Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan [1] **J. Zico Kolter** [1 2]

Abstract

We cast doubt on the power of uniform convergence-based generalization bounds to provide a complete picture of why overparameterized deep networks generalize well. While it is well-known that many existing bounds are numerically large, through a variety of experiments, we first bring to light another crucial and more concerning aspect of these bounds: in practice, these bounds can *increase* with the dataset size. Guided by our observations, we then show how uniform convergence could provably break down even in a simple setup that preserves the key elements of deep learning: we present a *noisy* algorithm that learns a mildly *overparameterized* linear classifier such that uniform convergence cannot "explain generalization," even if we take into account implicit regularization *to the fullest extent possible*. More precisely, even if we consider only the set of classifiers output by the algorithm that have test errors less than some small ϵ, applying (two-sided) uniform convergence on this set of classifiers yields a generalization guarantee that is larger than $1-\epsilon$ and is therefore nearly vacuous.

1. Introduction

Explaining why overparameterized deep networks generalize well (Neyshabur et al., 2015a; Zhang et al., 2017) has become an important open question in deep learning. How is it possible that networks that can be trained to comfortably fit (i.e., memorize) randomly labeled data, can also be trained to fit real training data while generalizing to unseen data? This called for a "rethinking" of conventional, algorithm-independent techniques to explain generalization. Specifically, it was argued that learning-theoretic approaches must be reformed by identifying and incorporating the implicit bias/regularization of stochastic gradient descent (SGD) (Brutzkus et al., 2018; Soudry et al., 2018; Neyshabur et al., 2017). Subsequently, there has been a slew of refined generalization bounds for deep networks – all based on uniform convergence, the most widely used tool in learning theory. The ultimate goal of these endeavors is to derive bounds on the generalization error that (a) are small, ideally non-vacuous (i.e., < 1), (b) reflect the same width/depth dependence as the generalization error (e.g., become smaller or at least stay constant with increasing width, as has been surprisingly observed in practice), (c) apply to the network learned by SGD rather than to a network modified through compression or stochasticization and (d) increase with the proportion of randomly flipped training labels.

However, all existing bounds fail to meet many of the above criteria – and even a crucial fifth criterion that we shortly introduce. While most bounds (Neyshabur et al., 2015b; Bartlett et al., 2017; Golowich et al., 2018; Neyshabur et al., 2018; Nagarajan & Kolter, 2019; Neyshabur et al., 2019) apply to the original network, they are neither numerically small for realistic dataset sizes, nor exhibit the desired width/depth dependencies (in fact, these bounds grow exponentially with the depth). The ones that are small for realistic dataset sizes hold either only on a compressed network (Arora et al., 2018) or a stochastic network (Langford & Caruana, 2001) or a network that has been further modified via optimization or more than one of the above (Dziugaite & Roy, 2017; Zhou et al., 2019). Extending these bounds to the original network is understood to be highly non-trivial (Nagarajan & Kolter, 2019).

In our paper, we bring to light another fundamental issue with these uniform convergence based bounds. We empirically show that existing bounds violate a natural but largely overlooked criterion for explaining generalization: (e) the bounds should decrease with the dataset size at the same rate as the generalization error. In fact, we observe that these bounds can even *increase* with dataset size, which is arguably a more concerning observation than the fact that these bounds are large for a specific dataset size.

Provoked by the seemingly insurmountable hurdles towards satisfying all the five necessary criteria, we take a step back and examine how the underlying technique of uniform convergence may itself be fundamentally limited in the overparameterized regime. Specifically, through an example, we

[1]Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA [2]Bosch Center for Artificial Intelligence, Pittsburgh, Pennsylvania, USA. Correspondence to: Vaishnavh Nagarajan <vaishnavh@cs.cmu.edu>, Zico J. Kolter <zkolter@cs.cmu.edu>.

show that uniform convergence can *provably* fail to explain generalization even for a simple case: a mildly overparameterized linear classifier (i.e., the number of parameters is linear in terms of the training set size and not exponential) under a noisy learning algorithm (i.e., many of the learned parameters are just noise). Notably, we will see that this example preserves our main empirical findings in deep learning, which also end up playing a crucial role in our theoretical analysis. While existing uniform convergence bounds provide partial explanations about generalization in SGD-trained overparameterized deep networks, through our arguments, we question their potential to explain the phenomenon fully.

1.1. Our contributions

In the first part, we show that certain norms – such as the distance from initialization – that occur in recent generalization bounds for ReLU networks, *increase* polynomially with the number of training examples m in practice. We then show that as a result, the overall generalization bounds do not reflect the same dependence on m as the actual test error, violating criterion (e); for sufficiently small batch sizes, these bounds even grow with the number of examples. This observation uncovers a new conceptual gap in our understanding of the puzzle, by pointing towards a source of vacuity in these bounds, unrelated to parameter count.

In the second part, we present a noisy algorithm that learns a mildly overparameterized linear classifier, and is consistent with our empirical observations about how norms such as distance from initialization grow with m. We then show that there are linearly separable distributions of the underlying data on which our learner generalizes well and yet, applying *any* two-sided uniform convergence bound would yield a (nearly) vacuous generalization bound.

Notably, this holds even if we "aggressively" take implicit regularization into account. That is, recall that for a hypothesis class, two-sided uniform convergence demands that for most draws of a dataset S of m training datapoints from an underlying data distribution $\mathcal{D}$, and *for every hypothesis in the class* (and not just the hypothesis learned by the given algorithm $\mathcal{A}$ on S), the expected error on $\mathcal{D}$ and the empirical error on S must be close to each other. Typically, one can get stronger uniform convergence bounds by considering a smaller hypothesis class that excludes extraneous hypotheses never picked by $\mathcal{A}$ under $\mathcal{D}$. In our setup, even if we apply uniform convergence on the set of *only those hypotheses picked by the algorithm whose test errors are all negligible ($\leq \epsilon$ for some small $\epsilon > 0$)*, one can get no better than a nearly vacuous bound on the generalization error (that is $\geq 1 - \epsilon$). While nearly all existing techniques are based on two-sided uniform convergence, we also show that even PAC-Bayesian bounds, which are typically presented only as one-sided convergence, also boil down to nearly vacuous guarantees.

Our proof crucially relies on i) our empirical observations in deep networks (regarding the norms like distance from initialization) that is captured by our theoretical setup, and ii) the noise in the learned weights. Essentially, we will see that uniform convergence fails to "explain away" this noise under overparametrization. Since generalization in deep networks is known to be aided by noise during training, the fact that uniform convergence can fail in arguably the simplest possible noisy setup – a mildly overparameterized linear classifier – calls into question the ability of using uniform convergence to fully explain generalization in deep learning.

2. Related Work

Neural network bounds. Besides the bounds mentioned in the introduction, considerable progress has been made in deriving width-independent bounds for two-layer ReLU networks – although under specific settings different from our empirical setup. The bound in Brutzkus et al. (2018) does not apply to ReLU networks and the distance from initialization does not grow with training set size m. Li & Liang (2018) conduct an intricate analysis, but rely on a carefully curated, sufficiently small learning rate ($\approx \mathcal{O}(1/m^{1.2})$) and large batch size ($\approx \Omega(\sqrt{m})$). Hence, the resulting bound cannot describe how generalization varies with training set size alone, with everything else fixed. A similar analysis in Allen-Zhu et al. (2018) requires fixing the learning rate to be inversely proportional to width. Their bound decreases only as $\Omega(1/m^{0.16})$, although, the actual generalization error is typically as small as $\mathcal{O}(1/m^{0.43})$.

It is intriguing that on one hand, generalization is aided by larger learning rates and smaller batch sizes (Jastrzebski et al., 2018; Hoffer et al., 2017; Keskar et al., 2017) due to increased noise in SGD. On the other, as evident from above, theoretical analyses benefit from the opposite; Allen-Zhu et al. (2018) even explicitly regularize SGD for their three-layer-network result to help "forget false information" gathered by SGD's haphazard search. In other words, it seems that *noise aids generalization, yet hinders attempts at explaining generalization*. We formally explain this paradox by arguing how such "false information" could provably impair uniform convergence, without affecting generalization.

Weaknesses of Uniform Convergence. Uniform convergence is said to provide vacuous bounds for complex classifiers like k-nearest neighbors because these have infinite VC-dimension, motivating the need for stability based generalization bounds for these algorithms (Rogers & Wagner, 1978; Bousquet & Elisseeff, 2002). However, this does not

rule out the ability of uniform convergence to explain generalization when applied to a space of hypotheses that is aggressively pruned for a given algorithm and data distribution. Our result is thus stronger as it proves that uniform convergence may break down even under such careful application, even for a much simpler model (a linear classifier).

Learnability and Uniform Convergence. Prior works have focused on understanding uniform convergence for *learnability of learning problems*. Roughly speaking, learnability is a strict notion that does not have to hold even though an algorithm may generalize well for simple distributions (see Appendix G for a more technical discussion). In fact, Vapnik & Chervonenkis (1971) showed in their seminal paper that uniform convergence is *equivalent* to learnability in binary classification problems.

Shalev-Shwartz et al. (2010) prove that uniform convergence is not necessary for learnability of a *general learning problem*, which includes a larger class of problems besides binary classification; thus, they rely on an example of a stochastic optimization problem. Their result is an orthogonal claim to ours as their result does not imply that uniform convergence may not be able to *explain generalization* of a particular algorithm for simple distributions in binary classification problems. It is also worth noting that their result requires an extreme level of overparameterization ($\Omega(2^m)$ parameters), while we require only mild overparameterization ($\Omega(m)$).

3. Experiments

In this section, we present the first part of our contribution. As stated in criterion (e) in the introduction, a fundamental requirement from a generalization bound – however numerically large the bound may be – is that it should vary inversely with the size of the training dataset size (m) like the observed generalization error. Such a requirement is satisfied even by standard parameter-count-based VC-dimension bounds – like $\mathcal{O}(dh/\sqrt{m})$ for depth d, width h ReLU networks (Harvey et al., 2017). However, we now show that recent efforts to replace the parameter-count-dependent terms in the numerator with seemingly innocuous norm-based quantities that are parameter-count-independent have also inadvertently introduced training-size-count dependencies in the numerator – contributing to the vacuity of bounds. With these dependencies, the generalization bounds even increase with training dataset size for small batch sizes.

We present our main observations for fully connected networks of depth $d = 5$, width $h = 1024$ trained on the MNIST dataset. We use SGD with learning rate 0.1 and batch size 1 to minimize cross-entropy loss until 99% of the training data are classified correctly by a margin of at least $\gamma^\star = 10$ i.e., if we denote by $f(\mathbf{x})[y]$ the real-valued logit output (i.e., pre-softmax) on class y for an input $\mathbf{x}$, we ensure that for 99% of the data $(\mathbf{x}, y)$, the *margin* $\Gamma(f(\mathbf{x}), y) := f(\mathbf{x})[y] - \max_{y' \neq y} f(\mathbf{x})[y']$ for that input is at least $\gamma^\star$. We emphasize that, from the perspective of generalization guarantees, this stopping criterion helps standardize training across different hyperparameter settings, including different values of m (Neyshabur et al., 2017). For this particular stopping criterion, the test error decreases with training set size m as $1/m^{0.43}$ as seen in Figure 1 bottom-left; however, the story is starkly different for the generalization bounds. While the bounds might show better m-dependence for other settings (indeed, for larger batches, we show that the bounds behave better), the egregious break down of these bounds in this setting (and many other hyperparameter settings as presented in Appendix A) implies fundamental issues with the bounds themselves.

3.1. Norms grow with training set size m

Before we examine the overall bounds, we first focus on two quantities that recur in the numerator of many recent bounds: the ℓ_2 distance of the weights from their initialization (Dziugaite & Roy, 2017; Nagarajan & Kolter, 2017) and the product of spectral norms of the weight matrices of the network (Neyshabur et al., 2018; Bartlett et al., 2017). We observe in Figure 1 (top left and right, blue lines) that both these quantities grow at a polynomial rate with m: the former at the rate of at least $m^{0.4}$ and the latter at a rate of m. Our observation is a follow-up to Nagarajan & Kolter (2017) who argued that while distance of the parameters from the origin (i.e., the ℓ_2 norm of the parameters) grows as $\Omega(\sqrt{h})$, the distance of the learned parameters from initialization is width-independent (and even decreases with width); hence, incorporating the initialization would improve generalization bounds by a $\Omega(\sqrt{h})$ factor. However, we argue that, even though distance from initialization would help explain generalization better in terms of width, it conspicuously fails to help explain generalization in terms of its dependence of m (and so does distance from origin as we show in Appendix Figure 5). [1]

Width of the explored space. We also examine a new alternative to distance from initialization. Consider the *explored space* of SGD: the set of parameters learned across different dataset draws for a fixed initialization. If this space was contained in a ball of m-independent radius, one could then explain generalization better by replacing the distance from initialization with the distance from the center of this ball in existing bounds. Unfortunately, we rule out this possibility, as we observe that the distance between the weights learned on two different datasets shows identical behavior with respect to m as distance from initialization

[1] It may be tempting to think that the above observations are peculiar to the cross-entropy loss for which the optimization algorithm diverges to infinity. But we observe that even for the squared error loss (Appendix A) where the optimization procedure does not diverge to infinity, distance from initialization grows with m.

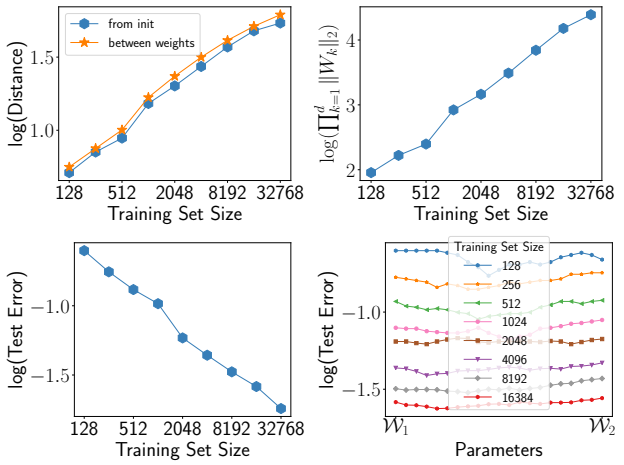

Figure 1. In the **top-left** figure, we plot i) ℓ_2 the distance of the network from the initialization and ii) the ℓ_2 distance between two weights $\mathcal{W}_1, \mathcal{W}_2$ learned on two random draws of training data starting from the same initialization. Both these quantities grow as $\Omega(m^{0.42})$. In the **top-right** figure we plot the product of spectral norms of the weights matrices $W_1, W_2, \ldots, W_d$ and observe that it grows as fast as $\Omega(m)$ for $d = 5$. See Figure 4 in the appendix, for a layer-by-layer plot of these terms. In the **bottom-left** figure, we plot the actual test error which decreases with sample size as $\mathcal{O}(1/m^{0.43})$. Note that we have presented log-log plots and the exponent of m can be recovered from the slope of these plots. In the **bottom-right** plot, we plot the test errors of the networks that lie on the straight line between $\mathcal{W}_1$ and $\mathcal{W}_2$.

(see Figure 1, top-left, orange line).

