arXiv:1710.05468v3 [stat.ML] 22 Feb 2018

Generalization in Deep Learning

Kenji Kawaguchi Leslie Pack Kaelbling
Massachusetts Institute of Technology

Yoshua Bengio
University of Montreal, CIFAR Fellow

Abstract

With a direct analysis of neural networks, this paper presents a mathematically tight generalization theory to partially address an open problem regarding the generalization of deep learning. Unlike previous bound-based theory, our main theory is quantitatively as tight as possible for every dataset individually, while producing qualitative insights competitively. Our results give insight into why and how deep learning can generalize well, despite its large capacity, complexity, possible algorithmic instability, nonrobustness, and sharp minima, answering to an open question in the literature. We also discuss limitations of our results and propose additional open problems.

1. Introduction

Deep learning has seen significant practical success and has had a profound impact on the conceptual bases of machine learning and artificial intelligence. Along with its practical success, the theoretical properties of deep learning have been a subject of active investigation. For *expressivity* of neural networks, there are classical results regarding their universality (Leshno et al., 1993) and exponential advantages over hand-crafted features (Barron, 1993). Another series of theoretical studies have considered how *trainable* (or optimizable) deep hypothesis spaces are, revealing structural properties that may enable non-convex optimization (Choromanska et al., 2015; Kawaguchi, 2016a). However, merely having an *expressive* and *trainable* hypothesis space does not guarantee good performance in predicting the values of future inputs, because of possible over-fitting to training data. This leads to the study of *generalization*, which is the focus of this paper.

Some classical theory work attributes generalization ability to the use of a low-capacity class of hypotheses (Vapnik, 1998; Mohri et al., 2012). From the viewpoint of compact representation, which is related to small capacity, it has been shown that deep hypothesis spaces have an exponential advantage over shallow hypothesis spaces for representing some classes of natural target functions (Pascanu et al., 2014; Montufar et al., 2014; Livni et al., 2014; Telgarsky, 2016; Poggio et al., 2017). In other words, when some assumptions implicit in the hypothesis space (e.g., deep composition of piecewise linear transformations) are approximately satisfied by the target function, one can achieve very good generalization, compared to methods that do not rely on that assumption. However, a recent paper (Zhang et al., 2017) has empirically shown that successful deep hypothesis spaces have sufficient capacity to memorize random labels. This observation has been called an "apparent paradox" and has led to active discussion by many researchers (Arpit et al., 2017; Krueger et al., 2017; Hoffer et al., 2017; Wu et al., 2017; Dziugaite and Roy, 2017; Dinh et al., 2017). Zhang et al. (2017) concluded with an open problem stating that understanding such observations

require rethinking generalization, while Dinh et al. (2017) stated that explaining why deep learning models can generalize well, despite their overwhelming capacity, is an open area of research.

We begin, in Section 3, by illustrating that, even in the case of linear models, hypothesis spaces with overwhelming capacity can result in arbitrarily small test errors and expected risks. We closely examine this phenomenon, extending the original open problem from previous papers (Zhang et al., 2017; Dinh et al., 2017) into a new open problem that strictly includes the original. We reconcile the possible apparent paradox by pointing out a difference in the underlying assumptions.

Our primary objective, however, is to arrive at a quantitatively tight generalization theory with qualitative insights and make progress toward solving these open problems. In Section 4, we focus on a specific case of feed-forward neural networks with ReLU units, max-pooling, and squared loss. Under these conditions, we develop a constructive theory that provides new quantitatively tight theoretical insights into the generalization behavior of neural networks. In Section 5, we address a more general class of learning problems, relaxing the restriction to neural networks with squared loss, and provide another style of generalization bounds, which depend on validation error.

2. Preliminaries

Let $x \in \mathcal{X}$ be an input and $y \in \mathcal{Y}$ be a target. Let ℓ be a loss function. Let $R[f]$ be the expected risk of a function f, $R[f] = \mathbb{E}_{x,y\sim P}[\ell(f(x), y)]$, where P is the true distribution. Let $f_{\mathcal{A}(S_m)} : \mathcal{X} \to \mathcal{Y}$ be a model learned by a learning algorithm $\mathcal{A}$ (including random seeds for simplicity) using a training dataset $S_m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of size m. Let $\hat{R}_{S_m}[f]$ be the empirical risk of f, $\hat{R}_{S_m}[f] = \frac{1}{m}\sum_{i=1}^{m} \ell(f(x_i), y_i)$ with $\{(x_i, y_i)\}_{i=1}^{m} = S_m$. Let F be a set of functions or a *hypothesis space*. Let $\mathcal{L}_F$ be a family of loss functions associated with F, defined by $\mathcal{L}_F = \{g : f \in F, g(x, y) \triangleq \ell(f(x), y)\}$. All vectors are *column* vectors in this paper. For any given variable v, let d_v be the dimensionality of the variable v.

A goal in machine learning is typically framed as the minimization of the expected risk $R[f_{\mathcal{A}(S_m)}]$. We typically aim to minimize the non-computable expected risk $R[f_{\mathcal{A}(S_m)}]$ by minimizing the computable empirical risk $\hat{R}_{S_m}[f_{\mathcal{A}(S_m)}]$ (i.e., empirical risk minimization). One goal of the generalization theory is to explain and justify when and how minimizing $\hat{R}_{S_m}[f_{\mathcal{A}(S_m)}]$ is a sensible approach to minimizing $R[f_{\mathcal{A}(S_m)}]$ by analyzing

$$\text{the generalization gap} \triangleq R[f_{\mathcal{A}(S_m)}] - \hat{R}_{S_m}[f_{\mathcal{A}(S_m)}].$$

In this section only, we use the typical assumption that S_m is generated by i.i.d. draws according to the true distribution P; in general, this paper does not utilize this assumption. Under this assumption, a primary challenge of analyzing the generalization gap stems from the *dependence* of $f_{\mathcal{A}(S_m)}$ on the same dataset S_m used in the definition of $\hat{R}_{S_m}$. Several approaches in *statistical learning theory* have been developed to handle this dependence.

The *hypothesis-space complexity* approach handles this dependence by decoupling $f_{\mathcal{A}(S_m)}$ from the particular S_m by considering the worst-case gap for functions in the hypothesis space as

$$R[f_{\mathcal{A}(S_m)}] - \hat{R}_{S_m}[f_{\mathcal{A}(S_m)}] \leq \sup_{f\in F} R[f] - \hat{R}_{S_m}[f],$$

and by carefully analyzing the right-hand side. Because the cardinality of F is typically (uncountably) infinite, a direct use of the union bound over all elements in F yields a vacuous bound, leading to the need to consider different quantities to characterize F; e.g., Rademacher complexity and the Vapnik–Chervonenkis (VC) dimension. For example, if the codomain of ℓ is in $[0, 1]$, we have (Mohri et al., 2012, Theorem 3.1) that for any $\delta > 0$, with probability at least $1 - \delta$,

$$\sup_{f \in F} R[f] - \hat{R}_{S_m}[f] \leq 2\mathfrak{R}_m(\mathcal{L}_F) + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}},$$

where $\mathfrak{R}_m(\mathcal{L}_F)$ is the Rademacher complexity of $\mathcal{L}_F$, which then can be bounded by the Rademacher complexity of F, $\mathfrak{R}_m(F)$. For the deep-learning hypothesis spaces F, there are several well-known bounds on $\mathfrak{R}_m(F)$ including those with explicit exponential dependence on depth (Sun et al., 2016; Neyshabur et al., 2015b; Xie et al., 2015) and explicit linear dependence on the number of trainable parameters (Shalev-Shwartz and Ben-David, 2014). There has been significant work on improving the bounds in this approach, but all existing solutions with this approach still depend on the complexity of a hypothesis space or a sequence of hypothesis spaces.

The *stability* approach deals with the dependence of $f_{\mathcal{A}(S_m)}$ on the dataset S_m by considering the *stability* of algorithm $\mathcal{A}$ with respect to different datasets. The considered stability is a measure of how much changing a data point in S_m can change $f_{\mathcal{A}(S_m)}$. For example, if the algorithm $\mathcal{A}$ has uniform stability β (w.r.t. ℓ) and if the codomain of ℓ is in $[0, M]$, we have (Bousquet and Elisseeff, 2002) that for any $\delta > 0$, with probability at least $1 - \delta$,

$$R[f_{\mathcal{A}(S_m)}] - \hat{R}_{S_m}[f_{\mathcal{A}(S_m)}] \leq 2\beta + (4m\beta + M)\sqrt{\frac{\ln \frac{1}{\delta}}{2m}}.$$

Based on previous work on stability (e.g., Hardt et al. 2015; Kuzborskij and Lampert 2017; Gonen and Shalev-Shwartz 2017), one may conjecture some reason for generalization in deep learning, the proving of which requires future work.

