}\Big[\neg\textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \{\hat{\zeta}'_{d/d'}\}, \mathbf{x}) \wedge \textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \mathscr{C}^\vartriangle_{\text{prev}}/2, \mathbf{x})\Big] \leq \frac{1}{3D^2\sqrt{m}}$$

Now, from Equation 13 we have that, if $\textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \{\hat{\zeta}'_{d/d'}\}, \mathbf{x})$ holds, then $\textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \{\zeta^\vartriangle_{d/d'}/2\}, \mathbf{x})$ holds. This implies that we can replace the first term as below:

$$Pr_{\mathcal{U}}\Big[\neg\textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \{\zeta^\vartriangle_{d/d'}/2\}, \mathbf{x}) \wedge \textsf{PERT-BOUND}(\mathcal{W}+\mathcal{U}, \mathscr{C}^\vartriangle_{\text{prev}}/2, \mathbf{x})\Big] \leq \frac{1}{3D^2\sqrt{m}}$$

This inequality is precisely of the same form as Lemma E.1. Through identical arguments for the other functions, our generalization bound holds.

[9]Note that the Jacobian for d/d is nothing but an identity matrix regardless of the input datapoint; thus we do not need any generalization analysis to bound its value on a test datapoint. But to make the presentation simpler in some ways, we will include it in our generalization analysis.

Note that in the final bound, we use Lemma B.3 to simplify the KL-divergence term between the posterior centered at $\mathcal{W}$ and the prior at the random initialization $\mathcal{Z}$.

□

F.4 INTERPRETING THE BOUNDS.

Below we show empirically how the terms $\mathcal{B}_{\text{jac}}, \mathcal{B}_{\text{out}}, \mathcal{B}_{\text{hidden-preact}}, \mathcal{B}_{\text{output-preact}}$ vary with the width of a network after fixing the depth D of the network to be 6. For all examples, we use SGD with learning rate 0.01 to train the network on the MNIST dataset by minimizing cross entropy loss until all datapoints are classified by a margin of at least 10.

We see in Figures 3 that the first two terms $\mathcal{B}_{\text{jac}}, \mathcal{B}_{\text{out}}$ are width-independent; on the other hand, $\mathcal{B}_{\text{output-preact}}$ decreases with width as $1/\sqrt{H}$ and this is because the numerator of this term is width-independent, while the denominator of this term, which is $\sqrt{H}\gamma_{\text{class}}$, increases with width (we have chosen γ_{class} to be a constant of 10 in all our experiments). Furthermore, in Figure 2 we see that all these three terms show an increase with depth that is only nearly linear or mildly exponential – these terms certainly do not grow as rapidly as the product of the spectral norms. This is because, in practice the terms $\zeta^{\star}_{d/d'}$ and $\alpha^{\star}_{d}$ are width-independent, and do not grow as rapidly as the product of the spectral norms.

As far as the term $\mathcal{B}_{\text{hidden-preact}}$ is concerned, recall from our discussion in the main paper that the minimum pre-activation value $\gamma^{\star}_{d}$ of the network tends to be quite small in practice, and therefore the term $\mathcal{B}_{\text{hidden-preact}}$ can be quite large; furthermore, this term exhibits considerable variance across different widths/depths and different training runs. As we noted, for future work, it would be important to understand how the dependence of the bound on this term can be better improved.

Theoretically speaking, the best scenario for $\mathcal{B}_{\text{hidden-preact}}$ can be realized when the preactivation values of each layer (which has a total ℓ_2 norm that is width-independent in practice) are equally spread out across the hidden units. Then we will have that the smallest pre-activation value is guaranteed to be as large as $\Omega(1/\sqrt{H})$. Then, $\mathcal{B}_{\text{hidden-preact}}$ would behave like the other terms, in terms of its width and depth dependence. In fact, we simulate this in practice, by plotting a redefined version of $\mathcal{B}_{\text{hidden-preact}}$ where instead of plugging in the minimum pre-activation value, we plug in the *median* pre-activation value (that is, for each datapoint we compute the median pre-activation value for that layer, and then find the lowest value of this median across all datapoints). We observe that since the median pre-activation value is only as small as $\Omega(1/\sqrt{H})$ in practice, the 'median-based-$\mathcal{B}_{\text{hidden-preact}}$' has similar variation with respect to the width and the depth as the other terms $\mathcal{B}_{\text{jac}}, \mathcal{B}_{\text{out}}, \mathcal{B}_{\text{output-preact}}$ and is also quite small. See Figure 4 and Figure 2 for how the original $\mathcal{B}_{\text{hidden-preact}}$ and the 'median-based-$\mathcal{B}_{\text{hidden-preact}}$' vary with width and depth.

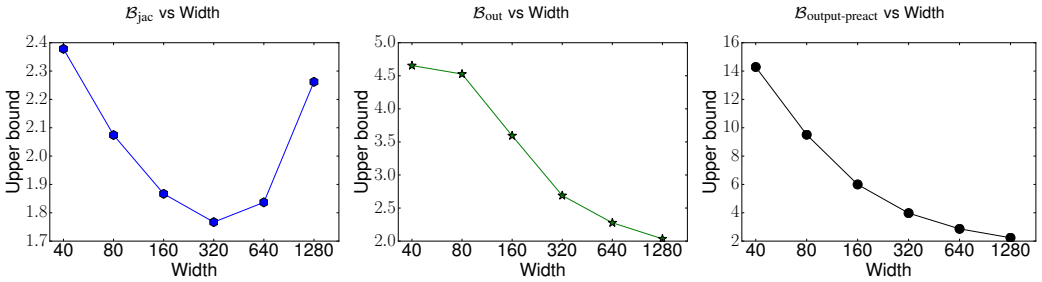

Figure 3: In the above figure, we plot the values of $\mathcal{B}_{\text{jac}}$ (**left**) $\mathcal{B}_{\text{out}}$ (**middle**) $\mathcal{B}_{\text{output-preact}}$ (**right**) computed on a network of depth 6 trained on the MNIST dataset, for varying widths. Observe that the first two values do not increase with width. Furthermore, the last value, $\mathcal{B}_{\text{output-preact}}$ shows a strong decrease with width as $1/\sqrt{H}$ (for every doubling of the width, the value drops by $\sqrt{2}$.)

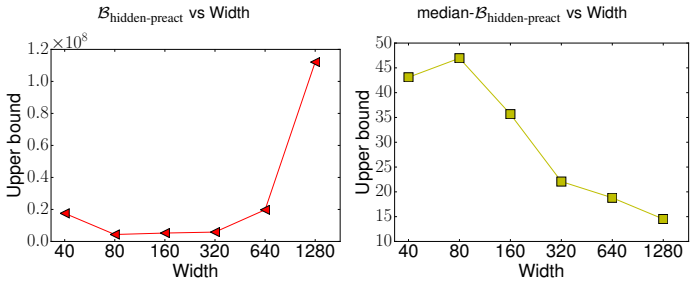

Figure 4: In the above figure, we plot the values of $\mathcal{B}_{\text{hidden-preact}}$ (**left**) and 'median-based'-$\mathcal{B}_{\text{hidden-preact}}$ (**right**) computed on a network of depth 6, trained on a subset of the MNIST dataset, for varying widths. While it is hard to quantify how $\mathcal{B}_{\text{hidden-preact}}$ varies, observe that the latter quantity decays with width.