We also relate this observation to the popular idea of "flat minima". Interestingly, Figure 1 (bottom right) demonstrates that walking linearly from the weights learned on one dataset draw to that on another draw (from the same initialization) preserves the test error. Note that although a similar observation was made in Dräxler et al. (2018); Garipov et al. (2018), they show the existence of *non-linear* paths of good solutions between parameters learned from *different* initializations. Our observation on the other hand implies that for a fixed initialization, SGD explores the *same* basin in the test loss minimum across different training sets. As discussed earlier, this explored basin/space has larger ℓ_2-width for larger m giving rise to a paradox (similar to the paradoxical role of noise noted in Section 2): on one hand, wider minima are believed to result in, or at least correlate with better generalization (Hochreiter & Schmidhuber, 1997; Hinton & van Camp, 1993; Keskar et al., 2017), but on the other, a larger ℓ_2-width of the explored space results in larger uniform convergence bounds, making it harder to explain generalization.

3.2. The bounds grow with training set size m.

We now turn to evaluating existing guarantees from Neyshabur et al. (2018); Bartlett et al. (2017). As we note later, our observations apply to many other bounds too. Let $W_1, \ldots, W_d$ be the weights of the learned network (with W_1 being the weights adjacent to the inputs), $Z_1, \ldots, Z_d$ the random initialization, $\mathcal{D}$ the true data distribution and S the training dataset. For all inputs $\mathbf{x}$, let $\|\mathbf{x}\|_2 \leq B$. Let $\|\cdot\|_2, \|\cdot\|_F, \|\|_{2,1}$ denote the spectral norm, the frobenius norm and the matrix $(2,1)$-norm respectively; let $\mathbf{1}[\cdot]$ be the indicator function. For any constant γ, the generalization guarantee is written as follows, ignoring log factors:

$$Pr_{\mathcal{D}}[\Gamma(f(\mathbf{x}), y) \leq 0] \leq \frac{1}{m} \sum_{(x,y)\in S} \mathbf{1}[\Gamma(f(\mathbf{x}), y) \leq \gamma] + \text{generalization error bound} \quad (1)$$

where the generalization error bound in Neyshabur et al. (2018) is:

$$\mathcal{O}\left(\frac{B}{\gamma\sqrt{m}} \cdot d\sqrt{h} \prod_{k=1}^{d} \|W_k\|_2 \sqrt{\sum_{k=1}^{d} \frac{\|W_k - Z_k\|_F^2}{\|W_k\|_2^2}}\right)$$

and the bound in Bartlett et al. (2017) is:

$$\mathcal{O}\left(\frac{B}{\gamma\sqrt{m}} \prod_{k=1}^{d} \|W_k\|_2 \left(\sum_{k=1}^{d} \left(\frac{\|W_k - Z_k\|_{2,1}}{\|W_k\|_2}\right)^{2/3}\right)^{3/2}\right)$$

In our experiments, since we train the networks to fit at least 99% of the datapoints with a margin of 10, in the above bounds, we set $\gamma = 10$ so that the first margin-based train error term in the right hand side of Equation 1 becomes 0.01. We then plot in Figure 2, the generalization error bounds above and observe that all these bounds *grow with the sample size* m as $\Omega(m^{0.68})$, thanks to the fact that the terms in the numerator of these bounds grow with m.

Even a relaxed notion of margin does not address the m-dependency. One might hope that networks that travel farther distances from initialization, and have larger spectral norms of the weight matrices, might have a proportionally larger margin on a subset of the training dataset; then we could choose a larger value of γ in Equation 1 to achieve a smaller generalization error bound. We consider this possibility by computing the median margin of the network over the training set (instead of the 1%-percentile'th margin) and substituting this in the second term in the right hand side of the guarantee in Equation 1. By doing this, the first margin-based train error term in the right hand side of Equation 1 would simplify to 0.5 (as half the training data are misclassified by this large margin). Thereby we already forgo an explanation of half of the generalization behavior. At least

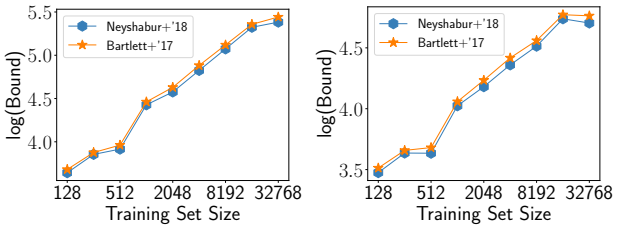

Figure 2. In the **left** figure, we plot the bounds from (Neyshabur et al., 2018; Bartlett et al., 2017) taking into account the random initialization. Both these bounds grow as $\Omega(m^{0.68})$. In the **right** plot, we plot the bounds after setting γ to be the median margin on the training data – these bounds grow as $\Omega(m^{0.48})$.

we could hope that the second term no longer grows with m. Unfortunately, we observe in Figure 2 that the bounds still grow with m. This is because, as will demonstrate in the appendix, the median margin only grows as $m^{0.2}$ and this is insufficient to cancel out the sample-size-dependencies in the numerator of these bounds.

Finally, we defer experiments conducted for other varied settings, and the neural network bound from Neyshabur et al. (2019), to Appendix A. Notably, the m-dependencies observed in this discussion are more pronounced with larger depth, and less pronounced with larger batch sizes. Unfortunately, this also implies that these bounds also do not reflect, even roughly, the true dependency on the batch size: even though the generalization error decreases slightly with decreasing batch size (at least for our chosen stopping criterion, with all other hyperparameters fixed), these bounds increase by a couple of orders of magnitude (see Figure 9).

Although we do not plot the bounds from Nagarajan & Kolter (2019); Golowich et al. (2018), these have nearly identical norms in their numerator, and so one would not expect these bounds to show radically better behavior. Overall, our observations reveal previously unknown, fundamental problems with existing generalization bounds – barring the ones that do not apply on the original network (criterion d) and the ones that are constructed for specifically curated hyperparameter choices. While these issues may be addressed to some extent with a better understanding of implicit regularization in deep learning, we regard our observations as a call for taking a step back and clearly understanding inherent limitations in the theoretical tool underlying all these bounds: uniform convergence.

A possible fundamental limitation in existing approaches (ruled out). Before proceeding to our main theoretical result about uniform convergence, we emphasize that we considered the possibility that some existing approaches may suffer from the above problems due to a fundamental issue that does not involve uniform convergence (we defer much of this discussion to Appendix B). Specifically, we considered margin-based Rademacher bounds (e.g., Bartlett et al. (2017)) which, besides bounding the generalization error, also effectively act as an upper bound on the difference in the mean margins of the network on the test and the training set. Now, even if the generalization error decreases with sample size m, it is possible for the difference in the margins to not decrease with m, if the classifier suffers from a phenomenon we call as *pseudo-overfitting* i.e., roughly speaking, the classifier magnifies its real-valued output specifically near each training point even though it learns a simple decision boundary. Then, the Rademacher bound would not decrease with m either. As discussed in detail in Appendix B, we establish that deep networks however do *not* pseudo-overfit their margins and the differences in test/train margins do decrease substantially with m – thus, justifying our interest in scrutinizing uniform convergence.

4. Preliminaries

Let $\mathcal{H}$ be a class of hypotheses mapping from $\mathcal{X}$ to $\mathbb{R}$, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{-1, +1\}$. For the loss function, typically, we care about the 0-1 error. But a direct analysis of the uniform convergence of the 0-1 error is hard, and hence, occasionally a more general margin-based surrogate of this error (also called as ramp loss) is analyzed for uniform convergence. Specifically, given the classifier's logit output $y' \in \mathbb{R}$ and the true label $y \in \{-1, +1\}$, define:

$$\mathcal{L}^{(\gamma)}(y', y) = \begin{cases} 0, & y \cdot y' > \gamma \\ 1 - \frac{y \cdot y'}{\gamma} & y \cdot y' \in (0, \gamma] \\ 1, & y \cdot y' \leq 0 \end{cases}$$

Note that $\mathcal{L}^{(0)}$ is the 0-1 error, and $\mathcal{L}^{(\gamma)}$ is an upper bound on the 0-1 error. We define for any loss $\mathcal{L}$, the expected loss and the empirical loss on a dataset S of m datapoints, as:

$$\mathcal{L}_{\mathcal{D}}(h) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}(h(\mathbf{x}), y)]$$
$$\hat{\mathcal{L}}_S(h) := \frac{1}{m} \sum_{(\mathbf{x},y)\in S} \mathcal{L}(h(\mathbf{x}), y)$$

Let $\mathcal{A}$ be the learning algorithm and let h_S be the hypothesis output by the algorithm on a dataset S (assume that any training-data-independent randomness, such as the initialization/data-shuffling is fixed). The generalization error of the algorithm is essentially a bound on the difference between the error of the hypothesis h_S learned on a training set S and the expected error over $\mathcal{D}$; this bound holds with high probability over the draws of S. More formally:

Definition 4.1. The **generalization error** of $\mathcal{A}$ with respect

to loss $\mathcal{L}$ is the smallest value $\epsilon_{\text{gen}}(m, \delta)$ such that:

$$Pr_{S\sim\mathcal{D}^m}\left[\mathcal{L}_{\mathcal{D}}(h_S) - \hat{\mathcal{L}}_S(h_S) > \epsilon_{\text{gen}}(m,\delta)\right] < \delta \quad (2)$$

To theoretically bound the generalization error of the algorithm, the most common approach is to provide a two-sided uniform convergence bound on the hypothesis class used by the algorithm, where, for a given draw of S, we look at convergence for all the hypotheses in $\mathcal{H}$ instead of just h_S:

Definition 4.2. The **uniform convergence bound** with respect to loss $\mathcal{L}$ is the smallest value $\epsilon_{\text{unif}}(m, \delta)$ such that:

$$Pr_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}\left|\mathcal{L}_{\mathcal{D}}(h) - \hat{\mathcal{L}}_S(h)\right| > \epsilon_{\text{unif}}(m,\delta)\right] < \delta \quad (3)$$

4.1. Algorithm-dependent uniform convergence

The bound ϵ_{unif} is often quite loose because $\mathcal{H}$ might be a complex space that contains extraneous hypotheses that are never picked by the learning algorithm $\mathcal{A}$ for a given simple distribution $\mathcal{D}$. One can obtain tighter bounds by taking into account implicit regularization by the algorithm and applying uniform convergence on a smaller class of hypothesis. This is typically done by examining what norms are implicitly controlled by the algorithm, and then focusing on a norm-bounded class of hypotheses. We take this to the extreme by applying uniform convergence on the smallest possible class of hypotheses, namely, *only* those hypotheses that are picked by $\mathcal{A}$ under $\mathcal{D}$, excluding everything else. Crucially, applying uniform convergence on this aggressively pruned hypothesis class would lead to *the tightest possible uniform convergence bound*, and pruning it any further would not imply a bound on the generalization error. We care about this formulation because our goal is to show that even such a careful application of uniform convergence fails to explain generalization in some simple cases.

To formally capture this idea, it is helpful to first rephrase the previous definition of ϵ_{unif}: we can say that $\epsilon_{\text{unif}}(m, \delta)$ is the smallest value for which there exists a set of sample sets $\mathcal{S}_\delta \subseteq (\mathcal{X} \times \{-1, 1\})^m$ such that $Pr_{S\sim\mathcal{D}^m}[S \in \mathcal{S}_\delta] \geq 1-\delta$ and furthermore, $\sup_{S\in\mathcal{S}_\delta} \sup_{h\in\mathcal{H}} |\mathcal{L}_{\mathcal{D}}(h) - \hat{\mathcal{L}}_S(h)| \leq \epsilon_{\text{unif}}(m, \delta)$. This is equivalent to Definition 4.2.

Extending the above rephrased definition, we can define the tightest uniform convergence bound by replacing $\mathcal{H}$ in the above definition with an aggressively pruned class that corresponds to only those hypotheses that are explored by the algorithm $\mathcal{A}$ under $\mathcal{S}_\delta$:

Definition 4.3. The **algorithm-dependent uniform convergence bound** with respect to loss $\mathcal{L}$ is the smallest value $\epsilon_{\text{unif-alg}}(m, \delta)$ for which there exists a set of sample sets $\mathcal{S}_\delta$ such that $Pr_{S\sim\mathcal{D}^m}[S \in \mathcal{S}_\delta] \geq 1 - \delta$ and if we define the space of hypotheses explored by $\mathcal{A}$ on $\mathcal{S}_\delta$ as $\mathcal{H}_\delta := \bigcup_{S\in\mathcal{S}_\delta}\{h_S\} \subseteq \mathcal{H}$, the following holds:

$$\sup_{S\in\mathcal{S}_\delta}\sup_{h\in\mathcal{H}_\delta}\left|\mathcal{L}_{\mathcal{D}}(h) - \hat{\mathcal{L}}_S(h)\right| \leq \epsilon_{\text{unif-alg}}(m,\delta)$$

Remark 1. Since $\forall S \in \mathcal{S}_\delta$, we have $h_S \in \mathcal{H}_\delta$, we also have $\epsilon_{\text{gen}}(m, \delta) \leq \epsilon_{\text{unif-alg}}(m, \delta)$. On the other hand, since $\mathcal{H}_\delta \subseteq \mathcal{H}$, $\epsilon_{\text{unif-alg}}(m, \delta) \leq \epsilon_{\text{unif}}(m, \delta)$.

Remark 2. Often, uniform convergence bounds are written with an explicit dependence on the weights learned e.g., for a classifier with weights $\mathbf{w}$, the bound might be of the form $\|\mathbf{w}\|_2/\sqrt{m}$. However, the term $\epsilon_{\text{unif-alg}}(m, \delta)$ is evidently devoid of any dependence on the weights. This is because our bound applies in supremum over all hypotheses picked by $\mathcal{A}$, which is what we ultimately care about in order to explain generalization e.g., if the algorithm ensures $\|\mathbf{w}\|_2 \leq 1$, this bound would correspond to $1/\sqrt{m}$. Thus, we emphasize that our definition is not restrictive in any sense.

5. Theoretical Model

We present the second part of our contribution in this section: we argue how even the algorithm-dependent uniform convergence, as in $\epsilon_{\text{unif-alg}}$, can fail to explain generalization by yielding nearly vacuous bounds under overparameterization. First we quickly outline the main mathematical goal. Deriving a small uniform convergence bound boils down to picking a sample set space $\mathcal{S}_\delta$ of mass $1 - \delta$ such that $\sup_{S\in\mathcal{S}_\delta} \sup_{h\in\mathcal{H}_\delta} |\mathcal{L}_{\mathcal{D}}(h) - \hat{\mathcal{L}}_S(h)|$ is small. On one hand, it is possible that for every $S \in \mathcal{S}_\delta$, the corresponding $h_S \in \mathcal{H}_\delta$ has low empirical error on S and low expected error on $\mathcal{D}$, resulting in low generalization error. On the other hand, two-sided uniform convergence demands that *every possible pair* of $S \in \mathcal{S}_\delta$ and $h \in \mathcal{H}_\delta$ have empirical error that is close to the small test error of $h \in \mathcal{H}_\delta$. However, we will construct a situation where there exist 'bad' (S, h) pairs *with large empirical error* and low test error that ruin two-sided uniform convergence.

We now present our model. Specifically, we devise a learning algorithm for an overparameterized linear classifier which has a generalization error less than some small ϵ for a simple class of linearly separable data distributions. On the other hand, any uniform convergence bound will be at least $1 - \epsilon$ (and this holds with respect to both the 0-1 error and the margin-based losses). As we will see, our theoretical setup is intended to model our main empirical observations about norms and margins and some other well-known intuitive aspects of deep networks.