The *robustness* approach avoids dealing with certain details of the dependence of $f_{\mathcal{A}(S_m)}$ on S_m by considering the robustness of algorithm $\mathcal{A}$ for all possible datasets. In contrast to stability, robustness is the measure of how much the loss value can vary w.r.t. *the input space* of (x, y). For example, if algorithm $\mathcal{A}$ is $(\Omega, \zeta(\cdot))$-robust and the codomain of ℓ is upper-bounded by M, given a dataset S_m, we have (Xu and Mannor, 2012) that for any $\delta > 0$, with probability at least $1 - \delta$,

$$|R[f_{\mathcal{A}(S_m)}] - \hat{R}_{S_m}[f_{\mathcal{A}(S_m)}]| \leq \zeta(S_m) + M\sqrt{\frac{2\Omega \ln 2 + 2 \ln \frac{1}{\delta}}{m}}.$$

The robustness approach requires an *a priori known and fixed* partition of the input space such that the number of sets in the partition is Ω and the change of loss values in each set of the partition is bounded by $\zeta(S_m)$ *for all* S_m (Definition 2 and the proof of Theorem 1 in Xu and Mannor 2012). In classification, if the *margin* is ensured to be large, we can fix the partition with balls of radius corresponding to the large *margin*, filling the input space. Recently, this idea was applied to deep learning (Sokolic et al., 2017a,b), producing

insightful and effective generalization bounds, while still suffering from the curse of the dimensionality of a priori-known fixed input manifold.

With regard to the above approaches, *flat minima* can be viewed as the concept of low variation in the *parameter space*; i.e., a small perturbation in the parameter space around a solution results in a small change in the loss surface. Several studies have provided arguments for generalization in deep learning based on flat minima (Keskar et al., 2017). However, Dinh et al. (2017) showed that flat minima in practical deep learning hypothesis spaces can be turned into sharp minima via re-parameterization without affecting the generalization gap, indicating that it requires further investigation.

3. Rethinking Generalization

Zhang et al. (2017) empirically demonstrated that several deep hypothesis spaces can memorize random labels, while having the ability to produce zero training error and small test errors for particular natural datasets (e.g., CIFAR-10). They also empirically observed that regularization on the norm of weights seemed to be unnecessary to obtain small test errors, in contradiction to conventional wisdom. These observations suggest the following problem to be solved:

Problem 1. Tightly characterize the expected risk $R[f]$ or the generalization gap $R[f] - \hat{R}_{S_m}[f]$ with a sufficiently complex deep-learning hypothesis space $F \ni f$, producing theoretical insights and distinguishing the case of "natural" problem instances (P, S_m) (e.g., images with natural labels) from the case of other problem instances (P', S'_m) (e.g., images with random-labels).

Supporting and extending the empirical observations by Zhang et al. (2017), we provide a theorem (Theorem 1) stating that the hypothesis space of over-parameterized linear models can memorize any training data *and* decrease the training and test errors arbitrarily close to zero (including zero) *with* the norm of parameters being arbitrarily large, *even when* the parameters are arbitrarily far from the ground-truth parameters. Furthermore, Corollary 2 shows that conventional wisdom regarding the norm of the parameters w can fail to explain generalization, even in linear models that might be seemingly not over-parameterized. All proofs for this paper are presented in the appendix.

Theorem 1. *Consider a linear model with the training prediction $\hat{Y}(w) = \Phi w \in \mathbb{R}^{m \times d_y}$, where $\Phi \in \mathbb{R}^{m \times n}$ is a fixed feature matrix of the training inputs. Let $\hat{Y}_{\text{test}}(w) = \Phi_{\text{test}} w \in \mathbb{R}^{m_{\text{test}} \times d_y}$ be the test prediction, where $\Phi_{\text{test}} \in \mathbb{R}^{m_{\text{test}} \times n}$ is a fixed feature matrix of the test inputs. Let $M = [\Phi^\top, \Phi_{\text{test}}^\top]^\top$. Then, if $n > m$ and if $\text{rank}(\Phi) \geq m$ and $\text{rank}(M) < n$,*

(i) For any $Y \in \mathbb{R}^{m \times d_y}$, there exists a parameter w' such that $\hat{Y}(w') = Y$, and

(ii) if there exists a ground truth w^* satisfying $Y = \Phi w^*$ and $Y_{\text{test}} = \Phi_{\text{test}} w^*$, then for any $\epsilon, \delta \geq 0$, there exists a parameter w such that

(a) $\hat{Y}(w) = Y + \epsilon A$ for some matrix A with $\|A\|_F \leq 1$, and
(b) $\hat{Y}_{\text{test}}(w) = Y_{\text{test}} + \epsilon B$ for some matrix B with $\|B\|_F \leq 1$, and
(c) $\|w\|_F \geq \delta$ and $\|w - w^*\|_F \geq \delta$.

Corollary 2. *If $n \leq m$ and if* $\text{rank}(M) < n$*, then statement (ii) in Theorem 1 holds.*

Whereas Theorem 1 and Corollary 2 concern test errors instead of expected risk (in order to be consistent with empirical studies), Proposition 8 in Appendix A.1 proves the same phenomena for expected risk for general machine learning models not limited to deep learning and linear hypothesis spaces; i.e., Proposition 8 in Appendix A.1 states that *none of small capacity, low complexity, stability, robustness, and flat minima is necessary for generalization in general machine learning for each given problem instance* (P, S_m). We capture the essence of all of these observations in the following remark.

Remark 3. The expected risk $R[f]$ and the generalization gap $R[f] - \hat{R}_{S_m}[f]$ of a hypothesis f with a true distribution P and a dataset S_m are completely determined by *the* tuple (P, S_m, f), independently of other factors, such as a hypothesis space F (and hence its properties such as capacity, Rademacher complexity, pre-defined bound on norms, and flat-minima) and properties of random datasets different from the given S_m (e.g., stability and robustness of the learning algorithm $\mathcal{A}$). In contrast, the conventional wisdom states that these other factors are what matter. This has created the "apparent paradox" in the literature.

From these observations, we propose the following open problem:

Problem 2. Tightly characterize the expected risk $R[f]$ or the generalization gap $R[f] - \hat{R}_{S_m}[f]$ of a hypothesis f with a pair (P, S_m) of a true distribution and a dataset, producing theoretical insights, based only on properties of *the* hypothesis f and *the* pair (P, S_m).

Solving Problem 2 for deep learning implies solving Problem 1, but not vice versa. Problem 2 encapsulates the essence of Problem 1 and all the issues from our Theorem 1, Corollary 2 and Proposition 8.

3.1 Conceptually Resolving the Generalization Puzzle

The empirical observations in (Zhang et al., 2017) and our results above may seem to contradict the results of the generalization theory. However, there is no contradiction, and the apparent inconsistency arises from differences in assumptions. Indeed, under certain assumptions, many results in statistical learning theory have been shown to be tight and insightful (e.g., Mukherjee et al. 2006; Mohri et al. 2012).

Figure 1 illustrates the differences in assumptions in statistical learning theory and some empirical studies. On one hand, in statistical learning theory, a distribution P and a dataset S_m are usually unspecified except that P is in some set $\mathcal{P}$ and a dataset $S_m \in D$ is drawn randomly according to P (typically with the i.i.d. assumption). On the other hand, in most empirical studies and in our theoretical results (Theorem 1 and Proposition 8), a distribution P is still unknown yet specified (e.g., via a real world process) and a dataset S_m is specified and usually known (e.g., CIFAR-10 or ImageNet). Intuitively, whereas statistical learning theory considers a set $\mathcal{P} \times D$ because of weak assumptions, some empirical studies can focus on a specified point (P, S_m) in a set $\mathcal{P} \times D$ because of stronger assumptions. Therefore, by using the same terminology such as "expected risk" and "generalization" in both cases, we are susceptible to confusion and apparent contradiction.