Distribution $\mathcal{D}$: Let each input be a $K + D$ dimensional vector, where K can be thought of as a small constant ($\ll$

m, D). The value of any input $\mathbf{x}$ is denoted by $(\mathbf{x}_1, \mathbf{x}_2)$ where $\mathbf{x}_1 \in \mathbb{R}^K$ and $\mathbf{x}_2 \in \mathbb{R}^D$. Let the centers of the classes be determined by an arbitrary vector $\mathbf{u} \in \mathbb{R}^K$ such that $\|\mathbf{u}\|_2 = 1$. Let $\mathcal{D}$ be such that the label y has equal probability of being $+1$ and -1, and $\mathbf{x}_1 = 2 \cdot y \cdot \mathbf{u}$ while $\mathbf{x}_2$ is sampled independently from $\mathcal{N}(0, \frac{32}{D} I)$ i.e., a spherical Gaussian with mean at the origin and variance $32/D$ along all the axes. Note that the distribution is linearly separable based on the first K dimensions. [2]

Learning algorithm $\mathcal{A}$: Consider a linear classifier with weights $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$ and whose output is $h(\mathbf{x}) = \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2$. Assume the weights are initialized to origin. Given $S = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}$, $\mathcal{A}$ learns $\mathbf{w}_1 = \frac{1}{m}\sum_i y^{(i)}\mathbf{x}_1^{(i)}$ (i.e., $2\mathbf{u}$) and $\mathbf{w}_2 = \sum_i y^{(i)}\mathbf{x}_2^{(i)}$.

While the absence of an $1/m$ factor in $\mathbf{w}_2$ might seem artificial, this is *necessary* to model our empirical observation: *the norms grow as $poly(m)$, while the training margins do not*. Specifically, $\mathbf{w}_1$ is aligned along the class boundary and has an ℓ_2 norm of 2; $\mathbf{w}_2 \sim \mathcal{N}(0, \frac{32m}{D} I)$, which is noise and does not contribute much to the margins, has ℓ_2 norm w.h.p $\Theta(\sqrt{m})$. Then, the distance of $\mathbf{w}$ from the origin, and also from the weights learned on other draws of the training set grow as $\Theta(\sqrt{m})$; on the other hand, we can show that the margins on the training set is $\Theta(1)$. [3]

Note that the intuition behind using noise to model the m-dependence of the norm is that with decreasing batch size, the norm increases empirically (Figure 9, **left**), like the behavior of the magnitude of noise in SGD.

Below, we present our main result about uniform convergence. Effectively, our result holds in a setup that parallels the deep learning world in terms of (i) noise in the parameters, (ii) our empirical observations about the norms and margins and (iii) mild overparameterization.

Theorem 5.1. *For any $\epsilon, \delta > 0, \delta < 1/8$, under mild overparameterization such that $D = \Omega\left(\max\left(m \ln \frac{m}{\delta}, m \ln \frac{1}{\epsilon}\right)\right)$, $\gamma \in [0, 1]$, for the $\mathcal{L}^{(\gamma)}$ loss:*

$$\epsilon_{gen}(m, \delta) \leq \epsilon, \text{ while } \epsilon_{unif\text{-}alg}(m, \delta) \geq 1 - \epsilon.$$

Furthermore, for all $\gamma \geq 0$, for the $\mathcal{L}^{(\gamma)}$ loss, $\epsilon_{unif\text{-}alg}(m, \delta) \geq 1 - \epsilon_{gen}(m, \delta)$.

Before we delve into the proof outline, it will be helpful to emphasize why our result is particularly interesting. Typically, uniform convergence is said to provide weak bounds as one often applies it on a hypothesis class that includes complex hypotheses that are never picked by the algorithm – the accusation here is not really on uniform convergence itself, but on how it is applied. One could tighten these bounds by taking into account the implicit bias/regularization of the algorithm, such as any observations about norms that are controlled by the algorithm: in this case, we know that w.h.p, $\|\mathbf{w}\| = \Theta(\sqrt{m})$ and the training set is classified by a constant margin say, $\gamma^\star$. This can be incorporated into a standard Rademacher complexity based analysis to get a generalization bound that scales as $\mathcal{O}(\|\mathbf{w}\|/(\gamma^\star\sqrt{m}))$ – which would essentially be a constant independent of m, D, ϵ and much larger than the actual generalization error ϵ.

One might persist and think that perhaps, the characterization of $\mathbf{w}$ to be bounded in ℓ_2 norm does not fully capture the implicit bias. Are there other properties of the Gaussian $\mathbf{w}_2$ that one could take into account to identify an even smaller class of hypotheses? In fact, the best we could hope for is to consider a class that includes only those classifiers that are picked by the algorithm with probability $1 - \delta$ over the training sets i.e., a class such that (i) every hypothesis has $\mathbf{w}_1 = 2\mathbf{u}$, and (ii) the set of all $\mathbf{w}_2$ of the hypotheses in the class corresponds to some $1 - \delta$ mass of the Gaussian. We show that even after fixing $\mathbf{w}_1$ this way, and for any possible truncation of the distribution of $\mathbf{w}_2$, the resulting, carefully pruned class of weights – despite all of them having a test error less than ϵ – would give only nearly vacuous bounds as $\epsilon_{\text{unif-alg}}(m, \delta) \geq 1 - \epsilon$.

In summary, even in a simple, mildly overparameterized setup that preserves the key elements of deep learning, however hard one might try to incorporate the implicit bias of the algorithm, and whatever uniform convergence based tool one employs, one could grossly over-estimate the generalization error.

5.1. Proof outline

We provide an outline of our argument here, deferring the proof to the appendix. Note that our discussion below applies for all losses $\mathcal{L}^{(\gamma)}$ with $\gamma \in [0, 1]$. At a high level, our analysis rests on the fact that $\mathbf{w}_1$ is aligned correctly along the true boundary; but for large dimensions, the noisy part of the classifier $\mathbf{w}_2$ is poorly aligned with the inputs, and hence does not dominate the output of the classifier – preserving the good fit of $\mathbf{w}_1$ on the test/training data. On the other hand, under the purview of uniform convergence, the noise vector $\mathbf{w}_2$ is stripped of its randomness, and becomes 'adversarial'. This misleads uniform convergence into accounting for the D noise dimensions into the representational complexity of the classifier, giving nearly vacuous bounds. Below, we provide a more technical outline.

Small generalization error. For any training/test input $(\mathbf{x}, y)$, we can show that $y \cdot h_S(\mathbf{x}) \approx 4 + \mathcal{N}(0, \Theta(m/D))$,

[2] As discussed in Appendix D.1, one could easily extend the discussion by assuming that $\mathbf{x}_1$ is spread out around $2y\mathbf{u}$.

[3] Our goal here is to show how the m-dependence in norms, specifically through noise, is enough to hurt uniform convergence. Our goal is *not* to provide a model that naturally explains how an m-dependent magnitude of noise comes about in deep learning, which is an interesting research question of its own. Thus, we emphasize that the artificiality in our algorithm is unrelated to the insights about uniform convergence that our results provide.

where the two terms correspond to the components $\mathbf{w}_1$ and $\mathbf{w}_2$ respectively. Since $D = \Omega(m \ln 1/\delta)$, the second term is quite small, and w.h.p $1-\delta$ over draws of S, $y \cdot h_S(\mathbf{x}) > 1$ for all training inputs and also for $1-\epsilon$ mass of the test inputs. Hence, $\epsilon_{\text{gen}}(m, \delta) \leq \epsilon$.

Nearly vacuous uniform convergence bound. We first argue that w.h.p $1 - \mathcal{O}(\delta)$ over draws of S, h_S completely misclassifies the 'noise-negated' version of S, namely $S' = \{(\mathbf{x}_{\text{neg}}^{(1)}, y^{(1)}), \ldots (\mathbf{x}_{\text{neg}}^{(m)}, y^{(m)})\}$ where $\mathbf{x}_{\text{neg}}^{(i)} = (\mathbf{x}_1^{(i)}, -\mathbf{x}_2^{(i)})$. To prove this, we first show, $y^{(i)} h_S(\mathbf{x}_{\text{neg}}^{(i)}, y^{(i)}) \approx 4 - \|\mathbf{x}_2^{(i)}\|^2 - \mathcal{N}(0, \Theta(m/D))$, where the last two terms come from $\mathbf{w}_2$. The last term is quite small as we noted already. However, the second term that is negative (which was positive on the training set) has a magnitude that, in high dimensions, tightly concentrates around the expected ℓ_2 norm of $\mathbf{x}_2^{(i)} \sim \mathcal{N}(0, \Theta(1/D)I)$, which is a sufficiently large constant. As a result, the second (negative) term dominates the output, resulting in complete misclassification of S'.

Now recall that to compute $\epsilon_{\text{unif-alg}}$ one has to pick a sample set space $\mathcal{S}_\delta$ of mass $1-\delta$. We first argue that for *any* choice of $\mathcal{S}_\delta$, there must exist $S_\star$ such that (i) $S_\star \in \mathcal{S}_\delta$, (ii) the noise-negated $S'_\star \in \mathcal{S}_\delta$, (iii) $h_{S_\star}$ has test error less than ϵ and (iv) $h_{S_\star}$ completely misclassifies $S'_\star$.

We show this by arguing that over the draws of $S \sim \mathcal{D}^m$, there is non-zero probability of picking an S that satisfies all these conditions. First, over the draws of S, by construction of $\mathcal{S}_\delta$, (i) alone fails with probability at most δ. We have established that (iii) and (iv) too fail with probability $\mathcal{O}(\delta)$. As for (ii), note that under the draws of S, the noise-negated dataset S' has the same distribution as that of $\mathcal{D}^m$ (since negating the Gaussian vector does not affect its distribution); hence, by construction of $\mathcal{S}_\delta$, even (ii) fails with probability δ. Thus, by a union bound, the probability of picking an $S \sim \mathcal{D}^m$ that satisfies (i)-(iv) is at least $1 - \mathcal{O}(\delta) > 0$. Then, for a given $\mathcal{S}_\delta$, there must exist an $S_\star$ satisfying (i)-(iv), which in turn implies our claim as $\epsilon_{\text{unif-alg}}(m, \delta) = \sup_{S \in \mathcal{S}_\delta} \sup_{h \in \mathcal{H}_\delta} |\mathcal{L}_\mathcal{D}(h) - \hat{\mathcal{L}}_S(h)| \geq |\mathcal{L}_\mathcal{D}(h_{S_\star}) - \hat{\mathcal{L}}_{S'_\star}(h)| = |\epsilon - 1| = 1 - \epsilon$.

Remark 1. Our analysis crucially depends on the fact that $\epsilon_{\text{unif-alg}}$ is a two-sided convergence bound – which is what existing techniques bound – and our result would not apply for hypothetical one-sided uniform convergence bounds. Existing uniform convergence based tools are only two-sided as it is more natural to bound the absolute value of the difference in the test/train errors. While PAC-Bayes based bounds are typically presented as one-sided bounds, we show in Appendix H that even these are lower-bounded by the two-sided $\epsilon_{\text{unif-alg}}$. To the best of our knowledge, it is non-trivial to make any of these tools purely one-sided.

Remark 2. It is worth noting that in our example, it is easy to modify the learned classifier to derive non-vacuous bounds on a modified classifier. Specifically, we can zero out the weights $\mathbf{w}_2$ without affecting the training/test error of the classifier. The resulting classifier with much fewer parameters, would give us a non-vacuous uniform convergence bound. However, such a bound would not fully explain why the original classifier generalizes well; one might then wonder if such a bound could be extended to the original classifier (like it was explored in (Nagarajan & Kolter, 2019) for deep networks). Our result implies that no such extension is possible in this particular example.

Remark 3. Although the input distribution is high-dimensional in our setup, this aspect is not critical as our negative result extends to low-dimensional input distributions provided the classifier is more complex (Appendix E).

5.2. Our deep learning conjecture

We now extend the above insights to deep learning. Our main conjecture is that in the case of overparameterized deep networks, SGD finds a fit that is simple at a macroscopic level (leading to good generalization) but also has many microscopic fluctuations (hurting uniform convergence). Stated differently, it should be possible to decompose the network learned by SGD into two components: (i) a "low-complexity" component (like $\mathbf{w}_1$) and (ii) a "high-dimensional noisy" component (like $\mathbf{w}_2$). While in our example the noise directly arises from the training data, in deep ReLU networks, this would arise from a more complicated interaction between SGD and the non-convex, non-smooth training loss.

We first argue how this helps explain our empirical observations. First, the magnitude of noise in SGD naturally grows with the training data set size m and with decreasing batch size; this would cause the norms of the weights to show similar dependence on m and the batch size, as we observed. However, for wider networks, as the parameter space increases in dimensionality, the noise has poorer alignment with the inputs/hidden layers; at the same time, the magnitude of noise does not increase, as indicated by the fact that the ℓ_2 distance from initialization does not increase with width (Nagarajan & Kolter, 2017). Thus, for larger networks, the noisy component is less likely to contribute to the output. This would explain why the margins do not grow even though the ℓ_2 norms grow.

We now argue how this noise impairs uniform convergence. With so many parameters to spare, the noisy component could create many fluctuations in the function landscape (and the decision boundary), although covering only low-probability regions of the input space (and hence, not affect-

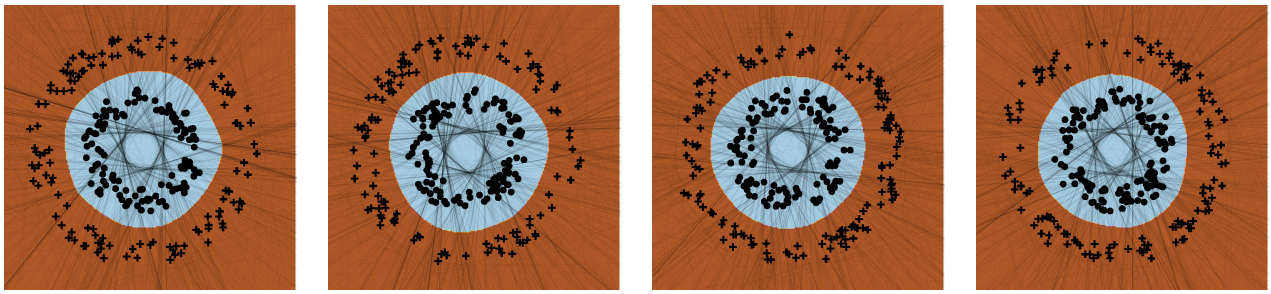

Figure 3. **An illustration of our conjecture**: In each figure above we plot the boundary learned by a two layer neural network (of width $10k$), *from the same initialization*, for four different draws of 1024 training datapoints (depicted as $+$ and $\bullet$) in a circular binary classification task. The faint lines represent the $10k$ hyperplanes learned by the hidden layer (each hyperplane is faded according to the magnitude of the corresponding weight in the output layer). Observe that although each of the learned boundaries is roughly circular at a macroscopic level, more precisely, each boundary is a many-sided, irregular polygon – the fine-grained description of this polygon is determined by the orientation of each of the $10k$ hyperplanes, which are roughly arranged in a circle. This arrangement however varies widely across the different datasets (this can be better appreciated in the figures after zooming); empirically, this variation would manifest in terms of how the ℓ_2 distance between the parameters learned on different datasets increases with the dataset size (without a corresponding increase in the margins). Now, although the learned boundary is only small deviations away from a simpler boundary, we conjecture that uniform convergence misjudges these deviations to be symptomatic of overfitting because they are highly dependent on the training data (and we provably demonstrate this intuition for an overparameterized linear classifier).

ing the generalization error). Crucially, these fluctuations would vary widely across different draws of the datasets, as hinted by the fact that the trajectory of SGD diverges widely across datasets (orange line in Figure 1 top right). Now, even if uniform convergence is applied only to the set of low-test-error networks, it estimates the "representational power" of *all possible such widely different, complex, fluctuations*, while being blind to the fact that they each cover only low probability regions – thus, producing vacuous bounds. We can also intuit this differently: despite the fact that the noisy component has the property that it does not dominate the output on training data, *uniform convergence may not able to generalize this property over to test data*.

To help illustrate our conjecture, we present Figure 3 where we show the boundaries learned by a single-hidden-layer neural network across different draws of the training set in a toy example. Although we leave a precise validation of this conjecture for future work, we present a preliminary discussion in Appendix F, where we suggest that in our MNIST experiments, all but the top few singular directions of the update matrices $W_d - Z_d$ could be considered as the noisy component. Perhaps, existing works that have achieved strong uniform convergence bounds on modified networks, may have done so by effectively suppressing such a noisy component – either by compression, optimization or stochasticization – leaving behind the low-complexity component which obeys uniform convergence. Revisiting these approaches may help validate our conjecture.

6. Conclusion and Future Work

While uniform convergence bounds may provide partial intuition for why deep networks generalize well, through empirical and theoretical evidences, we cast doubt on their potential to achieve the grand goal: a small bound that shows appropriate dependence on the sample size, width, depth, label noise, and batch size. First, we empirically showed that existing uniform convergence bounds can surprisingly increase with training set size for small batch sizes. We then presented a noisy, overparameterized linear classifier that reflects our empirical observations, and for which uniform convergence provably fails to explain generalization even after taking implicit bias into account.

In the future, notwithstanding our negative result, one could explore whether existing bounds can be tightened with the knowledge of the newly discovered source of vacuity. Or perhaps, algorithmic stability (Feldman & Vondrák, 2018; Hardt et al., 2016; Bousquet & Elisseeff, 2002; Shalev-Shwartz et al., 2010) which has not been as extensively studied for deep networks, might hold more promise in explaining generalization (although there are challenges specific to using them under non-smooth optimization).

Our linear setup might also inspire new tools to explain generalization. We envision the following: assume we manage to explicitly characterize the distribution of noise in deep networks; then, by applying uniform convergence on the low-complexity component (e.g., a compressed network), followed by standard tail bounds to argue that the noisy component does not affect the output, one would have a complete story for why deep networks generalize.

Acknowledgements. Vaishnavh Nagarajan is supported by a grant from the Bosch Center for AI.

References

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. abs/1811.04918, 2018. URL http://arxiv.org/abs/1811.04918.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In *The 35th International Conference on Machine Learning, ICML*, 2018.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of Machine Learning Research*, 2, 2002.

Brutzkus, A., Globerson, A., Malach, E., and Shalev-Shwartz, S. SGD learns over-parameterized networks that provably generalize on linearly separable data. *International Conference on Learning Representations (ICLR)*, 2018.

Dräxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially no barriers in neural network energy landscape. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018.

Dziugaite, G. K. and Roy, D. M. 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*, 2017.

Feldman, V. and Vondrák, J. Generalization bounds for uniformly stable algorithms. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, 2018.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, 2018.

Golowich, N., Rakhlin, A., and Shamir, O. Size-independent sample complexity of neural networks. *Computational Learning Theory, COLT 2018*, 2018.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 2016.

Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension bounds for piecewise linear neural networks. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*, 2017.

Hinton, G. E. and van Camp, D. 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.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1), 1997.

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in Neural Information Processing Systems (*to appear*)*, 2017.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. Width of minima reached by stochastic gradient descent is influenced by learning rate to batch size ratio. In *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks*, 2018.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*, 2017.

Langford, J. and Caruana, R. (not) bounding the true error. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001]*, 2001.

Langford, J. and Shawe-Taylor, J. Pac-bayes & margins. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002*, 2002.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, 2018.

McAllester, D. Simplified pac-bayesian margin bounds. In *Learning Theory and Kernel Machines*. Springer Berlin Heidelberg, 2003.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012.

Nagarajan, V. and Kolter, J. Z. 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.

Nagarajan, V. and Kolter, Z. Deterministic PAC-bayesian generalization bounds for deep networks via generalizing noise-resilience. In *International Conference on Learning Representations (ICLR)*, 2019.

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

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In *Proceedings of The 28th Conference on Learning Theory, COLT*, 2015b.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *Advances in Neural Information Processing Systems* to appear, 2017.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations (ICLR)*, 2018.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Rogers, W. H. and Wagner, T. J. A finite sample distribution-free performance bound for local discrimination rules. *The Annals of Statistics*, 6(3), 1978.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11, 2010.

Soudry, D., Hoffer, E., and Srebro, N. The implicit bias of gradient descent on separable data. *International Conference on Learning Representations (ICLR)*, 2018.

Vapnik, V. N. and Chervonenkis, A. Y. *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities*. 1971.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2017.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. In *International Conference on Learning Representations (ICLR)*, 2019.

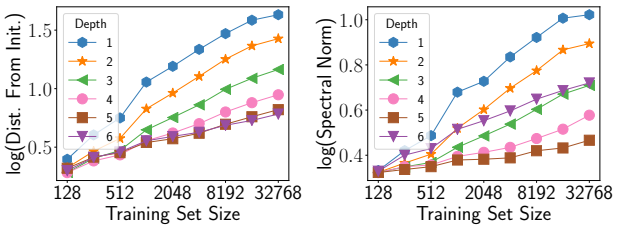

Figure 4. We plot the distance from initialization and the spectral norm of each individual layer, and observe that the lowermost layer shows the greatest dependence on m.

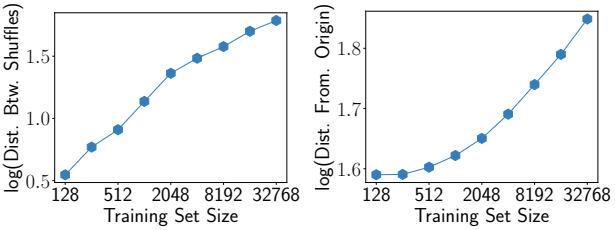

Figure 5. On the **left**, we plot the distance between the weights learned on the two different shuffles of the same dataset, and it grows as fast as the distance from initialization. On the **right**, we plot the distance of the weights from the origin, learned for a network of width $h = 256$ and depth $d = 6$; for sufficiently large m, this grows as $\Omega(m^{0.42})$.

A. More Experiments

In this section, we present more experiments along the lines of what we presented in Section 3.

Layerwise dependence on m. Recall that in the main paper, we show how the distance from initialization and the product of spectral norms vary with m for network with six layers. In Figure 4, we show how the terms grow with sample size m for each layer individually. Our main observation is that the first layer suffers from the largest dependence on m.

Distance between trajectories of shuffled datasets grows with m. In the main paper, we saw that the distance between the solutions learned on different draws of the dataset grow substantially with m. In Figure 5 (left), we show that even the distance between the solutions learned on the same draw, but a different shuffling of the dataset grows substantially with m.

Frobenius norms grow with m when $m \gg h$. Some bounds like (Golowich et al., 2018) depend on the Frobenius norms of the weight matrices (or the distance from origin), which as noted in (Nagarajan & Kolter, 2017) are in fact width-dependent, and grow as $\Omega(\sqrt{h})$. However, even these terms do grow with the number of samples in the regime where m is larger than h. In Figure 5, we report the total distance from origin of the learned parameters for a network with $h = 256$ (we choose a smaller width to better emphasize the growth of this term with m); here, we see that for $m > 8192$, the distance from origin grows at a rate of $\Omega(m^{0.42})$ that is quite similar to what we observed for distance from initialization.

Median margin. Recall that in the main paper, we considered the possibility of setting γ to be the median value of the margin $\Gamma(f(\mathbf{x}), y)$ on the training dataset, and computing the generalization bound from Equation 1. We observed that even on applying this median value, the generalization bounds grow with sample size. This is because, as shown in Figure 6, the median margin value does not grow as fast with m as the numerators of these bounds grow.

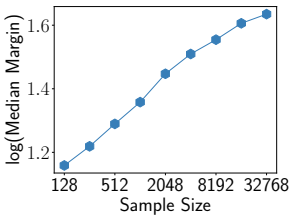

Figure 6. We plot the median value of the margin $\Gamma(f(\mathbf{x}), y)$ on the training dataset and observe that it grows as $\mathcal{O}(m^{0.2})$.

Effect of depth. We observed that as the network gets shallower the bounds show better dependence with m. As an extreme case, we consider a network with only one hidden layer, and with $h = 50000$. Here we also present a third bound, namely that of Neyshabur et al. (2019), besides the two bounds discussed in the main paper. Specifically, if Z_1, Z_2 are the random initializations of the weight matrices in the network, the generalization error bound (the last term in Equation 1) here is of the following form, ignoring log factors:

$$\frac{\|W_2\|_F(\|W_1 - Z_1\|_F + \|Z_1\|_2)}{\gamma\sqrt{m}} + \frac{\sqrt{h}}{\sqrt{m}}$$

The first term here is meant to be width-independent, while the second term clearly depends on the width and does decrease with m at the rate of $m^{-0.5}$. Hence, in our plots in Figure 7, we only focus on the first term. We see that these bounds are almost constant and decrease at a minute rate of $\Omega(m^{-0.066})$ while the test errors decrease much faster, at the rate of $O(m^{-0.35})$.

Effect of width. In Figure 8, we demonstrate that our observation that the bounds increase with m extends to widths $h = 128$ and $h = 2000$ too.

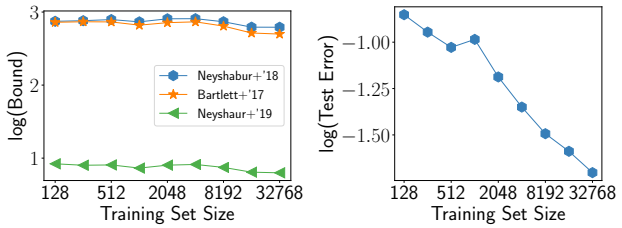

Figure 7. On the **left**, we plot how the bounds vary with sample size for a single hidden layer network with $50k$ hidden units. We observe that these bounds are almost constant, and at best decrease at a meagre rate of $\Omega(m^{-0.066})$. On the **right**, we plot the test errors for this network and observe that it decreases with m at the rate of at least $O(m^{0.35})$.

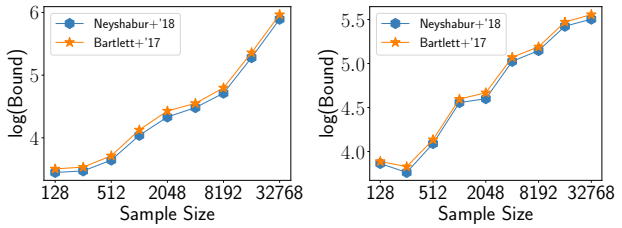

Figure 8. On the **left**, we plot the bounds for varying m for $h = 128$. All these bounds grow with m as $\Omega(m^{0.94})$. On the **right**, we show a similar plot for $h = 2000$ and observe that the bounds grow as $\Omega(m^{0.79})$.

A.1. Effect of batch size

Bounds vs. batch size for fixed m. In Figure 9, we show how the bounds vary with the batch size for a fixed sample size of 16384. It turns out that even though the test error decreases with decreasing batch size (for our fixed stopping criterion), all these bounds *increase* (by a couple of orders of magnitude) with decreasing batch size. Again, this is because the terms like distance from initialization *increase* for smaller batch sizes (perhaps because of greater levels of noise in the updates). Overall, existing bounds do not reflect the same behavior as the actual generalization error in terms of their dependence on the batch size.

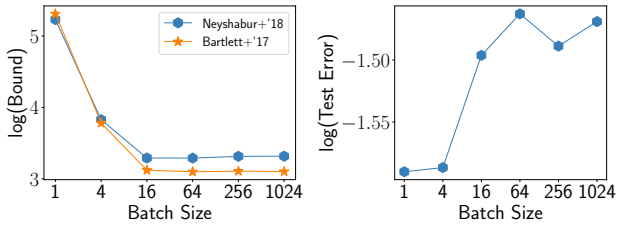

Figure 9. On the **left**, we plot the bounds for varying batch sizes for $m = 16384$ and observe that these bounds *decrease* by around 2 orders of magnitude. On the **right**, we plot the test errors for varying batch sizes and observe that test error increases with batch size albeit slightly.

Bounds vs. m for batch size of 32. In the main paper, we only dealt with a small batch size of 1. In Figure 10, we show bounds vs. sample size plots for a batch size of 32. We observe that in this case, the bounds do decrease with sample size, although only at a rate of $\mathcal{O}(m^{-0.23})$ which is not as fast as the observed decrease in test error which is $\Omega(m^{-0.44})$. Our intuition as to why the bounds behave better (in terms of m-dependence) in the larger batch size regime is that here the amount of noise in the parameter updates is much less compared to smaller batch sizes (and as we discussed earlier, uniform convergence finds it challenging to explain away such noise).

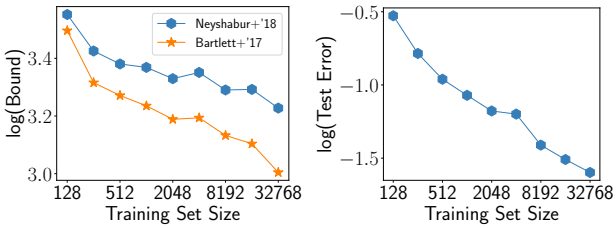

Figure 10. On the **left**, we plot the bounds for varying m for a batch size of 32 and observe that these bounds do decrease with m as $\mathcal{O}(1/m^{0.23})$. On the **right**, we plot the test errors for various m for batch size 32 and observe that test error varies as $\Omega(1/m^{0.44})$.

Squared error loss. All the experiments presented so far deal with the cross-entropy loss, for which the optimization procedure ideally diverges to infinity; thus, one might suspect that our results are sensitive to the stopping criterion. It would therefore be useful to consider the squared error loss where the optimum on the training loss can be found in a finite distance away from the random initialization. Specifically, we consider the case where the squared error loss between the outputs of the network and the one-hot encoding of the true labels is minimized to a value of 0.05 on average over the training data.

We observe in Figure 11 that even for this case, the distance from initialization and the spectral norms grow with the sample size at a rate of at least $m^{0.3}$. On the other hand, the test error decreases with sample size as $1/m^{0.38}$, indicating that even for the squared error loss, these terms hurt would hurt the generalization bound with respect to its dependence on m.

B. Pseudo-overfitting

Recall that in the main paper, we briefly discussed ruling out the possibility of pseudo-overfitting being the reason behind why some techniques lead to vacuous generalization bounds. We describe this in more detail here. We emphasize this discussion because i) it brings up a fundamental and so far unknown issue that might potentially exist in current approaches to explaining generalization and ii) rules

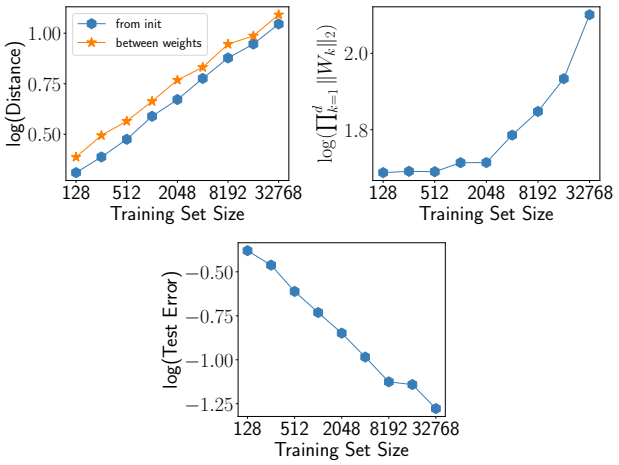

Figure 11. On the **top left** we plot the distance from initialization and the distance between weights learned on two different random draws of the datasets, as a function of varying training set size m, when trained on the squared error loss. Both these quantities grow as $\Omega(m^{0.35})$. On the **top right**, we show how the product of spectral norms grow as $\Omega(m^{0.315})$ for sufficiently large $m \geq 2048$. In the **bottom**, we observe that the test error (i.e., the averaged squared error loss on the test data) decreases with m as $\mathcal{O}(m^{-0.38})$.

it out before making more profound claims about uniform convergence.