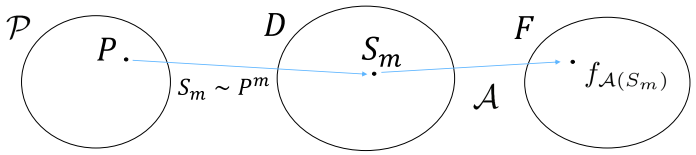

Figure 1: An illustration of differences in assumptions. Statistical learning theory analyzes the generalization behaviors of $f_{\mathcal{A}(S_m)}$ over randomly-drawn *unspecified* datasets $S_m \in D$ according to some *unspecified* distribution $P \in \mathcal{P}$. Intuitively, statistical learning theory concerns more about questions regarding a set $\mathcal{P} \times D$ because of the *unspecified* nature of (P, S_m), whereas certain empirical studies (e.g., Zhang et al. 2017) can focus on questions regarding each *specified* point $(P, S_m) \in \mathcal{P} \times D$.

Lower bounds, necessary conditions and tightness in statistical learning theory are typically defined via a worst-case distribution $P_{\text{worst}} \in \mathcal{P}$. For instance, classical "no free lunch" theorems and certain lower bounds on the generalization gap (e.g., Mohri et al. 2012, Section 3.4) have been proven *for* the worst-case distribution $P_{\text{worst}} \in \mathcal{P}$. Therefore, "tight" and "necessary" typically mean "tight" and "necessary" for the set $\mathcal{P} \times D$ (e.g., through the worst or average case), but *not* for each particular point $(P, S_m) \in \mathcal{P} \times D$. From this viewpoint, we can understand that even if the quality of the set $\mathcal{P} \times D$ is "bad" overall, there may exist a "good" point $(P, S_m) \in \mathcal{P} \times D$.

Several approaches in statistical learning theory, such as the data-dependent and Bayesian approaches (Herbrich and Williamson, 2002; Dziugaite and Roy, 2017), use more assumptions on the set $\mathcal{P} \times D$ to take advantage of more prior and posterior information; these have an ability to tackle Problem 1. However, these approaches do not apply to Problem 2 as they still depend on other factors than the given (P, S_m, f). For example, data-dependent bounds with the *luckiness framework* (Shawe-Taylor et al., 1998; Herbrich and Williamson, 2002) and *empirical* Rademacher complexity (Koltchinskii and Panchenko, 2000; Bartlett et al., 2002) still depend on a concept of hypothesis spaces (or the sequence of hypothesis spaces), and the robustness approach (Xu and Mannor, 2012) depend on different datasets than a given S_m via the definition of robustness (i.e., in Section 2, $\zeta(S_m)$ is a data-dependent term, but the definition of ζ itself and Ω depend on other datasets than S_m).

We note that analyzing a set $\mathcal{P} \times D$ is of significant interest for its own merits and is a natural task along the field of computational complexity (e.g., categorizing a set of problem instances into subsets with or without polynomial solvability). Indeed, the situation where theory focuses more on a set and many practical studies focus on each element in the set is prevalent in computer science (see the discussion in Appendix B.1 for more detail). We further validate the logical consistency in our observations in Appendix B.2 and propose several practical roles of generalization theory in Appendix B.3.

4. Direct Analyses of Neural Networks

In the previous section, we extended Problem 1 to Problem 2, and identified the different assumptions in theoretical and empirical studies. Accordingly, this section aims to solve these problems, both in the case of each specified dataset and the case of random unspecified datasets. To achieve this goal, this section presents a *direct analysis* of neural networks,

rather than deriving results about neural networks from more generic theories based on capacity, Rademacher complexity, stability, or robustness. This section focuses on the generalization gap $R[f_{\mathcal{A}(S_m)}] - \hat{R}_{S_m}[f_{\mathcal{A}(S_m)}]$ with a training dataset S_m and with squared loss. For 0-1 loss with multi-labels, our probabilistic bound is presented in Appendix A.2.

4.1 Model Description via Deep Paths

We consider general neural networks of any depth that have the structure of a directed acyclic graph (DAG) with ReLU nonlinearity and/or max pooling. This includes any structure of a feedforward network with convolutional and/or fully connected layers, potentially with skip connections. For pedagogical purposes, we first discuss our model description for layered networks without skip connections, and then describe it for DAGs.

Layered nets without skip connections Let $h^{(l)}(x,w) \in \mathbb{R}^{n_l}$ be the pre-activation vector of the l-th hidden layer, where n_l is the width of the l-th hidden layer, and w represents the trainable parameters. Let H be the number of hidden layers. For layered networks without skip connections, the pre-activation (or pre-nonlinearity) vector of the l-th layer can be written as

$$h^{(l)}(x,w) = W^{(l)}\sigma^{(l-1)}\left(h^{(l-1)}(x,w)\right),$$

with a boundary definition $\sigma^{(0)}\left(h^{(0)}(x,w)\right) \equiv x$, where $\sigma^{(l-1)}$ represents nonlinearity via ReLU and/or max pooling at the $(l-1)$-th hidden layer, and $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is a matrix of weight parameters connecting the $(l-1)$-th layer to the l-th layer. Here, $W^{(l)}$ can have *any* structure (e.g., shared and sparse weights to represent a convolutional layer). Let $\dot{\sigma}^{(l)}(x,w)$ be a vector with each element being 0 or 1 such that $\sigma^{(l)}\left(h^{(l)}(x,w)\right) = \dot{\sigma}^{(l)}(x,w) \circ h^{(l)}(x,w)$, which is an element-wise product of the vectors $\dot{\sigma}^{(l)}(x,w)$ and $h^{(l)}(x,w)$. Then, we can write the pre-activation of the k-th output unit at the last layer $l = H+1$ as

$$h_k^{(H+1)}(x,w) = \sum_{j_H=1}^{n_H} W_{kj_H}^{(H+1)} \dot{\sigma}_{j_H}^{(H)}(x,w) h_{j_H}^{(H)}(x,w).$$

By expanding $h^{(l)}(x,w)$ repeatedly and exchanging the sum and product via the distributive law of multiplication,

$$h_k^{(H+1)}(x,w) = \sum_{j_H=1}^{n_H} \sum_{j_{H-1}=1}^{n_{H-1}} \cdots \sum_{j_0=1}^{n_0} \overline{W}_{kj_Hj_{H-1}\cdots j_0} \dot{\sigma}_{j_Hj_{H-1}\cdots j_1}(x,w) x_{j_0},$$

where $\overline{W}_{kj_Hj_{H-1}\cdots j_0} = W_{kj_H}^{(H+1)} \prod_{l=1}^{H} W_{j_l j_{l-1}}^{(l)}$ and $\dot{\sigma}_{j_Hj_{H-1}\cdots j_1}(x,w) = \prod_{l=1}^{H} \dot{\sigma}_{j_l}^{(l)}(x,w)$. By merging the indices $j_0, \ldots, j_H$ into j with some bijection between $\{1,\ldots,n_0\} \times \cdots \times \{1,\ldots,n_H\} \ni (j_0,\ldots,j_H)$ and $\{1,\ldots,n_0 n_1 \cdots n_H\} \ni j$,

$$h_k^{(H+1)}(x,w) = \textstyle\sum_j \bar{w}_{k,j} \bar{\sigma}_j(x,w) \bar{x}_j,$$

where $\bar{w}_{k,j}, \bar{\sigma}_j(x,w)$ and $\bar{x}_j$ respectively represent $\overline{W}_{kj_Hj_{H-1}\cdots j_0}, \dot{\sigma}_{j_Hj_{H-1}\cdots j_1}(x,w)$ and x_{j_0} with the change of indices (i.e., $\sigma_j(x,w)$ and $\bar{x}_j$ respectively contain the n_0 numbers and $n_1 \cdots n_H$ numbers of the same copy of each $\dot{\sigma}_{j_Hj_{H-1}\cdots j_1}(x,w)$ and x_{j_0}). Note that $\sum_j$ represents summation over all the paths from the input x to the k-th output unit.