Our argument specifically applies to margin-based Rademacher complexity approaches (such as Bartlett et al. (2017); Neyshabur et al. (2019)). These result in a bound like in Equation 1 that we recall here:

$$Pr_{(x,y)\sim\mathcal{D}}[\Gamma(f(\mathbf{x}),y) \leq 0] \leq \frac{1}{m}\sum_{(x,y)\in S} \mathbf{1}[\Gamma(f(\mathbf{x}),y) \leq \gamma] + \text{generalization error bound } (1)$$

These methods upper bound the uniform convergence bound on the $\mathcal{L}^{(\gamma)}$ error on the network in terms of a uniform convergence bound on the margins of the network (see (Mohri et al., 2012) for more details about margin theory of Rademacher complexity). The resulting generalization error bound in Equation 1 would take the following form, as per our notation from Definition 4.3:

$$\sup_{S\in\mathcal{S}_\delta} \sup_{h\in\mathcal{H}_\delta} \frac{1}{\gamma}\left|\mathbb{E}_{\mathcal{D}}[\Gamma(h(\mathbf{x}),y)] - \frac{1}{m}\sum_{(x,y)\in S}\Gamma(h(\mathbf{x}),y)\right| \quad (4)$$

This particular upper bound on the generalization of the $\mathcal{L}^{(\gamma)}$ error is also an upper bound on the generalization of the margins. That is, with high probability $1-\delta$ over the draws of S, the above bound is larger than the following term that corresponds to the difference in test/train margins:

$$\frac{1}{\gamma}\left(\mathbb{E}_{(x,y)\sim\mathcal{D}}[\Gamma(h_S(\mathbf{x}),y)] - \frac{1}{m}\sum_{(x,y)\in S}\Gamma(h_S(\mathbf{x}),y)\right) \quad (5)$$

We argue that it is possible for the generalization error of the algorithm to decrease with m (as roughly $m^{-0.5}$), but for the above quantity to be independent of m. As a result, the margin-based bound in Equation 4 (which is larger than Equation 5) will be non-decreasing in m, and even vacuous.

Such a scenario would happen if the network does learn a simple hypothesis to fit the data (say, by learning a simple linear input-output mapping on linearly separable data), but proceeds to *pseudo-overfit* to the samples by skewing up (down) the output of the network by some constant Δ in a tiny neighborhood around the positive (negative) training inputs. Note that this would be possible if and only if the network is overparameterized. By training the output to pseudo-overfit in this manner, we do not modify the classification itself on any input – and thus the boundary is still simple and linear and the generalization error small – but the training margins are at least a constant Δ larger than the test margins (which are not affected by the bumps created in tiny regions around the training data). Then, the term in Equation 5 would be larger than Δ/γ and so would the term in Equation 4. Now in the generalization guarantee of Equation 1, recall that we must pick a value of γ such that the first term is low i.e., most of the training datapoints must be classified by at least γ margin. In this case, we can at best let $\gamma \approx \Delta$ as any larger value of γ would make the margin-based training error non-negligible; as a result of this choice of γ, the bound in Equation 5 would be an m-independent constant close to 1. The same would also hold for its upper bound in Equation 4, which is the generalization bound provided by the margin-based techniques.

Clearly, this is a potential fundamental limitation in existing approaches, and if deep networks were indeed pseudo-overfitting this way, we would have identified the reason why at least some existing bounds are vacuous. However, (un)fortunately, we rule this out by observing that the difference in the train and test margins in Equation 5 does decrease with training dataset size m (see Figure 12) as $\mathcal{O}(m^{-0.33})$. Additionally, this difference is numerically much less than $\gamma^\star = 10$ (which is the least margin by which 99% of the training data is classified) as long as m is large, implying that Equation 5 is non-vacuous.

It is worth noting that the generalization error decreases at a

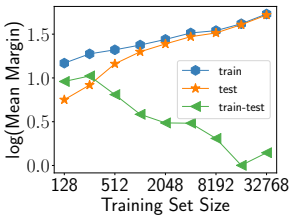

Figure 12. We plot the average margin of the network on the train and test data, and the difference between the two, the last of which decreases with m as $\mathcal{O}(1/m^{0.33})$.

faster rate of $\mathcal{O}(m^{-0.43})$ implying that the upper bound in Equation 5 which decreases only as $m^{-0.33}$, is loose. This already indicates a partial weakness in this specific approach to deriving generalization guarantees. Nevertheless, even this upper bound decreases at a significant rate with m which the subsequent uniform convergence-based upper bound in Equation 4 is unable to capture, thus hinting at more fundamental weaknesses specific to uniform convergence.

C. Useful Lemmas

In this section, we state some standard results we will use in our proofs. We first define some constants: $c_1 = 1/32$, $c_2 = 1/2$ and $c_3 = 3/2$ and $c_4 = \sqrt{2}$.

First, we state a concentration inequality for chi-squared random variables i.e., sum of squared normal variables.

Lemma C.1. *Let $z_1, z_2, \ldots, z_D \sim \mathcal{N}(0,1)$, then for all $t \in (0,1)$*

$$Pr\left[\left|\frac{1}{D}\sum_{d=1}^{D} z_d^2 - 1\right| \geq t\right] \leq 2\exp(-Dt^2/8)$$

We can then restate the above as follows, for use in our proof:

Corollary C.1.1. *For $z_1, z_2, \ldots, z_D \sim \mathcal{N}(0,1)$, we have that:*

$$Pr\left[\frac{1}{D}\sum_{d=1}^{D} z_d^2 \in [c_2, c_3]\right] \leq 2\exp(-c_1 D)$$

We now state a standard Gaussian concentration inequality.

Lemma C.2. *Let $z_1, z_2, \ldots, z_D$ be such that $z_d \sim \mathcal{N}(0, \sigma_d^2)$.*

$$Pr\left[\left|\sum_{d=1}^{D} z_d\right| \geq t\right] \leq 2\exp(-t^2/2\sum_{d=1}^{D}\sigma_d^2)$$

Again, we restate it as follows:

Corollary C.2.1. *For any $\mathbf{u} = (u_1, u_2, \ldots, u_d) \in \mathbb{R}^D$, for $z_1, z_2, \ldots, z_D \sim \mathcal{N}(0,1)$,*

$$Pr\left[\left|\sum_{d=1}^{D} u_d z_d\right| \geq \|\mathbf{u}\|_2 \cdot c_4\sqrt{\ln\frac{2}{\delta}}\right] \leq \delta$$

D. Proof for Theorem 5.1

Below, we state the precise theorem statement (where we've used the constants $c_1 = 1/32$, $c_2 = 1/2$ and $c_3 = 3/2$ and $c_4 = \sqrt{2}$):

Theorem 5.1 *In the set up above, for any $\epsilon, \delta > 0$ and $\delta < 1/4$, let D be sufficiently large that it satisfies:*

$$D \geq \frac{1}{c_1}\ln\frac{6m}{\delta} \tag{6}$$

$$D \geq m\left(\frac{4c_4c_3}{c_2^2}\right)^2 \ln\frac{6m}{\delta} \tag{7}$$

$$D \geq m\left(\frac{4c_4c_3}{c_2^2}\right)^2 \cdot 2\ln\frac{2}{\epsilon} \tag{8}$$

then we have that for all $\gamma \geq 0$, for the $\mathcal{L}^{(\gamma)}$ loss, $\epsilon_{\text{unif-alg}}(m,\delta) \geq 1 - \epsilon_{gen}(m,\delta)$.

Specifically, for $\gamma \in [0,1]$, $\epsilon_{gen}(m,\delta) \leq \epsilon$, and so $\epsilon_{\text{unif-alg}}(m,\delta) \geq 1 - \epsilon$.

Proof. The above follows from Lemma D.1, where we upper bound the generalization error, and from Lemma D.2 where we lower bound uniform convergence. □

We first prove that the above algorithm generalizes well with respect to the losses corresponding to $\gamma \in [0,1]$. First for the training data, we argue that both $\mathbf{w}_1$ and a small part of the noise vector $\mathbf{w}_2$ align along the correct direction, while the remaining part of the high-dimensional noise vector are orthogonal to the input; this leads to correct classification of the training set. Then, on the test data, we argue that $\mathbf{w}_1$ aligns well, while $\mathbf{w}_2$ contributes very little to the output of the classifier because it is high-dimensional noise. As a result, for most test data, the classification is correct, and hence the test and generalization error are both small.

Lemma D.1. *In the setup of Section 5, when $\gamma \in [0,1]$, for $\mathcal{L}^{(\gamma)}$, $\epsilon_{gen}(m,\delta) \leq \epsilon$.*

Proof. The parameters learned by our algorithm satisfies $\mathbf{w}_1 = 2 \cdot \mathbf{u}$ and $\mathbf{w}_2 = \sum y^{(i)}\mathbf{x}_2^{(i)} \sim \mathcal{N}(0, \frac{8m}{c_2^2 D})$.

First, we have from Corollary C.1.1 that with probability $1 - \frac{\delta}{3m}$ over the draws of $\mathbf{x}_2^{(i)}$, as long as $\frac{\delta}{3m} \geq 2e^{-c_1 D}$

(which is given to hold by Equation 6),

$$c_2 \leq \frac{1}{2\sqrt{2}} c_2 \|\mathbf{x}_2^{(i)}\| \leq c_3 \tag{9}$$

Next, for a given $\mathbf{x}^{(i)}$, we have from Corollary C.2.1, with probability $1 - \frac{\delta}{3m}$ over the draws of $\sum_{j \neq i} y^{(j)} \mathbf{x}_2^{(j)}$,

$$|\mathbf{x}_2^{(i)} \cdot \sum_{j \neq i} y^{(j)} \mathbf{x}_2^{(j)}| \leq c_4 \|\mathbf{x}_2^{(i)}\| \frac{2\sqrt{2} \cdot \sqrt{m}}{c_2 \sqrt{D}} \sqrt{\ln \frac{6m}{\delta}} \tag{10}$$

Then, with probability $1 - \frac{2}{3}\delta$ over the draws of the training dataset we have for all i:

$$\begin{aligned}
&y^{(i)} h(\mathbf{x}^{(i)}) \\
&= y^{(i)} \mathbf{w}_1 \cdot \mathbf{x}_1^{(i)} + y^{(i)} \cdot y^{(i)} \|\mathbf{x}_2^{(i)}\|^2 + y^{(i)} \cdot \mathbf{x}_2^{(i)} \cdot \sum_{j \neq i} y^{(j)} \mathbf{x}_2^{(j)} \\
&= 4 + \underbrace{\|\mathbf{x}_2^{(i)}\|^2}_{\text{apply Equation 9}} + \underbrace{y^{(i)} \mathbf{x}_2^{(i)} \cdot \sum_{j \neq i} y^{(j)} \mathbf{x}_2^{(j)}}_{\text{apply Equation 10}} \\
&\geq 4 + 4 \cdot 2 \\
&- c_4 \frac{2\sqrt{2} c_3}{c_2} \cdot \underbrace{\frac{2\sqrt{2} \cdot \sqrt{m}}{c_2 \sqrt{D}} \sqrt{\ln \frac{6m}{\delta}}}_{\text{apply Equation 7}} \\
&\geq 4 + 8 - 2 = 10 > 1
\end{aligned} \tag{11}$$

Thus, for all $\gamma \in [0, 1]$, the $\mathcal{L}^{(\gamma)}$ loss of this classifier on the training dataset S is zero.

Now, from Corollary C.1.1, with probability $1 - \frac{\delta}{3}$ over the draws of the training data, we also have that, as long as $\frac{\delta}{3m} \geq 2e^{-c_1 D}$ (which is given to hold by Equation 6),

$$c_2 \sqrt{m} \leq \frac{1}{2\sqrt{2}} c_2 \|\sum y^{(i)} \mathbf{x}_2^{(i)}\| \leq c_3 \sqrt{m} \tag{12}$$

Next, conditioned on the draw of S and the learned classifier, for any $\epsilon' > 0$, with probability $1 - \epsilon'$ over the draws of a test data point, $(\mathbf{x}, y)$, we have from Corollary C.2.1 that:

$$|\mathbf{x}_2 \cdot \sum y^{(i)} \mathbf{x}_2^{(i)}| \leq c_4 \|\sum y^{(i)} \mathbf{x}_2^{(i)}\| \cdot \frac{2\sqrt{2}}{c_2 \sqrt{D}} \cdot \ln \frac{1}{\epsilon'} \tag{13}$$

Using this, we have that with probability $1 - 2\exp\left(-\frac{1}{2}\left(\frac{c_2^2}{4c_4 c_3}\sqrt{\frac{D}{m}}\right)^2\right)$ over the draws of a test data point, $(\mathbf{x}, y)$:

$$\begin{aligned}
&yh(\mathbf{x}) \\
&= y\mathbf{w}_1 \cdot \mathbf{x}_1 + \underbrace{y \cdot \mathbf{x}_2 \cdot \sum_j y^{(j)} \mathbf{x}_2^{(j)}}_{\text{apply Equation 13}} \\
&\geq 4 - c_4 \underbrace{\|\sum y^{(i)} \mathbf{x}_2^{(i)}\|}_{\text{apply Equation 12}} \cdot \frac{2\sqrt{2}}{c_2 \sqrt{D}} \frac{c_2^2}{4c_4 c_3} \sqrt{\frac{D}{m}} \\
&\geq 4 - 2 \geq 2
\end{aligned} \tag{14}$$

Thus, we have that for $\gamma \in [0, 1]$, the $\mathcal{L}^{(\gamma)}$ loss of the classifier on the distribution $\mathcal{D}$ is $2\exp\left(-\frac{1}{2}\left(\frac{c_2^2}{4c_4 c_3}\sqrt{\frac{D}{m}}\right)^2\right)$ which is at most ϵ as assumed in Equation 8. In other words, the absolute difference between the distribution loss and the train loss is at most ϵ and this holds for at least $1 - \delta$ draws of the samples S. Then, by the definition of ϵ_{gen} we have the result.

□

We next prove our uniform convergence lower bound. The main idea is that when the noise vectors in the training samples are negated, with high probability, the classifier misclassifies the training data. We can then show that for any choice of $\mathcal{S}_\delta$ as required by the definition of $\epsilon_{\text{unif-alg}}$, we can always find an $S_\star$ and its noise-negated version $S'_\star$ both of which belong to $\mathcal{S}_\delta$. Furthermore, we can show that $h_{S_\star}$ has small test error but high empirical error on $S'_\star$, and that this leads to a nearly vacuous uniform convergence bound.

Lemma D.2. *In the setup of Section 5, for any $\epsilon > 0$ and for any $\delta \leq 1/4$, and for the same lower bounds on D, and for any $\gamma \geq 0$, we have that:*

$$\epsilon_{\textit{unif-alg}}(m, \delta) \geq 1 - \epsilon_{\textit{gen}}(m, \delta)$$

for the $\mathcal{L}^{(\gamma)}$ loss.