DAGs Remember that every DAG has at least one topological ordering, which can be used to to create a layered structure with possible skip connections (e.g., see Healy and Nikolov 2001; Neyshabur et al. 2015b). In other words, we consider DAGs such that the pre-activation vector of the l-th layer can be written as

$$h^{(l)}(x,w) = \sum_{l'=0}^{l-1} W^{(l,l')} \sigma^{(l')} \left(h^{(l')}(x,w) \right)$$

with a boundary definition $\sigma^{(0)} \left(h^{(0)}(x,w) \right) \equiv x$, where $W^{(l,l')} \in \mathbb{R}^{n_l \times n_{l'}}$ is a matrix of weight parameters connecting the l'-th layer to the l-th layer. Again, $W^{(l,l')}$ can have *any* structure. Thus, in the same way as with layered networks without skip connections, for all $k \in \{1, \dots, d_y\}$,

$$h_k^{(H+1)}(x,w) = \textstyle\sum_j \bar{w}_{k,j} \bar{\sigma}_j(x,w) \bar{x}_j,$$

where $\sum_j$ represents the summation over all paths from the input x to the k-th output unit; i.e., $\bar{w}_{k,j} \bar{\sigma}_j(x,w) \bar{x}_j$ is the contribution from the j-th path to the k-th output unit. Each of $\bar{w}_{k,j}, \bar{\sigma}_j(x,w)$ and $\bar{x}_j$ is defined in the same manner as in the case of layered networks without skip connections. In other words, the j-th path weight $\bar{w}_{k,j}$ is the product of the weight parameters in the j-th path, and $\bar{\sigma}_j(x,w)$ is the product of the 0-1 activations in the j-th path, corresponding to ReLU nonlinearity and max pooling; $\bar{\sigma}_j(x,w) = 1$ if all units in the j-th path are active, and $\bar{\sigma}_j(x,w) = 0$ otherwise. Also, $\bar{x}_j$ is the input used in the j-th path. Therefore, for DAGs, including layered networks without skip connections,

$$h_k^{(H+1)}(x,w) = [\bar{x} \circ \bar{\sigma}(x,w)]^\top \bar{w}_k, \tag{1}$$

where $[\bar{x} \circ \bar{\sigma}(x,w)]_j = \bar{x}_j \bar{\sigma}_j(x,w)$ and $(\bar{w}_k)_j = \bar{w}_{k,j}$ are the vectors of the size of the number of the paths.

4.2 Tight Theory for Every Pair (P, S_m)

Theorem 4 solves Problem 2 (and hence Problem 1) for neural networks with squared loss by stating that the generalization gap of a w with respect to a problem (P, S_m) is tightly analyzable with theoretical insights, based only on the quality of the w and the pair (P, S_m). We do *not* assume that S_m is generated randomly based on some relationship with P; the theorem holds for any dataset, regardless of how it was generated. Let w^{S_m} and $\bar{w}_k^{S_m}$ be the parameter vectors w and $\bar{w}_k$ learned with a dataset S_m. Let $R[w^{S_m}]$ and $\hat{R}_{S_m}[w^{S_m}]$ be the expect risk and empirical risk of the model with the learned parameter w^{S_m}. Let $z_i = [\bar{x}_i \circ \bar{\sigma}(x_i, w^{S_m})]$. Let $G = \mathbb{E}_{x,y\sim P}[zz^\top] - \frac{1}{m}\sum_{i=1}^m z_i z_i^\top$ and $v = \frac{1}{m}\sum_{i=1}^m y_{ik} z_i - \mathbb{E}_{x,y\sim P}[y_k z]$. Given a matrix M, let $\lambda_{\max}(M)$ be the largest eigenvalue of M.

Theorem 4. *Let $\{\lambda_j\}_j$ and $\{u_j\}_j$ be a set of eigenvalues and a corresponding orthonormal set of eigenvectors of G. Let $\theta^{(1)}_{\bar{w}_k,j}$ be the angle between u_j and $\bar{w}_k$. Let $\theta^{(2)}_{\bar{w}_k}$ be the angle*

between v and $\bar{w}_k$. Then (deterministically),

$$R[w^{\mathcal{S}_m}] - \hat{R}_{S_m}[w^{\mathcal{S}_m}] - c_y = \sum_{k=1}^{d_y} \left(2\|v\|_2 \|\bar{w}_k^{\mathcal{S}_m}\|_2 \cos\theta^{(2)}_{\bar{w}_k^{\mathcal{S}_m}} + \|\bar{w}_k^{\mathcal{S}_m}\|_2^2 \sum_j \lambda_j \cos^2\theta^{(1)}_{\bar{w}_k^{\mathcal{S}_m},j} \right)$$
$$\leq \sum_{k=1}^{d_y} \left(2\|v\|_2 \|\bar{w}_k^{\mathcal{S}_m}\|_2 + \lambda_{\max}(G) \|\bar{w}_k^{\mathcal{S}_m}\|_2^2 \right),$$

where $c_y = \mathbb{E}_y[\|y\|_2^2] - \frac{1}{m}\sum_{i=1}^m \|y_i\|_2^2$.

Proof idea. From Equation (1) with squared loss, we can decompose the generalization gap into three terms:

$$R[w^{\mathcal{S}_m}] - \hat{R}_{S_m}[w^{\mathcal{S}_m}] = \sum_{k=1}^{d_y} \left[(\bar{w}_k^{\mathcal{S}_m})^\top \left(\mathbb{E}[zz^\top] - \frac{1}{m}\sum_{i=1}^m z_i z_i^\top \right) \bar{w}_k^{\mathcal{S}_m} \right] \tag{2}$$
$$+ 2\sum_{k=1}^{d_y} \left[\left(\frac{1}{m}\sum_{i=1}^m y_{ik} z_i^\top - \mathbb{E}[y_k z^\top] \right) \bar{w}_k^{\mathcal{S}_m} \right]$$
$$+ \mathbb{E}[y^\top y] - \frac{1}{m}\sum_{i=1}^m y_i^\top y_i.$$

By manipulating each term, we obtain the desired statement. See Appendix D.1 for a complete proof. □

In Theorem 4, there is no concept of a hypothesis space or pre-specified bound on a norm of weights. Instead, it indicates that if the norm of the weights $\|\bar{w}_k^{\mathcal{S}_m}\|_2$ at the end of learning process with the actual given S_m is small, then the generalization gap is small, even if the norm $\|\bar{w}_k^{\mathcal{S}_m}\|_2$ is unboundedly large during the learning process with S_m or at anytime with any dataset other than S_m.

Importantly, in Theorem 4, there are two other significant factors in addition to the norm of the weights $\|\bar{w}_k^{\mathcal{S}_m}\|_2$. First, the eigenvalues of G and v measure the concentration of the given dataset S_m with respect to the (unknown) P in the space of the learned representation $z_i = [\bar{x}_i \circ \bar{\sigma}(x_i, w^{\mathcal{S}_m})]$. Here, we can see the benefit of deep learning from the viewpoint of "deep-path" feature learning: even if a given S_m is not concentrated in the original space, optimizing w can result in concentrating it in the space of z. Similarly, c_y measures the concentration of $\|y\|_2^2$, but c_y is independent of w and unchanged after a pair (P, S_m) is given. Second, the $\cos\theta$ terms measure the similarity between $\bar{w}_k^{\mathcal{S}_m}$ and these concentration terms. Because the norm of the weights $\|\bar{w}_k^{\mathcal{S}_m}\|_2$ is multiplied by those other factors, the generalization gap can remain small, even if $\|\bar{w}_k^{\mathcal{S}_m}\|_2$ is large, as long as some of those other factors are small.

Based on a generic bound-based theory, Neyshabur et al. (2015a,b) proposed to control the norm of the *path* weights $\|\bar{w}_k\|_2$, which is consistent with our direct bound-less result (and which is as computationally tractable as a standard forward-backward pass[1]). Unlike

1. From the derivation of Equation (1), one can compute $\|\bar{w}_k^{\mathcal{S}_m}\|_2^2$ with a single forward pass using element-wise squared weights, an identity input, and no nonlinearity. One can also follow the previous paper (Neyshabur et al., 2015a) for its computation.

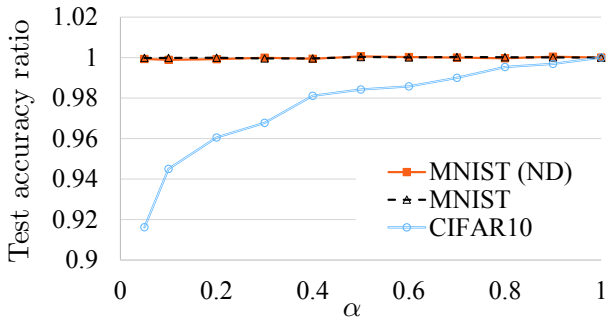

Figure 2: Test accuracy ratio (Two-phase/Base). Notice that the y-axis starts with high initial accuracy, even with a very small dataset size αm for learning w_σ.

the previous results, we do *not* require a pre-defined bound on $\|\bar{w}_k\|_2$ over different datasets, but depend only on its final value with each S_m as desired, in addition to more tight insights (besides the norm) via equality as discussed above. In addition to the pre-defined norm bound, these previous results have an explicit exponential dependence on the depth of the network, which does not appear in our Theorem 4. Similarly, some previous results specific to layered networks without skip connections (Sun et al., 2016; Xie et al., 2015) contain the 2^H factor *and* a bound on the product of the norm of weight matrices, $\prod_{\ell=1}^{H+1} \|W^{(\ell)}\|$, instead of $\sum_k \|\bar{w}_k^{S_m}\|_2$. Here, $\sum_k \|\bar{w}_k\|_2^2 \leq \prod_{\ell=1}^{H+1} \|W^{(\ell)}\|_F^2$ because the latter contains all of the same terms as the former as well as additional non-negative additive terms after expanding the sums in the definition of the norms.