Proof. For any S, let S' denote the set of noise-negated samples $S' = \{(\mathbf{x}_1, -\mathbf{x}_2) \mid (\mathbf{x}_1, \mathbf{x}_2) \in S\}$. We first show with high probability $1 - 2\delta/3$ over the draws of S, that the classifier learned on S, misclassifies S' completely. The proof for this is nearly identical to our proof for why the training loss is zero, except for certain sign changes. For any $\mathbf{x}_{\text{neg}}^{(i)} = (\mathbf{x}_1^{(i)}, -\mathbf{x}_2^{(i)})$, we have:

$$\begin{aligned}
&y^{(i)} h(\mathbf{x}_{\text{neg}}^{(i)}) \\
&= y^{(i)} \mathbf{w}_1 \cdot \mathbf{x}_1^{(i)} - y^{(i)} \cdot y^{(i)} \|\mathbf{x}_2^{(i)}\|^2 - y^{(i)} \cdot \mathbf{x}_2^{(i)} \cdot \sum_{j \neq i} y^{(j)} \mathbf{x}_2^{(j)} \\
&= 4 - \underbrace{\|\mathbf{x}_2^{(i)}\|^2}_{\text{apply Equation 9}} - \underbrace{y^{(i)} \mathbf{x}_2^{(i)} \cdot \sum_{j \neq i} y^{(j)} \mathbf{x}_2^{(j)}}_{\text{apply Equation 10}}
\end{aligned}$$

$$\leq 4 - 4 \cdot 2$$
$$+ c_4 \frac{2\sqrt{2}c_3}{c_2} \cdot \underbrace{\frac{2\sqrt{2} \cdot \sqrt{m}}{c_2\sqrt{D}} \ln \frac{3m}{\delta}}_{\text{apply Equation 7}}$$
$$\leq 4 - 8 + 2 = -2 < 0$$

Since the learned hypothesis misclassifies all of S', it has loss of 1 on S'.

Now recall that, by definition, to compute $\epsilon_{\text{unif-alg}}$, one has to pick a sample set space $\mathcal{S}_\delta$ of mass $1-\delta$ i.e., $Pr_{S\sim\mathcal{S}^m}[S \in \mathcal{S}_\delta] \geq 1-\delta$. We first argue that for *any* choice of $\mathcal{S}_\delta$, there must exist a 'bad' $S_\star$ such that (i) $S_\star \in \mathcal{S}_\delta$, (ii) $S'_\star \in \mathcal{S}_\delta$, (iii) $h_{S_\star}$ has test error less than $\epsilon_{\text{gen}}(m,\delta)$ and (iv) $h_{S_\star}$ completely misclassifies $S'_\star$.

We show the existence of such an $S_\star$, by arguing that over the draws of S, there is non-zero probability of picking an S that satisfies all the above conditions. Specifically, we have by the union bound that:

$$\begin{aligned} &Pr_{S\sim\mathcal{D}^m}\big[S \in \mathcal{S}_\delta, S' \in \mathcal{S}_\delta, \\ &\quad \mathcal{L}_\mathcal{D}(h_S) \leq \epsilon_{\text{gen}}(m,\delta), \hat{\mathcal{L}}_{S'}(h_S) = 1\big] \geq \\ &1 - Pr_{S\sim\mathcal{D}^m}[S \notin \mathcal{S}_\delta] - Pr_{S\sim\mathcal{D}^m}[S' \notin \mathcal{S}_\delta] \\ &- Pr_{S\sim\mathcal{D}^m}[\mathcal{L}_\mathcal{D}(h_S) > \epsilon_{\text{gen}}(m,\delta)] \\ &- Pr_{S\sim\mathcal{D}^m}\left[\hat{\mathcal{L}}_{S'}(h_S) \neq 1\right] \end{aligned} \tag{15}$$

By definition of $\mathcal{S}_\delta$, we know $Pr_{S\sim\mathcal{D}^m}[S \notin \mathcal{S}_\delta] \leq \delta$. Similarly, by definition of the generalization error, we know that $Pr_{S\sim\mathcal{D}^m}[\mathcal{L}_\mathcal{D}(h_S) > \epsilon_{\text{gen}}(m,\delta)] \leq \delta$. We have also established above that $Pr_{S\sim\mathcal{D}^m}\left[\hat{\mathcal{L}}_{S'}(h_S) \neq 1\right] \leq 2\delta/3$. As for the term $Pr_{S\sim\mathcal{D}^m}[S' \notin \mathcal{S}_\delta]$, observe that under the draws of S, the distribution of the noise-negated dataset S' is identical to $\mathcal{D}^m$. This is because the isotropic Gaussian noise vectors have the same distribution under negation. Hence, again by definition of $\mathcal{S}_\delta$, even this probability is at most δ. Thus, we have that the probability in the left hand side of Equation 15 is at least $1-4\delta$, which is positive as long as $\delta < 1/4$.

This implies that for any given choice of $\mathcal{S}_\delta$, there exists $S_\star$ that satisfies our requirement. Then, from the definition of $\epsilon_{\text{unif-alg}}(m,\delta)$, we essentially have that:

$$\begin{aligned} \epsilon_{\text{unif-alg}}(m,\delta) &= \sup_{S\in\mathcal{S}_\delta} \sup_{h\in\mathcal{H}_\delta} |\mathcal{L}_\mathcal{D}(h) - \hat{\mathcal{L}}_S(h)| \\ &\geq |\mathcal{L}_\mathcal{D}(h_{S_\star}) - \hat{\mathcal{L}}_{S'_\star}(h)| = |\epsilon - 1| = 1 - \epsilon, \end{aligned}$$

□

D.1. Further Remarks.

As seen in the proof of Lemma D.1, the generalization error ϵ depends on m and D as $\mathcal{O}(e^{-D/m})$ ignoring some constants in the exponent. Clearly, this error decreases with the parameter count D.

On the other hand, we also note that the generalization error grows with the number of samples m, which might at first make this model seem inconsistent with our real world observations. However, we emphasize that this is a minor artefact of the simplifications in our setup, rather than a conceptual issue. With a small modification to our setup, we can make the generalization error decrease with m, mirroring our empirical observations. Specifically, in the current setup, we learn the true boundary along the first K dimensions exactly. We can however modify it to a more standard learning setup where the boundary is not exactly recoverable and needs to be estimated from the examples. This would lead to an additional generalization error that scales as $\mathcal{O}(\sqrt{\frac{K}{m}})$ that is non-vacuous as long as $K \ll m$. Thus, the overall generalization error would be $\mathcal{O}(e^{-D/m} + \sqrt{\frac{K}{m}})$. Note that even in this model, the generalization error decreases with the parameter count.

What about the overall dependence on m? Now, assume we have an overparameterization level of $D \gg m\ln(m/K)$, so that $e^{-D/m} \ll \sqrt{K/m}$. Hence, in the sufficiently overparameterized regime, the generalization error $\mathcal{O}(e^{-D/m})$ that comes from the noise we have modeled, pales in comparison with the generalization error that would stem from estimating the low-complexity boundary. Overall, as a function of m, the resulting error would behave like $\mathcal{O}(\sqrt{\frac{K}{m}})$ and hence show a decrease with increasing m (as long the increase in m is within the overparameterized regime).

E. A more complex model & a low-dimensional distribution

Recall that our main setup consisted of a linear model; in this setup, our data distribution was a high-dimensional distribution. In this section, we argue that the dimensionality of the input distribution is not crucial to our argument. In particular, if the model could afford more complex boundaries, the data distribution can be low-dimensional (even 1D) and one can still impair uniform convergence.

Consider the following setup which, although unconventional in some ways, is useful to convey the essence of our argument. Let the underlying distribution over the inputs be a spherical Gaussian in $\mathbb{R}^D$ where D can be however small or large as the reader desires. Note that our setup would apply to many other distributions, but a Gaussian would make our discussion easier. Let the labels of the inputs be determined by some $h^\star : \mathbb{R}^D \to \{-1,+1\}$. Consider a scenario where the learning algorithm outputs a very slightly modified version of $h^\star$. Specifically, let $S' = \{-\mathbf{x} \mid \mathbf{x} \in S\}$;

then, the learner outputs:

$$h_S(\mathbf{x}) = \begin{cases} -h^\star(\mathbf{x}) & \text{if } \mathbf{x} \in S' \\ h^\star(\mathbf{x}) & \text{otherwise} \end{cases}$$

That is, the learner misclassifies inputs that correspond to the negations of the samples in the training data – this would be possible if and only if the classifier is overparameterized with $\Omega(mD)$ parameters to store S'. We will show that uniform convergence fails to explain generalization for this learner.

First we establish that this learner generalizes well. Note that a given S has zero probability mass under $\mathcal{D}$, and so does S'. Then, the training and test error are zero – except for pathological draws of S that intersect with S', which are almost surely never drawn from $\mathcal{D}^m$ – and hence, the generalization error of $\mathcal{A}$ is zero too.

It might thus seem reasonable to expect that one could explain this generalization using implicit-regularization-based uniform convergence by showing $\epsilon_{\text{unif-alg}}(m, \delta) = 0$. Surprisingly, this is not the case as $\epsilon_{\text{unif-alg}}(m, \delta)$ is in fact 1!

First it is easy to see why the looser bound $\epsilon_{\text{unif}}(m, \delta)$ equals 1, if we let $\mathcal{H}$ be the space of all hypotheses the algorithm could output: there must exist a non-pathological $S \in \mathcal{S}_\delta$, and we know that $h_{S'} \in \mathcal{H}$ misclassifies the negation of its training set, namely S. Then, $\sup_{h \in \mathcal{H}} |\mathcal{L}_\mathcal{D}(h) - \hat{\mathcal{L}}_S(h)| = |\mathcal{L}_\mathcal{D}(h_{S'}) - \hat{\mathcal{L}}_S(h_{S'})| = |0 - 1| = 1$.

One might hope that in the stronger bound of $\epsilon_{\text{unif-alg}}(m, \delta)$ since we truncate the hypothesis space, it is possible that the above adversarial situation would fall apart. However, with a more nuanced argument, we can similarly show that $\epsilon_{\text{unif-alg}}(m, \delta) = 1$. First, recall that any bound on $\epsilon_{\text{unif-alg}}(m, \delta)$, would have to pick a truncated sample set space $\mathcal{S}_\delta$. Consider any choice of $\mathcal{S}_\delta$, and the corresponding set of explored hypotheses $\mathcal{H}_\delta$. We will show that for any choice of $\mathcal{S}_\delta$, there exists $S_\star \in \mathcal{S}_\delta$ such that (i) $h_{S_\star}$ has zero test error and (ii) the negated training set $S'_\star$ belongs to $\mathcal{S}_\delta$ and (iii) $h_{S_\star}$ has error 1 on $S_\star$. Then, it follows that $\epsilon_{\text{unif-alg}}(m, \delta) = \sup_{S \in \mathcal{S}_\delta} \sup_{h \in \mathcal{H}_\delta} |\mathcal{L}_\mathcal{D}(h) - \hat{\mathcal{L}}_S(h)| \geq |\mathcal{L}_\mathcal{D}(h_{S_\star}) - \hat{\mathcal{L}}_{S'_\star}(h)| = |0 - 1| = 1$.

We can prove the existence of such an $S_\star$ by showing that the probability of picking one such set under $\mathcal{D}^m$ is non-zero for $\delta < 1/2$. Specifically, under $S \sim \mathcal{D}^m$, we have by the union bound that:

$$Pr\left[\mathcal{L}_\mathcal{D}(h_S) = 0, \hat{\mathcal{L}}_{S'}(h_S) = 1, S \in \mathcal{S}_\delta, S' \in \mathcal{S}_\delta\right] \geq$$
$$1 - Pr\left[\mathcal{L}_\mathcal{D}(h_S) \neq 0, \hat{\mathcal{L}}_{S'}(h_S) \neq 1\right] - Pr\left[S \notin \mathcal{S}_\delta\right]$$
$$- Pr\left[S' \notin \mathcal{S}_\delta\right]$$

Since the pathological draws have probability zero, the first probability term on the right hand side is zero. The second term is at most δ by definition of $\mathcal{S}_\delta$. Crucially, the last term too is at most δ because S' (which is the negated version of S) obeys the same distribution as S (since the isotropic Gaussian is invariant to a negation). Thus, the above probability is at least $1 - 2\delta > 0$, implying that there exist (many) $S_\star$, proving our main claim.

Remark. While our particular learner might seem artificial, much of this artificiality is only required to make the argument simple. The crucial trait of the learner that we require is that the misclassified region in the input space (i) covers low probability and yet (ii) is complex and highly dependent on the training set draw. Our intuition is that SGD-trained deep networks possess these traits.

F. Deep Learning Conjecture

In this section, we present the experiments that we alluded to in the main paper in the context of our deep learning conjecture. Specifically, we present experiments that suggest why the parameters of a deep network could be decomposed into a low complexity component, and a high-dimensional noisy component that does not dominate the output of the network. We illustrate this by considering the singular value decomposition of the update matrices of the network $W_d - Z_d$ (where, recall that Z_d is the random initialization).

Specifically, we take a network of width $h = 1024$ and $d = 6$ trained to classify 99% of a 32768 MNIST examples by a margin of 10; we then remove many of the lower singular directions of all the update matrices $W_d - Z_d$ (except the output layer which is already a matrix of rank 10). On one hand, we observe (see Figure 13 left) that the (0-1) error of the network on the test and training data is able to endure this pruning even until only as many as 16 top directions remain (which constitute the 'low complexity' component).

On the other hand, we also observe (see Figure 13 right) that the spectral norm of the matrix that is removed can be quite high, and yet, removing it does not affect the output of the network on most inputs. In particular, if we consider the case where we retain only 16 top directions, the spectral norms of the matrices removed from the first layer is as large as 5.5. Since the inputs have an ℓ_2 norm of roughly 10 to 15, the change in the output of the network under the removal of these singular directions, could be at least as large as 55 to 82.5 in the worst case. This could potentially misclassify a significant proportion of the datapoints that were originally classified by a margin of roughly 10 to 30. Yet, we see that the effect of this removal is hardly seen on most inputs, indicating that the huge number of singular directions that we removed could be seen as our conjectured 'high-dimensional noisy component' that does not dominate the network's output.

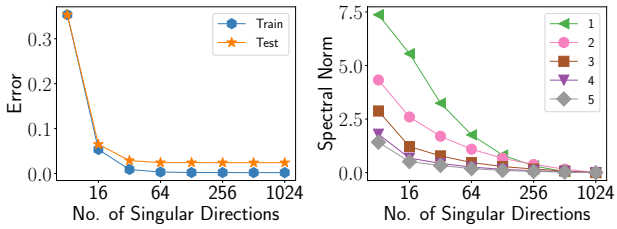

Figure 13. On the **left**, we plot the $0-1$ test and train error of the network for different numbers of singular directions that are retained across all the matrices of the network. We see that even after reducing the update matrices to a rank of 16, the network has a test error of only 0.1. On the **right**, we plot the spectral norm of the matrices that we removed from each layer, for the different number of singular directions that we retain. We observe that for a rank of 16, the spectral norm of the removed matrix of the lowermost layer can be as large as 5.5.

G. Learnability and Uniform Convergence

Below, we provide a detailed discussion on learnability, uniform convergence and generalization. Specifically, we argue why the fact that uniform convergence is necessary for learnability does not preclude the fact that uniform convergence maybe unable to *explain* generalization of a particular algorithm for a particular distribution.