Therefore, unlike previous bounds, Theorem 4 generates these new theoretical insights based on *the tight equality* (in the first line of the equation in Theorem 4).

4.3 Probabilistic Bound over Random Datasets

While the previous subsection tightly analyzed each given point (P, S_m), this subsection considers the set $\mathcal{P} \times D \ni (P, S_m)$, where D is the set of possible datasets S_m endowed with an i.i.d. product measure P^m where $P \in \mathcal{P}$ (see Section 3.1).

In Equation (2), the generalization gap is decomposed into three terms, each of which contains the difference between a sum of *dependent* random variables and its expectation. The dependence comes from the fact that $z_i = [\bar{x}_i \circ \bar{\sigma}(x_i, w^{S_m})]$ are dependent over the sample index i, because of the dependence of w^{S_m} on the entire dataset S_m. We then observe the following: in $h_k^{(H+1)}(x, w) = [\bar{x} \circ \bar{\sigma}(x, w)]^\top \bar{w}$, the derivative of $z = [\bar{x} \circ \bar{\sigma}(x, w)]$ with respect to w is zero everywhere (except for the measure zero set, where the derivative does not exist). Therefore, each step of the (stochastic) gradient decent greedily chooses the best direction in terms of $\bar{w}$ (with the current $z = [\bar{x} \circ \bar{\sigma}(x, w)]$), but not in terms of w in $z = [\bar{x} \circ \bar{\sigma}(x, w)]$ (see Appendix D.2 for more detail). This observation leads to a conjecture that the dependence of $z_i = [\bar{x}_i \circ \bar{\sigma}(x_i, w^{S_m})]$ via the training process with the whole dataset S_m is not entirely "bad"in terms of the concentration of the sum of the terms with z_i.

4.3.1 Empirical Observations

As a first step to investigate the dependence of z_i, we evaluated the following novel *two-phase* training procedure that explicitly breaks the dependence of z_i over the sample index i.

We first train a network in a standard way, but only using a *partial* training dataset $S_{\alpha m} = \{(x_1, y_1), \ldots, (x_{\alpha m}, y_{\alpha m})\}$ of size αm, where $\alpha \in (0, 1)$ (standard phase). We then assign the value of $w^{\mathcal{S}_{\alpha m}}$ to a new placeholder $w_\sigma := w^{\mathcal{S}_{\alpha m}}$ and freeze w_σ, meaning that as w changes, w_σ does not change. At this point, we have that $h_k^{(H+1)}(x, w^{\mathcal{S}_{\alpha m}}) = [\bar{x} \circ \bar{\sigma}(x, w_\sigma)]^\top \bar{w}_k^{\mathcal{S}_{\alpha m}}$. We then keep training only the $\bar{w}_k^{\mathcal{S}_{\alpha m}}$ part with the entire training dataset of size m (freeze phase), yielding the final model via this two-phase training procedure as

$$\tilde{h}_k^{(H+1)}(x, w^{\mathcal{S}_m}) = [\bar{x} \circ \bar{\sigma}(x, w_\sigma)]^\top \bar{w}_k^{\mathcal{S}_m}. \tag{3}$$

Note that the vectors $w_\sigma = w^{\mathcal{S}_{\alpha m}}$ and $\bar{w}_k^{\mathcal{S}_m}$ contain the untied parameters in $\tilde{h}_k^{(H+1)}(x, w^{\mathcal{S}_m})$. See Appendix D.3 for a simple implementation of this two-phase training procedure that requires at most (approximately) twice as much computational cost as the normal training procedure.

We implemented the two-phase training procedure with the MNIST and CIFAR-10 datasets. The test accuracies of the standard training procedure (base case) were 99.47% for MNIST (ND), 99.72% for MNIST, and 92.89% for CIFAR-10. MNIST (ND) indicates MNIST with no data augmentation. The experimental details are in Appendix D.4. Our source code is available at: http://lis.csail.mit.edu/code/gdl.html

Figure 2 presents the test accuracy ratios for varying α: the test accuracy of the two-phase training procedure divided by the test accuracy of the standard training procedure. The plot in Figure 2 begins with $\alpha = 0.05$, for which $\alpha m = 3000$ in MNIST and $\alpha m = 2500$ in CIFAR-10. Somewhat surprisingly, using a much smaller dataset for learning w_σ still resulted in competitive performance. A dataset from which we could more easily obtain a better generalization (i.e., MNIST) allowed us to use smaller αm to achieve competitive performance, which is consistent with our discussion above.

4.3.2 Theoretical Results

We now prove a probabilistic bound for the hypotheses resulting from the two-phase training algorithm. Let $\tilde{z}_i = [\bar{x}_i \circ \bar{\sigma}(x_i, w_\sigma)]$ where $w_\sigma := w^{\mathcal{S}_{\alpha m}}$, as defined in the two-phase training procedure above. Our two-phase training procedure forces $\tilde{z}_{\alpha m+1}, \ldots, \tilde{z}_m$ *over samples* to be independent random variables (each $\tilde{z}_i$ is dependent *over coordinates*, which is taken care of in our proof), while maintaining the competitive practical performance of the output model $\tilde{h}_k^{(H+1)}(\cdot, w^{\mathcal{S}_m})$. As a result, we obtain the following bound on the generalization gap for the practical deep models $\tilde{h}_k^{(H+1)}(\cdot, w^{\mathcal{S}_m})$. Let $m_\sigma = (1 - \alpha)m$. Given a matrix M, let $\|M\|_2$ be the spectral norm of M.

Assumption 1. *Let* $G^{(i)} = \mathbb{E}_x[\tilde{z}\tilde{z}^\top] - \tilde{z}_i\tilde{z}_i^\top$, $V_{kk'}^{(i)} = y_{ik}\tilde{z}_{i,k'} - \mathbb{E}_{x,y}[y_k\tilde{z}_{k'}]$, *and* $c_y^{(i)} = \mathbb{E}_y[\|y\|_2^2] - \|y_i\|_2^2$. *Assume that for all* $i \in \{\alpha m + 1, \ldots, m\}$,

- $C_{zz} \geq \lambda_{\max}(G^{(i)})$ *and* $\gamma_{zz}^2 \geq \|\mathbb{E}_x[(G^{(i)})^2]\|_2$
- $C_{yz} \geq \max_{k,k'} |V_{kk'}^{(i)}|$ *and* $\gamma_{yz}^2 \geq \max_{k,k'} \mathbb{E}_x[(V_{kk'}^{(i)})^2])$
- $C_y \geq |c_y^{(i)}|$ *and* $\gamma_y^2 \geq \mathbb{E}_x[(c_y^{(i)})^2]$.

Theorem 5. *Suppose that Assumption 1 holds. Assume that* $S_m \setminus S_{\alpha m}$ *is generated by i.i.d. draws according to true distribution* P. *Assume that* $S_m \setminus S_{\alpha m}$ *is independent of* $S_{\alpha m}$. *Let*

$f_{\mathcal{A}(S_m)}$ be the model learned by the two-phase training procedure with S_m. Then, for each $w_\sigma := w^{S_{\alpha m}}$, for any $\delta > 0$, with probability at least $1-\delta$,

$$R[f_{\mathcal{A}(S_m)}] - \hat{R}_{S_m \setminus S_{\alpha m}}[f_{\mathcal{A}(S_m)}] \leq \beta_1 \sum_{k=1}^{d_y} \left\|\bar{w}_k^{\mathcal{S}_m}\right\|_1 + 2\beta_2 \sum_{k=1}^{d_y} \left\|\bar{w}_k^{\mathcal{S}_m}\right\|_2^2 + \beta_3,$$

where $\beta_1 = \frac{2C_{zz}}{3m_\sigma} \ln \frac{3d_z}{\delta} + \sqrt{\frac{2\gamma_{zz}^2}{m_\sigma} \ln \frac{3d_z}{\delta}}$, $\beta_2 = \frac{2C_{yz}}{3m_\sigma} \ln \frac{6d_y d_z}{\delta} + \sqrt{\frac{\gamma_{yz}^2}{m_\sigma} \ln \frac{6d_y d_z}{\delta}}$, *and* $\beta_2 = \frac{2C_y}{3m_\sigma} \ln \frac{3}{\delta} + \sqrt{\frac{2\gamma_y^2}{m_\sigma} \ln \frac{3}{\delta}}$.