We first recall the notion of learnability. First, formally, a binary classification problem consists of a hypothesis class $\mathcal{H}$ and an instance space $\mathcal{X}\times\{-1,1\}$. The problem is said to be *learnable* if there exists a learning rule $\mathcal{A}' : \bigcup_{m=1}^{\infty} \mathcal{Z}^m \to \mathcal{H}$ and a monotonically decreasing sequence $\epsilon_{\text{lnblty}}(m)$ such that $\epsilon_{\text{lnblty}}(m) \xrightarrow{m\to\infty} 0$ and

$$\forall \mathcal{D}' \; \mathbb{E}_{S\sim\mathcal{D}'^m}\left[\mathcal{L}^{(0)}_{\mathcal{D}'}(\mathcal{A}'(S)) - \min_{h\in\mathcal{H}} \mathcal{L}^{(0)}_{\mathcal{D}'}(h)\right] \le \epsilon_{\text{lnblty}}(m). \tag{16}$$

Vapnik & Chervonenkis (1971) showed that finite VC dimension of the hypothesis class is necessary and sufficient for learnability in binary classification problems. As Shalev-Shwartz et al. (2010) note, since finite VC dimension is equivalent to uniform convergence, it can thus be concluded that uniform convergence is necessary and sufficient for learnability binary classification problems.

However, learnability is a strong notion that does not necessarily have to hold for a particular learning algorithm to generalize well for a particular underlying distribution. Roughly speaking, this is because learnability evaluates the algorithm under all possible distributions, including many complex distributions; while a learning algorithm may generalize well for a particular distribution under a given hypothesis class, it may fail to do so on more complex distributions under the same hypothesis class.

For more intuition, we present a more concrete but informal argument below. However, this argument is technically redundant because learnability is equivalent to uniform convergence for binary classification, and since we established the lack of necessity of uniform convergence, we effectively established the same for learnability too. However, we still provide the following informal argument as it provides a different insight into why learnability and uniform convergence are not necessary to explain generalization.

Our goal is to establish that in the set up of Section 5, even if we considered the binary classification problem corresponding to $\mathcal{H}_\delta$ (the class consisting of only those hypotheses explored by the algorithm $\mathcal{A}$ under a distribution $\mathcal{D}$), the corresponding binary classification problem is not learnable i.e., Equation 16 does not hold when we plug in $\mathcal{H}_\delta$ in place of $\mathcal{H}$.

First consider distributions of the following form that is more complex than the linearly separable $\mathcal{D}$: for any dataset S', let $\mathcal{D}_{S'}$ be the distribution that has half its mass on the part of the linearly separable distribution $\mathcal{D}$ excluding S', and half its mass on the distribution that is uniformly distributed over S'. Now let S' be a random dataset drawn from $\mathcal{D}$ *but with all its labels flipped*; consider the corresponding complex distribution $\mathcal{D}_{S'}$.

We first show that there exists $h \in \mathcal{H}_\delta$ that fits this distribution well. Now, for most draws of the "wrongly" labeled S', we can show that the hypothesis h for which $\mathbf{w}_1 = 2\cdot\mathbf{u}$ and $\mathbf{w}_2 = \sum_{(x,y)\in S'} y\cdot\mathbf{x}_2$ fits the "wrong" labels of S' perfectly; this is because, just as argued in Lemma D.2, $\mathbf{w}_2$ dominates the output on all these inputs, although $\mathbf{w}_1$ would be aligned incorrectly with these inputs. Furthermore, since $\mathbf{w}_2$ does not align with most inputs from $\mathcal{D}$, by an argument similar to Lemma D.1, we can also show that this hypothesis has at most ϵ error on $\mathcal{D}$, and that this hypothesis belongs to $\mathcal{H}_\delta$. Overall this means that, w.h.p over the choice of S', there exists a hypothesis $h \in \mathcal{H}_\delta$ for which the error on the complex distribution $\mathcal{D}_{S'}$ is at most $\epsilon/2$ i.e.,

$$\min_{h\in\mathcal{H}} \mathbb{E}_{(x,y)\sim\mathcal{D}_{S'}}[\mathcal{L}(h(x),y)] \le \epsilon/2$$

On the other hand, let $\mathcal{A}'$ be any learning rule which outputs a hypothesis given $S \sim \mathcal{D}_{S'}$. With high probability over the draws of $S \sim \mathcal{D}_{S'}$, only at most, say $3/4$th of S (i.e., $0.75m$ examples) will be sampled from S' (and the rest from $\mathcal{D}$). Since the learning rule which has access only to S, has not seen at least a quarter of S', with high probability over the random draws of S', the learning rule will fail to classify roughly half of the unseen examples from S' correctly (which would be about $(m/4)\cdot 1/2 = m/8$).

Then, the error on $\mathcal{D}_{S'}$ will be at least $1/16$. From the above arguments, we have that $\epsilon_{\text{learnability}}(m) \geq 1/16 - \epsilon/2$, which is a non-negligible constant that is independent of m.

H. Deterministic PAC-Bayes bounds are two-sided uniform convergence bounds

By definition, VC-dimension, Rademacher complexity and other covering number based bounds are known to upper bound the term $\epsilon_{\text{unif-alg}}$ and therefore our negative result immediately applies to all these bounds. However, it may not be immediately clear if bounds derived through the PAC-Bayesian approach fall under this category too. In this discussion, we show that existing deterministic PAC-Bayes based bounds are in fact two-sided in that they are lower bounded by $\epsilon_{\text{unif-alg}}$ too.

For a given prior distribution P over the parameters, a PAC-Bayesian bound is of the following form: with high probability $1-\delta$ over the draws of the data S, we have that *for all distributions* Q over the hypotheses space:

$$KL\left(\mathbb{E}_{\tilde{h}\sim Q}[\hat{\mathcal{L}}_S(\tilde{h})] \middle\| \mathbb{E}_{\tilde{h}\sim Q}[\mathcal{L}_{\mathcal{D}}(\tilde{h})]\right) \leq \underbrace{\frac{KL(Q\|P) + \ln\frac{2m}{\delta}}{m-1}}_{:=\epsilon_{\text{pb}}(P,Q,m,\delta)} \tag{17}$$

Note that here for any $a, b \in [0,1]$, $KL(a\|b) = a\ln\frac{a}{b} + (1-a)\ln\frac{1-a}{1-b}$. Since the precise form of the PAC-Bayesian bound on the right hand side is not relevant for the rest of the discussion, we will concisely refer to it as $\epsilon_{\text{pb}}(P,Q,m,\delta)$. What is of interest to us is the fact that the above bound holds for all Q for most draws of S and that the KL-divergence on the right-hand side is in itself two-sided, in some sense.

Typically, the above bound is simplified to derive the following one-sided bound on the difference between the expected and empirical errors of a stochastic network (see (McAllester, 2003) for example):

$$\mathbb{E}_{\tilde{h}\sim Q}[\mathcal{L}_{\mathcal{D}}(\tilde{h})] - \mathbb{E}_{\tilde{h}\sim Q}[\hat{\mathcal{L}}_S(\tilde{h})] \leq \sqrt{2\epsilon_{\text{pb}}(P,Q,m,\delta)} + 2\epsilon_{\text{pb}}(P,Q,m,\delta) \tag{18}$$

This bound is then manipulated in different ways to obtain bounds on the deterministic network. In the rest of this discussion, we focus on the two major such derandomizing techniques and argue that both these techniques boil down to two-sided convergence. While, we do not formally establish that there may exist other techniques which ensure that the resulting deterministic bound is strictly one-sided, we suspect that no such techniques may exist. This is because the KL-divergence bound in Equation 17 is in itself two-sided in the sense that for the right hand side bound to be small, both the stochastic test and train errors must be close to each other; it is not sufficient if the stochastic test error is smaller than the stochastic train error.

H.1. Deterministic PAC-Bayesian Bounds of Type A

To derive a deterministic generalization bound, one approach is to add extra terms that account for the perturbation in the loss of the network (Neyshabur et al., 2017; McAllester, 2003; Nagarajan & Kolter, 2019). That is, define:

$$\Delta(h,Q,\mathcal{D}) = |\mathcal{L}_{\mathcal{D}}(h) - \mathbb{E}_{\tilde{h}\sim Q}[\mathcal{L}_{\mathcal{D}}(\tilde{h})]|$$
$$\Delta(h,Q,S) = \left|\hat{\mathcal{L}}_S(h) - \mathbb{E}_{\tilde{h}\sim Q}[\hat{\mathcal{L}}_S(\tilde{h})]\right|$$

Then, one can get a deterministic upper bound as:

$$\mathcal{L}_{\mathcal{D}}(h) - \hat{\mathcal{L}}_S(h) \leq \sqrt{2\epsilon_{\text{pb}}(P,Q_h,m,\delta)} + 2\epsilon_{\text{pb}}(P,Q_h,m,\delta) + \Delta(h,Q,\mathcal{D}) + \Delta(h,Q,S)$$

Note that while applying this technique, for any hypothesis h, one picks a posterior Q_h specific to that hypothesis (typically, centered at that hypothesis).

We formally define the deterministic bound resulting from this technique below. We consider the algorithm-dependent version and furthermore, we consider a bound that results from the best possible choice of Q_h for all h. We define this deterministic bound in the format of $\epsilon_{\text{unif-alg}}$ as follows:

Definition H.1. The distribution-dependent, algorithm-dependent, deterministic PAC-Bayesian bound of (the hypothesis class $\mathcal{H}$, algorithm $\mathcal{A}$)-pair with respect to $\mathcal{L}$ is defined to be the smallest value $\epsilon_{\text{pb-det-A}}(m,\delta)$ such that the following holds:

1. there exists a set of m-sized samples $\mathcal{S}_\delta \subseteq (\mathcal{X}\times\{-1,+1\})^m$ for which:
$$Pr_{S\sim\mathcal{D}^m}[S \notin \mathcal{S}_\delta] \leq \delta$$
2. and if we define $\mathcal{H}_\delta = \bigcup_{S\in\mathcal{S}_\delta}\{h_S\}$ to be the space of hypotheses explored only on these samples, then there must exist a prior P and for each $h \in \mathcal{H}_\delta$, a distribution Q_h, such that uniform convergence must hold as follows:
$$\sup_{S\in\mathcal{S}_\delta}\sup_{h\in\mathcal{H}_\delta}\sqrt{2\epsilon_{\text{pb}}(P,Q_h,m,\delta)} + 2\epsilon_{\text{pb}}(P,Q_h,m,\delta)$$

$$+ \Delta(h, Q_h, \mathcal{D}) + \Delta(h, Q_h, S) < \epsilon_{\text{pb-det-A}}(m, \delta) \tag{19}$$

as a result of which, by Equation 18, the following one-sided uniform convergence also holds:

$$\sup_{S \in \mathcal{S}_\delta} \sup_{h \in \mathcal{H}_\delta} \mathcal{L}_\mathcal{D}(h) - \hat{\mathcal{L}}_S(h) < \epsilon_{\text{pb-det-A}}(m, \delta) \tag{20}$$

Now, recall that $\epsilon_{\text{unif-alg}}(m, \delta)$ is a two-sided bound, and in fact our main proof crucially depended on this fact in order to lower bound $\epsilon_{\text{unif-alg}}(m, \delta)$. Hence, to extend our lower bound to $\epsilon_{\text{pb-det-A}}(m, \delta)$ we need to show that it is also two-sided in that it is lower bounded by $\epsilon_{\text{unif-alg}}(m, \delta)$. The following result establishes this:

Theorem H.1. *Let $\mathcal{A}$ be an algorithm such that on at least $1 - \delta$ draws of the training dataset S, the algorithm outputs a hypothesis h_S that has $\hat{\epsilon}(m, \delta)$ loss on the training data S. Then*

$$e^{-3/2} \cdot \epsilon_{\textit{unif-alg}}(m, 3\delta) - (1 - e^{-3/2})(\hat{\epsilon}(m, \delta) + \epsilon_{\textit{gen}}(m, \delta)) \leq \epsilon_{\textit{pb-det-A}}(m, \delta)$$

Proof. First, by the definition of the generalization error, we know that with probability at least $1 - \delta$ over the draws of S:

$$\mathcal{L}_D(h_S) \leq \hat{\mathcal{L}}_S(h_S) + \epsilon_{\text{gen}}(m, \delta).$$

Furthermore since the training loss it at most $\hat{\epsilon}(m, \delta)$ on at least $1 - \delta$ draws we have that on at least $1 - 2\delta$ draws of the dataset:

$$\mathcal{L}_D(h_S) \leq \hat{\epsilon}(m, \delta) + \epsilon_{\text{gen}}(m, \delta)$$

Let $\mathcal{H}_\delta$ and $\mathcal{S}_\delta$ be the subset of hypotheses and sample sets as in the definition of $\epsilon_{\text{pb-det-A}}$. Then, from the above, there exist $\mathcal{H}_{3\delta} \subseteq \mathcal{H}_\delta$ and $\mathcal{S}_{3\delta} \subseteq \mathcal{S}_\delta$ such that

$$Pr_{S \sim \mathcal{D}^m}[S \notin \mathcal{S}_{3\delta}] \leq 3\delta$$

and $\mathcal{H}_{3\delta} = \bigcup_{S \in \mathcal{S}_{3\delta}} \{h_S\}$, and furthermore:

$$\sup_{h \in \mathcal{H}_{3\delta}} \mathcal{L}_\mathcal{D}(h) \leq \hat{\epsilon}(m, \delta) + \epsilon_{\text{gen}}(m, \delta)$$

.

Using the above, and the definition of Δ, we have for all $h \in \mathcal{H}_{3\delta}$, the following upper bound on its stochastic test error:

$$\begin{aligned}\mathbb{E}_{\tilde{h} \sim Q_h}[\mathcal{L}_\mathcal{D}(\tilde{h})] &\leq \mathcal{L}_\mathcal{D}(h) + \Delta(h, Q_h, \mathcal{D}) \\ &\leq \hat{\epsilon}(m, \delta) + \epsilon_{\text{gen}}(m, \delta) + \underbrace{\Delta(h, Q_h, \mathcal{D})}_{\text{applying Equation 19}} \\ &\leq \hat{\epsilon}(m, \delta) + \epsilon_{\text{gen}}(m, \delta) + \epsilon_{\text{pb-det-A}}(m, \delta)\end{aligned} \tag{21}$$

.

Now, for each pair of $h \in \mathcal{H}_{3\delta}$ and $S \in \mathcal{S}_{3\delta}$, we will bound its empirical error minus the expected error in terms of $\epsilon_{\text{pb-det-A}}(m, \delta)$. For convenience, let us denote by $a := \mathbb{E}_{\tilde{h} \sim Q_h}[\hat{\mathcal{L}}_S(\tilde{h})]$ and $b := \mathbb{E}_{\tilde{h} \sim Q_h}[\mathcal{L}_\mathcal{D}(\tilde{h})]$ (note that a and b are terms that depend on a hypothesis h and a sample set S).