Our proof does *not* require independence over the coordinates of $\tilde{z}_i$ and the entries of the random matrices $\tilde{z}_i \tilde{z}_i^\top$ (see the proof of Theorem 5).

The bound in Theorem 5 is data-dependent because the norms of the weights $\bar{w}_k^{\mathcal{S}_m}$ depend on each particular S_m. Similarly to Theorem 4, the bound in Theorem 5 does not contain a pre-determined bound on the norms of weights and can be independent of the concept of hypothesis space, as desired; i.e., Assumption 1 can be also satisfied without referencing a hypothesis space of w, because $\tilde{z} = [\bar{x}_i \circ \bar{\sigma}(x_i, w_\sigma)]$ with $\bar{\sigma}_j(x_i, w_\sigma) \in \{0, 1\}$. However, unlike Theorem 4, Theorem 5 *implicitly* contains the properties of datasets different from a given S_m, via the pre-defined bounds in Assumption 1. This is expected since Theorem 5 makes claims about the set of random datasets S_m instead of each instantiated S_m. Therefore, while Theorem 5 presents a strongly-data-dependent bound (over random datasets), Theorem 4 is tighter for each given S_m; indeed, the main equality of Theorem 4 is as tight as possible.

Theorems 4 and 5 provide generalization bounds for practical deep learning models that do not necessarily have explicit dependence on the number of weights, or exponential dependence on depth or effective input dimensionality. Although the size of the vector $\bar{w}_k^{\mathcal{S}_m}$ can be exponentially large in the depth of the network, the norms of the vector need not be. Because $\tilde{h}_k^{(H+1)}(x, w^{\mathcal{S}_m}) = \|\bar{x} \circ \bar{\sigma}(x, w_\sigma)\|_2 \|\bar{w}_k^{\mathcal{S}_m}\|_2 \cos\theta$, we have that $\|\bar{w}_k^{\mathcal{S}_m}\|_2 = h_k^{(H+1)}(x, w) / (\|\bar{x} \circ \bar{\sigma}(x, w_\sigma)\|_2 \cos\theta)$ (unless the denominator is zero), where θ is the angle between $\bar{x} \circ \bar{\sigma}(x, w_\sigma)$ and $\bar{w}_k^{\mathcal{S}_m}$. Additionally, as discussed in Section 4.2, $\sum_k \|\bar{w}_k\|_2^2 \leq \prod_{\ell=1}^{H+1} \|W^{(\ell)}\|_F^2$.

5. Generalization Bounds via Validation

The previous section presented a direct analysis of neural networks with squared loss to address Problems 1 and 2. While this illustrates the advantage of a direct analysis of each hypothesis space and each loss function, it requires future work to cover different cases of practical interest. Accordingly, for a general case, this section notes a simple way to avoid Problem 1, at the cost of less theoretical insight for training and additional computation for validation.

In practical deep learning, we typically adopt the training–validation paradigm, usually with a held-out validation set. We then search over hypothesis spaces by changing architectures (and other hyper-parameters) to obtain low validation error. In this view, we can conjecture the reason why deep learning can sometimes generalize well as follows: it is partially because we can obtain a good model via search using a validation dataset.

Indeed, as an example, Remark 6 states that if validation error is small, it is guaranteed to generalize well, regardless of its capacity, Rademacher complexity, stability, robustness, and flat minima. Let $S^{(\mathrm{val})}_{m_{\mathrm{val}}}$ be a held-out validation dataset of size m_{val}, which is independent of the training dataset S_m.

Proposition 6. (example of generalization guarantee via validation error) *Assume that $S^{(\mathrm{val})}_{m_{\mathrm{val}}}$ is generated by i.i.d. draws according to a true distribution P. Let $\kappa_{f,i} = R[f] - \ell(f(x_i), y_i)$ for $(x_i, y_i) \in S^{(\mathrm{val})}_{m_{\mathrm{val}}}$. Suppose that $\mathbb{E}[\kappa^2_{f,i}] \leq \gamma^2$ and $|\kappa_{f,i}| \leq C$ almost surely, for all $(f, i) \in F_{\mathrm{val}} \times \{1, \ldots, m_{\mathrm{val}}\}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f \in F_{\mathrm{val}}$:*

$$R[f] \leq \hat{R}_{S^{(\mathrm{val})}_{m_{\mathrm{val}}}}[f] + \frac{2C \ln(\frac{|F_{\mathrm{val}}|}{\delta})}{3m_{\mathrm{val}}} + \sqrt{\frac{2\gamma^2 \ln(\frac{|F_{\mathrm{val}}|}{\delta})}{m_{\mathrm{val}}}}.$$

Here, F_{val} is defined as a set of models f that is independent of a held-out validation dataset $S^{(\mathrm{val})}_{m_{\mathrm{val}}}$, but can depend on the training dataset S_m. For example, F_{val} can contain a set of models f such that each element f is a result at the end of each epoch during training with at least 99.5% *training* accuracy. In this example, $|F_{\mathrm{val}}|$ is at most (the number of epochs) $\times$ (the cardinality of the set of possible hyper-parameter settings), and is likely much smaller than that because of the 99.5% training accuracy criteria and the fact that a space of many hyper-parameters is narrowed down by using the training dataset as well as other datasets from different tasks. If a hyper-parameter search depends on the validation dataset, F_{val} must be the possible space of the search instead of the space actually visited by the search. We can also use a sequence $\{F^{(j)}_{\mathrm{val}}\}_j$ (see Appendix B).

The bound in Proposition 6 is non-vacuous and tight enough to be practically meaningful. For example, consider a classification task with 0–1 loss. Set $m_{\mathrm{val}} = 10,000$ (e.g., MNIST and CIFAR-10) and $\delta = 0.1$. Then, even in the worst case with $C = 1$ and $\gamma^2 = 1$ and even with $|F_{\mathrm{val}}| = 1,000,000,000$, we have with probability at least 0.9 that $R[f] \leq \hat{R}_{S^{(\mathrm{val})}_{m_{\mathrm{val}}}}[f] + 6.94\%$ for all $f \in F_{\mathrm{val}}$. In a non-worst-case scenario, for example, with $C = 1$ and $\gamma^2 = (0.05)^2$, we can replace 6.94% by 0.49%. With a larger validation set (e.g., ImageNet) and/or more optimistic C and γ^2, we can obtain much better bounds.

Although Proposition 6 poses the concern of increasing the generalization bound when using a single validation dataset with too large $|F_{\mathrm{val}}|$, the rate of increase is only $\ln |F_{\mathrm{val}}|$ and $\sqrt{\ln |F_{\mathrm{val}}|}$. We can also avoid dependence on the cardinality of F_{val} using Remark 7.

Remark 7. Assume that $S^{(\mathrm{val})}_{m_{\mathrm{val}}}$ is generated by i.i.d. draws according to P. Let $\mathcal{L}_{F_{\mathrm{val}}} = \{g : f \in F_{\mathrm{val}}, g(x, y) \triangleq \ell(f(x), y)\}$. By applying (Mohri et al., 2012, Theorem 3.1) to $\mathcal{L}_{F_{\mathrm{val}}}$, if the codomain of ℓ is in $[0, 1]$, with probability at least $1 - \delta$, for all $f \in \mathcal{F}_{\mathrm{val}}$, $R[f] \leq \hat{R}_{S^{(\mathrm{val})}_{m_{\mathrm{val}}}}[f] + 2\mathfrak{R}_m(\mathcal{L}_{F_{\mathrm{val}}}) + \sqrt{(\ln 1/\delta)/m_{\mathrm{val}}}$.

Unlike the standard use of Rademacher complexity with a training dataset, the set F_{val} can depend on the training dataset S_m in any manner, and hence F_{val} differs significantly from the typical hypothesis space defined by the parameterization of models. We can thus end up with a very different effective capacity and hypothesis complexity (as selected by model search using the validation set) depending on whether the training data are random or have interesting structure which the neural network can capture.

6. Discussions and Open Problems

It is very difficult to make a detailed characterization of how well a hypotheses generated by a learning algorithm will generalize, in the absence of detailed information about the given problem instance. Traditional learning theory addresses this very difficult question and has developed bounds that are as tight as possible given the generic information available. In this paper, we have worked toward drawing stronger conclusions by developing theoretical analyses tailored for the situations with more detailed information, including actual neural network structures, and actual performance on a validation set.