We consider two cases. First, for some $h \in \mathcal{H}_{3\delta}$ and $S \in \mathcal{S}_{3\delta}$, consider the case that $e^{3/2} b > a$. Then, we have:

$$\begin{aligned}&\hat{\mathcal{L}}_S(h) - \mathcal{L}_\mathcal{D}(h) \\ \leq& a - b + \underbrace{\Delta(h, Q_h, \mathcal{D}) + \Delta(h, Q_h, S)}_{\text{applying Equation 19}} \\ \leq& (e^{3/2} - 1) \underbrace{b}_{\text{apply Equation 21}} + \epsilon_{\text{pb-det-A}}(m, \delta) \\ \leq& (e^{3/2} - 1)(\hat{\epsilon}(m, \delta) + \epsilon_{\text{gen}}(m, \delta) + \epsilon_{\text{pb-det-A}}(m, \delta)) \\ &+ \epsilon_{\text{pb-det-A}}(m, \delta) \\ \leq& (e^{3/2} - 1)(\hat{\epsilon}(m, \delta) + \epsilon_{\text{gen}}(m, \delta)) + e^{3/2} \cdot \epsilon_{\text{pb-det-A}}(m, \delta)\end{aligned} \tag{22}$$

Now consider the case where $a > e^{3/2} b$. This means that $(1 - a) < (1 - b)$. Then, if we consider the PAC-Bayesian bound of Equation 17:

$$a \ln \frac{a}{b} + (1 - a) \ln \frac{1 - a}{1 - b} \leq \epsilon_{\text{pb}}(P, Q_h, m, \delta) \tag{23}$$

on the second term, we can apply the inequality $\ln x \geq \frac{(x-1)(x+1)}{2x} = \frac{1}{2}\left(x - \frac{1}{x}\right)$ which holds for $x \in [0, 1]$ to get:

$$\begin{aligned}&(1 - a) \ln \frac{1 - a}{1 - b} \geq \frac{1}{2}(1 - a)\left(\frac{1 - a}{1 - b} - \frac{1 - b}{1 - a}\right) \\ &= \left(\frac{(b - a)(2 - a - b)}{2(1 - b)}\right) \\ &\geq -(a - b)\left(\frac{(2 - a - b)}{2(1 - b)}\right) \geq -(a - b)\left(\frac{(2 - b)}{2(1 - b)}\right) \\ &\geq -\frac{(a - b)}{2}\left(\frac{1}{(1 - b)} + 1\right)\end{aligned}$$

Plugging this back in Equation 23, we have:

$$\epsilon_{\text{pb}}(P, Q_h, m, \delta) \geq \underbrace{a \ln \frac{a}{b}}_{\geq 3/2} - \frac{(a - b)}{2}\left(\frac{1}{(1 - b)} + 1\right)$$

$$\geq \frac{2a(1-b)-(a-b)}{2(1-b)} + \frac{b}{2}$$
$$\geq \frac{2a(1-b)-(a-b)}{2(1-b)} \geq \frac{a-2ab+b}{2(1-b)}$$
$$\geq \frac{a-2ab+ab}{2(1-b)} \geq \frac{a}{2} \geq \frac{a-b}{2}$$
$$\geq \frac{1}{2}\left(\hat{\mathcal{L}}_S(h) - \mathcal{L}_{\mathcal{D}}(h) - (\Delta(h,Q_h,\mathcal{D}) + \Delta(h,Q_h,S))\right)$$

Rearranging, we get:

$$\hat{\mathcal{L}}_S(h) - \mathcal{L}_{\mathcal{D}}(h)$$
$$\leq \underbrace{2\epsilon_{\text{pb}}(P,Q_h,m,\delta) + (\Delta(h,Q_h,\mathcal{D}) + \Delta(h,Q_h,S))}_{\text{Applying Equation 19}}$$
$$\leq \epsilon_{\text{pb-det-A}}(m,\delta) \tag{24}$$

Since, for all $h \in \mathcal{H}_{3\delta}$ and $S \in \mathcal{S}_{3\delta}$, one of Equations 22 and 24 hold, we have that:

$$\frac{1}{e^{3/2}}\left(\sup_{h\in\mathcal{H}_{3\delta}}\sup_{S\in\mathcal{S}_{3\delta}} \hat{\mathcal{L}}_S(h) - \mathcal{L}_{\mathcal{D}}(h)\right)$$
$$- \frac{(e^{3/2}-1)}{e^{3/2}}(\hat{\epsilon}(m,\delta) + \epsilon_{\text{gen}}(m,\delta)) \leq \epsilon_{\text{pb-det-A}}(m,\delta)$$

It follows from Equation 20 that the above bound holds good even after we take the absolute value of the first term in the left hand side. However, the absolute value is lower-bounded by $\epsilon_{\text{unif-alg}}(m,3\delta)$ (which follows from how $\epsilon_{\text{unif-alg}}(m,3\delta)$ is defined to be the smallest possible value over the choices of $\mathcal{H}_{3\delta}, \mathcal{S}_{3\delta}$).

□

As a result of the above theorem, we can show that $\epsilon_{\text{pb-det-A}}(m,\delta) = \Omega(1) - \mathcal{O}(\epsilon)$, thus establishing that, for sufficiently large D, even though the generalization error would be negligibly small, the PAC-Bayes based bound would be as large as a constant.

Corollary H.1.1. *In the setup of Section 5, for any $\epsilon, \delta > 0, \delta < 1/12$, when $D = \Omega\left(\max\left(m\ln\frac{3}{\delta}, m\ln\frac{1}{\epsilon}\right)\right)$, we have:*

$$e^{-3/2}\cdot(1-\epsilon) - (1-e^{-3/2})(\epsilon) \leq \epsilon_{\textit{pb-det-A}}(m,\delta)$$

Proof. The fact that $\epsilon_{\text{gen}}(m,\delta) \leq \epsilon$ follows from Theorem 5.1. Additionally, $\hat{\epsilon}(m,\delta) = 0$ follows from the proof of Theorem 5.1. Now, as long as $3\delta < 1/4$, and D is sufficiently large (i.e., in the lower bounds on D in Theorem 5.1, if we replace δ by 3δ), we have from Theorem 5.1 that $\epsilon_{\text{unif-alg}}(m,3\delta) > 1-\epsilon$. Plugging these in Theorem H.1, we get the result in the above corollary. □

H.2. Deterministic PAC-Bayesian Bounds of Type B

In this section, we consider another standard approach to making PAC-Bayesian bounds deterministic (Neyshabur et al., 2018; Langford & Shawe-Taylor, 2002). Here, the idea is to pick for each h a distribution Q_h such that for all $\mathbf{x}$:

$$\mathcal{L}^{(0)}(h(\mathbf{x}),y) \leq \mathbb{E}_{\tilde{h}\sim Q_h}[\mathcal{L}'^{(\gamma/2)}(\tilde{h}(\mathbf{x}),y)] \leq \mathcal{L}'^{(\gamma)}(h(\mathbf{x}),y)$$

where

$$\mathcal{L}'^{(\gamma)}(y,y') = \begin{cases} 0 & y\cdot y' \geq \gamma \\ 1 & \text{else} \end{cases}$$

Then, by applying the PAC-Bayesian bound of Equation 18 for the loss $\mathcal{L}'_{\gamma/2}$, one can get a deterministic upper bound as follows, without having to introduce the extra Δ terms:

$$\mathcal{L}^{(0)}_{\mathcal{D}}(h) - \hat{\mathcal{L}}^{(\gamma)}_S(h) \leq$$
$$\mathbb{E}_{\tilde{h}\sim Q_h}[\mathcal{L}'^{(\gamma/2)}(\tilde{h})] - \mathbb{E}_{\tilde{h}\sim Q_h}[\hat{\mathcal{L}}'^{(\gamma/2)}_S(\tilde{h})]$$
$$\leq \sqrt{2\epsilon_{\text{pb}}(P,Q_h,m,\delta)} + 2\epsilon_{\text{pb}}(P,Q_h,m,\delta)$$

We first define this technique formally:

Definition H.2. The distribution-dependent, algorithm-dependent, deterministic PAC-Bayesian bound of (the hypothesis class $\mathcal{H}$, algorithm $\mathcal{A}$)-pair is defined to be the smallest value $\epsilon_{\text{pb-det-B}}(m,\delta)$ such that the following holds:

1. there exists a set of m-sized samples $\mathcal{S}_\delta \subseteq (\mathcal{X}\times\{-1,+1\})^m$ for which:
$$Pr_{S\sim\mathcal{D}^m}[S \notin \mathcal{S}_\delta] \leq \delta$$
2. and if we define $\mathcal{H}_\delta = \bigcup_{S\in\mathcal{S}_\delta}\{h_S\}$ to be the space of hypotheses explored only on these samples, then there must exist a prior P and for each h a distribution Q_h, such that uniform convergence must hold as follows: for all $S \in \mathcal{S}_\delta$ and for all $h \in \mathcal{H}_\delta$

$$\sqrt{2\epsilon_{\text{pb}}(P,Q_h,m,\delta)} + 2\epsilon_{\text{pb}}(P,Q_h,m,\delta)$$
$$< \epsilon_{\text{pb-det-B}}(m,\delta) \tag{25}$$

and for all $\mathbf{x}$:

$$\mathcal{L}^{(0)}(h(\mathbf{x}),y) \leq \mathbb{E}_{\tilde{h}\sim Q_h}[\mathcal{L}'^{(\gamma/2)}(\tilde{h}(\mathbf{x}),y)]$$
$$\leq \mathcal{L}'^{(\gamma)}(h(\mathbf{x}),y) \tag{26}$$

as a result of which the following one-sided uniform convergence also holds:

$$\sup_{S\in\mathcal{S}_\delta}\sup_{h\in\mathcal{H}_\delta}\mathcal{L}^{(0)}_{\mathcal{D}}(h)-\hat{\mathcal{L}}'^{(\gamma)}_{S}(h)<\epsilon_{\text{pb-det-B}}(m,\delta)$$

We can similarly show that $\epsilon_{\text{pb-det-B}}(m,\delta)$ is lower-bounded by the uniform convergence bound of $\epsilon_{\text{unif-alg}}$ too.

Theorem H.2. *Let $\mathcal{A}$ be an algorithm such that on at least $1-\delta$ draws of the training dataset S, the algorithm outputs a hypothesis h_S such that the margin-based training loss can be bounded as:*

$$\hat{\mathcal{L}}'^{(\gamma)}_{S}(h_S)\leq\hat{\epsilon}(m,\delta)$$

and with high probability $1-\delta$ over the draws of S, the generalization error can be bounded as:

$$\mathcal{L}'^{(\gamma)}_{\mathcal{D}}(h_S)-\mathcal{L}'^{(\gamma)}_{S}(h_S)\leq\epsilon_{gen}(m,\delta)$$

Then there exists a set of samples $\mathcal{S}_{3\delta}$ of mass at least $1-3\delta$, and a corresponding set of hypothesis $\mathcal{H}_{3\delta}$ learned on these sample sets such that:

$$\left(\sup_{h\in\mathcal{H}_{3\delta}}\sup_{S\in\mathcal{S}_{3\delta}}\mathcal{L}^{(0)}_{S}(h)-\mathcal{L}'^{(\gamma)}_{\mathcal{D}}(h)\right)-(e^{3/2}-1)(\hat{\epsilon}(m,\delta)+\epsilon_{gen}(m,\delta))\leq\epsilon_{pb\text{-}det\text{-}B}(m,\delta)$$

Note that the above statement is slightly different from how Theorem H.1 is stated as it is not expressed in terms of $\epsilon_{\text{unif-alg}}$. In the corollary that follows the proof of this statement, we will see how it can be reduced in terms of $\epsilon_{\text{unif-alg}}$.

Proof. Most of the proof is similar to the proof of Theorem H.1. Like in the proof of Theorem H.1, we can argue that there exists $\mathcal{S}_{3\delta}$ and $\mathcal{H}_{3\delta}$ for which the test error can be bounded as:

$$\mathbb{E}_{\tilde{h}\sim Q_h}[\mathcal{L}'^{(\gamma/2)}_{\mathcal{D}}(\tilde{h})]\leq\mathcal{L}'^{(\gamma)}_{\mathcal{D}}(h)\leq\hat{\epsilon}(m,\delta)+\epsilon_{\text{gen}}(m,\delta)$$

. where we have used $\epsilon_{\text{gen}}(m,\delta)$ to denote the generalization error of $\mathcal{L}'^{(\gamma)}$ and not the 0-1 error (we note that this is ambiguous notation, but we keep it this way for simplicity).

For convenience, let us denote by $a:=\mathbb{E}_{\tilde{h}\sim Q_h}[\hat{\mathcal{L}}'^{(\gamma/2)}_{S}(\tilde{h})]$ and $b:=\mathbb{E}_{\tilde{h}\sim Q_h}[\mathcal{L}'^{(\gamma/2)}_{\mathcal{D}}(\tilde{h})]$. Again, let us consider, for some $h\in\mathcal{H}_{3\delta}$ and $S\in\mathcal{S}_{3\delta}$, the case that $e^{3/2}b\geq a$. Then, we have, using the above equation:

$$\begin{aligned}\hat{\mathcal{L}}^{(0)}_{S}(h)-\hat{\mathcal{L}}^{(\gamma)}_{\mathcal{D}}(h)\leq&a-b\\\leq&(e^{3/2}-1)b\\\leq&(e^{3/2}-1)(\hat{\epsilon}(m,\delta)+\epsilon_{\text{gen}}(m,\delta))\\\leq&(e^{3/2}-1)(\hat{\epsilon}(m,\delta)+\epsilon_{\text{gen}}(m,\delta)\\&+\epsilon_{\text{pb-det-B}}(m,\delta))\end{aligned}\tag{27}$$

Now consider the case where $a>e^{3/2}b$. Again, by similar arithmetic manipulation in the PAC-Bayesian bound of Equation 18 applied on $\mathcal{L}'^{(\gamma/2)}$, we get:

$$\begin{aligned}\epsilon_{\text{pb}}(P,Q_h,m,\delta)\geq&\underbrace{a\ln\frac{a}{b}}_{\geq 3/2}-\frac{(a-b)}{2}\left(\frac{1}{(1-b)}+1\right)\\\geq&\frac{a-b}{2}\\\geq&\frac{1}{2}\left(\mathcal{L}^{(0)}_{S}(h)-\mathcal{L}'^{(\gamma)}_{\mathcal{D}}(h)\right)\end{aligned}$$

Rearranging, we get:

$$\begin{aligned}\mathcal{L}^{(0)}_{S}(h)-\mathcal{L}'^{(\gamma)}_{\mathcal{D}}(h)\leq&\underbrace{2\epsilon_{\text{pb}}(P,Q_h,m,\delta)}_{\text{Applying Equation 25}}\\\leq&\epsilon_{\text{pb-det-B}}(m,\delta)\end{aligned}\tag{28}$$

Since, for all $h\in\mathcal{H}_{3\delta}$ and $S\in\mathcal{S}_{3\delta}$, one of Equations 27 and 28 hold, we have the claimed result.

□

Similarly, as a result of the above theorem, we can show that $\epsilon_{\text{pb-det-B}}(m,\delta)=\Omega(1)-\mathcal{O}(\epsilon)$, thus establishing that, for sufficiently large D, even though the generalization error would be negligibly small, the PAC-Bayes based bound would be as large as a constant and hence cannot explain generalization.

Corollary H.2.1. *In the setup of Section 5, for any $\epsilon,\delta>0,\delta<1/12$, when $D=\Omega\left(\max\left(m\ln\frac{3}{\delta},m\ln\frac{1}{\epsilon}\right)\right)$, we have:*

$$1-(e^{3/2}-1)\epsilon\leq\epsilon_{pb\text{-}det\text{-}B}(m,\delta)$$

Proof. It follows from the proof of Theorem 5.1 that $\hat{\epsilon}(m,\delta)=0$, since all training points are classified by a margin of γ (see Equation 11). Similarly, from Equation 14 in that proof, since most test points are classified by a margin of γ, $\epsilon_{\text{gen}}(m,\delta)\leq\epsilon$. Now, as long as $3\delta<1/4$, and D is sufficiently large (i.e., in the lower bounds on D in Theorem 5.1, if we replace δ by 3δ), we will get that there exists $S\in\mathcal{S}_{3\delta}$ and $h\in\mathcal{H}_{3\delta}$ for which the empirical loss

$\mathcal{L}^{(0)}$ loss is 1. Then, by Theorem H.2, we get the result in the above corollary. □