The bounds in Section 5 have the potential to address Problem 1, but do *not* solve Problem 2, because of the dependence on a set F_{val}. Theorem 5 partially addresses Problems 1 and 2 but leaves an issue for Problem 2 because of the subtle dependence on datasets different from a given S_m. This illustrates the important subtlety of Problem 2. Theorem 4 solves Problem 2 (and hence Problem 1) but with the limited applicability to squared loss. Future work is required to extend the direct analysis to different loss functions. Theorem 4 also suggests the possibility of solving the following open problem:

Problem 3. Tightly characterize the expected risk $R[f]$ or the generalization gap $R[f] - \hat{R}_{S_m}[f]$ of a hypothesis f with a pair (P, S_m), producing theoretical insights while partially yet provably preserving the partial order of (P, S_m, f) that is defined by the standard less-than-or-equal relation of the value $R[f]$ or $R[f] - \hat{R}_{S_m}[f]$ of (P, S_m, f).

Any theoretical insights without the partial order preservation can be misleading as it can change the ranking of the preference of (P, S_m, f). Theorem 4 solves Problem 3 by preserving the exact ordering via equality without bounds. However, for different loss functions and different hypothesis spaces, it may also be beneficial to consider a weaker notion of order preservation to gain analyzability and useful insights as in Problem 3.

Theorem 4 suggests several directions on deriving new algorithms if we notice the following fact. Instead of regularizing a possibly loose upper bound on the generalization gap, regularizing an *approximated* generalization gap would work well too in practice. Appendix A.3 proposes a family of new regularization methods, which is *not* based on our new theory, but illustrates this idea of approximation (as opposed to upper bound). In Appendix A.3, the proposed method based on the idea of approximation was empirically shown to improve base models and achieve competitive performance on MNIST and CIFAR-10 benchmarks.

Our discussion with Proposition 6 and Remark 7 suggests another open problem: analyzing the role and influence of *human intelligence* on generalization (see Appendix B for some examples). While this is a hard question, understanding it would be beneficial to further automate the role of human intelligence towards the goal of artificial intelligence.

Acknowledgements

We gratefully acknowledge support from NSF grants 1420316, 1523767 and 1723381, from AFOSR FA9550-17-1-0165, from ONR grant N00014-14-1-0486, and from ARO grant W911 NF1410433, as well as support from NSERC, CIFAR and Canada Research Chairs. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our sponsors.

References

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 2017.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Peter L Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Machine Learning*, 48(1):85–113, 2002.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–526, 2002.

Anna Choromanska, MIkael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 192–204, 2015.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, 2017.

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

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

Alon Gonen and Shai Shalev-Shwartz. Fast rates for empirical risk minimization of strict saddle problems. *arXiv preprint arXiv:1701.04271*, 2017.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

Patrick Healy and Nikola S Nikolov. How to layer a directed acyclic graph. In *International Symposium on Graph Drawing*, pages 16–30. Springer, 2001.

Ralf Herbrich and Robert C Williamson. Algorithmic luckiness. *Journal of Machine Learning Research*, 3:175–212, 2002.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *arXiv preprint arXiv:1705.08741*, 2017.

Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, 2016a.

Kenji Kawaguchi. Bounded optimal exploration in MDP. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016b.

Kenji Kawaguchi and Yoshua Bengio. Generalization in machine learning via analytical learning theory. *arXiv preprint arXiv:1802.07426*, 2018.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Bayesian optimization with exponential convergence. In *Advances in Neural Information Processing (NIPS)*, 2015.

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

Vladimir Koltchinskii and Dmitriy Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer, 2000.

Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, pages 1–50, 2002.

David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron Courville. Deep nets don't learn via memorization. In *Workshop Track of International Conference on Learning Representations*, 2017.

Ilja Kuzborskij and Christoph Lampert. Data-dependent stability of stochastic gradient descent. *arXiv preprint arXiv:1703.01678*, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1):161–193, 2006.

Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2015a.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Proceedings of The 28th Conference on Learning Theory*, pages 1376–1401, 2015b.

Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. In *International Conference on Learning Representations*, 2014.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, pages 1–17, 2017.

Ikuro Sato, Hiroki Nishimura, and Kensuke Yokoi. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*, 2015.

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

John Shawe-Taylor, Peter L Bartlett, Robert C Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory*, 44(5):1926–1940, 1998.

Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel Rodrigues. Generalization error of invariant classifiers. In *Artificial Intelligence and Statistics*, pages 1094–1103, 2017a.

Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 2017b.

Shizhao Sun, Wei Chen, Liwei Wang, Xiaoguang Liu, and Tie-Yan Liu. On the depth of deep neural networks: a theoretical view. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2066–2072. AAAI Press, 2016.

Matus Telgarsky. Benefits of depth in neural networks. In *29th Annual Conference on Learning Theory*, pages 1517–1539, 2016.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Vladimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1058–1066, 2013.

Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Pengtao Xie, Yuntian Deng, and Eric Xing. On the generalization error bounds of neural networks under diversity-inducing mutual angular regularization. *arXiv preprint arXiv:1511.07110*, 2015.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

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

Appendix

We present additional results in Appendix A, additional discussions in Appendix B, and complete proofs and extra explanations in Appendices C, D, and E.

Appendix A. Additional Results

A.1 None of small capacity, complexity, stability, robustness, and flat minima is necessary condition for generalization given a pair (P, S_m)

This statement does not contradict with necessary conditions or no free lunch theorem in previous learning theory. See Section 3.1 for the cause of the possible apparent contradiction, and see Appendix B.2 for its consistency.

Proposition 8. *Given a pair (P, S_m) and a desired $\epsilon > \inf_{f \in \mathcal{Y}^{\mathcal{X}}} R[f] - \hat{R}_{S_m}[f]$, let f^*_ϵ be a function such that $\epsilon \geq R[f^*_\epsilon] - \hat{R}_{S_m}[f^*_\epsilon]$. Then,*

(i) For any hypothesis space F whose hypothesis-space complexity is large enough to memorize any dataset and which includes f^*_ϵ possibly at an arbitrarily sharp minimum, there exist learning algorithms $\mathcal{A}$ such that the generalization gap of $f_{\mathcal{A}(S_m)}$ is at most ϵ, and

(ii) There exist arbitrarily unstable and arbitrarily non-robust algorithms $\mathcal{A}$ such that the generalization gap of $f_{\mathcal{A}(S_m)}$ is at most ϵ.

Proof. Consider statement (i). Given such a F, consider any $\mathcal{A}$ such that $\mathcal{A}$ takes F and S_m as input and outputs f^*_ϵ. Clearly, there are many such algorithms $\mathcal{A}$. For example, given a S_m, fix $\mathcal{A}$ such that $\mathcal{A}$ takes F and S_m as input and outputs f^*_ϵ (which already proves the statement), or even $f^*_\epsilon + \delta$ where δ becomes zero by the right choice of hyper-parameters and of small variations of F (e.g., architecture search in deep learning) such that F still satisfy the condition in the statement. This proves statement (i).

Consider statement (ii). Given any dataset S'_m, consider a look-up algorithm $\mathcal{A}'$ that always outputs f^*_ϵ if $S_m = S'_m$, and outputs f_1 otherwise such that f_1 is arbitrarily non-robust and $|\ell(f^*_\epsilon(x), y) - \ell(f_1(x), y)|$ is arbitrarily large (i.e., arbitrarily non-stable). This proves statement (ii). □

Note that while A' in the above proof suffices to prove statement (ii), we can also generate other non-stable and non-robust algorithms by noticing the essence captured in Remark 3.

A.2 Probabilistic bound for 0-1 loss with multi-labels

For the 0–1 loss with multi-labels, we use the Rademacher complexity for the two-phase training procedure in Section 4.3, yielding Theorem 9. Similarly to Theorems 4 and 5, Theorem 9 provides a generalization bound that does not necessarily have dependence on the number of weights, and exponential dependence on depth and effective input dimensionality. However, unlike Theorem 4, Theorem 9 does not solve Problem 2. Unlike Theorem 5, Theorem 9 does not even partially address Problem 2.

The empirical margin loss $\hat{R}^{(\rho)}_{S_m}[f]$ is defined as $\hat{R}^{(\rho)}_{S_m}[f] = \frac{1}{m}\sum_{i=1}^{m} \ell_{\text{margin},\rho}(f(x_i), y_i)$, where $\ell_{\text{margin},\rho}$ is defined as follows:

$$\ell_{\text{margin},\rho}(f(x), y) = \ell^{(2)}_{\text{margin},\rho}(\ell^{(1)}_{\text{margin},\rho}(f(x), y))$$

where

$$\ell^{(1)}_{\text{margin},\rho}(f(x), y) = h_y^{(H+1)}(x) - \max_{y \neq y'} h_{y'}^{(H+1)}(x) \in \mathbb{R},$$

and

$$\ell^{(2)}_{\text{margin},\rho}(z) = \begin{cases} 0 & \text{if } \rho \leq z \\ 1 - z/\rho & \text{if } 0 \leq z \leq \rho \\ 1 & \text{if } z \leq 0. \end{cases}$$

Theorem 9. *Assume that $S_m \setminus S_{\alpha m}$ is generated by i.i.d. draws according to true distribution P. Assume that $S_m \setminus S_{\alpha m}$ is independent of $S_{\alpha m}$. Fix $\rho > 0$ and w_σ. Let F be the set of the models with the two-phase training procedure. Suppose that $\mathbb{E}_x[\|\bar{x} \circ \bar{\sigma}(x, w_\sigma)\|_2^2] \leq C_\sigma^2$ and $\max_k \|\bar{w}_k\|_2 \leq C_w$ for all $f \in F$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f \in F$:*

$$R[f] \leq \hat{R}^{(\rho)}_{S_m \setminus S_{\alpha m}}[f] + \frac{2d_y^2(1-\alpha)^{-1/2} C_\sigma C_w}{\rho\sqrt{m_\sigma}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2m_\sigma}}.$$

Proof. Define S_{m_σ} as

$$S_{m_\sigma} = S_m \setminus S_{\alpha m} = \{(x_{\alpha m+1}, y_{\alpha m+1}), \ldots, (x_m, y_m)\}.$$

Recall the following fact: using the result by Koltchinskii and Panchenko (2002), we have that for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f \in F$:

$$R[f] \leq \hat{R}_{S_{m_\sigma},\rho}[f] + \frac{2d_y^2}{\rho m_\sigma}\mathfrak{R}'_{m_\sigma}(F) + \sqrt{\frac{\ln \frac{1}{\delta}}{2m_\sigma}},$$

where $\mathfrak{R}'_{m_\sigma}(F)$ is Rademacher complexity defined as

$$\mathfrak{R}'_{m_\sigma}(F) = \mathbb{E}_{S_{m_\sigma},\xi}\left[\sup_{k,w} \sum_{i=1}^{m_\sigma} \xi_i h_k^{(H+1)}(x_i, w)\right].$$

Here, ξ_i is the Rademacher variable, and the supremum is taken over all $k \in \{1, \ldots, d_y\}$ and all w allowed in F. Then, for our parameterized hypothesis spaces with any frozen w_σ,

$$\begin{aligned}
\mathfrak{R}'_{m_\sigma}(F) &= \mathbb{E}_{S_{m_\sigma},\xi}\left[\sup_{k,\bar{w}_k} \sum_{i=1}^{m_\sigma} \xi_i [\bar{x}_i \circ \bar{\sigma}(x_i, w_\sigma)]^\top \bar{w}_k\right] \\
&\leq \mathbb{E}_{S_{m_\sigma},\xi}\left[\sup_{k,\bar{w}_k} \left\| \sum_{i=1}^{m_\sigma} \xi_i [\bar{x}_i \circ \bar{\sigma}(x_i, w_\sigma)] \right\|_2 \|\bar{w}_k\|_2\right] \\
&\leq C_w \mathbb{E}_{S_{m_\sigma},\xi}\left[\left\| \sum_{i=1}^{m_\sigma} \xi_i [\bar{x}_i \circ \bar{\sigma}(x_i, w_\sigma)] \right\|_2\right].
\end{aligned}$$

Because square root is concave in its domain, by using Jensen's inequality and linearity of expectation,

$$\begin{aligned}
&\mathbb{E}_{S_{m\sigma},\xi}\left[\left\|\sum_{i=1}^{m_\sigma}\xi_i[\bar{x}_i\circ\bar{\sigma}(x_i,w_\sigma)]\right\|_2\right] \\
&\leq\left(\mathbb{E}_{S_{m\sigma}}\sum_{i=1}^{m_\sigma}\sum_{j=1}^{m_\sigma}\mathbb{E}_\xi[\xi_i\xi_j][\bar{x}_i\circ\bar{\sigma}(x_i,w_\sigma)]^\top[\bar{x}_j\circ\bar{\sigma}(x_j,w_\sigma)]\right)^{1/2} \\
&=\left(\sum_{i=1}^{m_\sigma}\mathbb{E}_{S_{m\sigma}}\left[\|[\bar{x}_i\circ\bar{\sigma}(x_i,w_\sigma)]\|_2^2\right]\right)^{1/2} \\
&\leq C_\sigma\sqrt{m_\sigma}.
\end{aligned}$$

Putting together, we have that $\mathfrak{R}'_m(F)\leq C_\sigma C_w\sqrt{m_\sigma}$. □

A.3 Regularization algorithms: an illustration of turning theoretical insights into algorithms via approximation instead of upper bound

In general, theoretical bounds can be too loose to be directly used in practice. Accordingly, this section illustrates the use of theoretical insight to guide search in practice based on approximation instead of upper bound.

In this section, we focus on multi-class classification problems with d_y classes, such as object classification with images. Accordingly, we analyze the expected risk with 0–1 loss as $R[f]=\mathbb{E}_x[\mathbb{1}\{f(x)=y(x)\}]$, where $f(x)=\text{argmax}_{k\in\{1,\dots,d_y\}}(h_k^{(H+1)}(x))$ is the model prediction, and $y(x)\in\{1,\dots,d_y\}$ is the true label of x (see Section 2.4.1 in Mohri et al. 2012 for an extension to stochastic labels).

A.3.1 Theoretical Insight

An application of the result by Koltchinskii and Panchenko (2002) yields the following statement: given a fixed $\rho>0$, for any $\delta>0$, with probability at least $1-\delta$, for all $f\in F$,

$$R[f]\leq\hat{R}_{m,\rho}[f]+\frac{2d_y^2}{\rho m}\mathfrak{R}'_m(F)+\sqrt{\frac{\ln\frac{1}{\delta}}{2m}},$$

where $\mathfrak{R}'_m(F)$ is a model complexity defined as

$$\mathfrak{R}'_m(F)=\mathbb{E}_{S_m,\xi}\left[\sup_{k,h_k^{(H+1)}}\sum_{i=1}^{m}\xi_i h_k^{(H+1)}(x_i)\right].$$

Here, ξ_i are the Rademacher variables (i.e., independent uniform random variables in $\{-1,+1\}$), and the supremum is taken over all $k\in\{1,\dots,d_y\}$ and all $h_k^{(H+1)}$ allowed in F.

Previous theories (Sun et al., 2016; Neyshabur et al., 2015b; Xie et al., 2015) characterize the generalization gap by the upper bounds on the model complexities via the norms of the weight matrices and exponential dependence on the depth 2^H. A close look at the proofs

reveals that the norms of weight matrices come from the Cauchy–Schwarz inequality, which can induce a very loose bound. Moreover, the exponential factor 2^H comes from bounding the effect of nonlinearity at each layer, which can also induce a loose bound, resulting in a gap between theory and practice.

This section proposes to solve this issue by directly *approximating* the model complexity $\mathfrak{R}'_m(F)$, instead of deriving a possibly too-loose bound on it (for worst possible measures). Here, the need for the approximation comes from the fact that $\mathfrak{R}'_m(F)$ contains the expectation with unknown measure on dataset S_m. The approximation of $\mathfrak{R}'_m(F)$ essentially reduces to the approximation of the expectation over the dataset S_m. In contrast to the worst-case bounds that motivated previous methods, the approximated $\mathfrak{R}'_m(F)$ and whole generalization gap do not necessarily grow along with the norms of the weights, as desired.

A.3.2 Method

The theoretical insight in the previous section suggests the following family of methods: given any architecture and method, add a new regularization term for each mini-batch as

$$\text{loss} = \text{original loss} + \frac{\lambda}{\bar{m}} \left| \max_k \sum_{i=1}^{\bar{m}} \xi_i h_k^{(H+1)}(x_i) \right|,$$

where x_i is drawn from some distribution approximating the true distribution of x, $\xi_1, \ldots, \xi_{\bar{m}}$ are independently and uniformly drawn from $\{-1, 1\}$, $\bar{m}$ is a mini-batch size and λ is a hyper-parameter. Importantly, the approximation of the true distribution of x is only used for regularization purposes and hence needs not be precisely accurate (as long as it plays its role for regularization). For example, it can be approximated by populations generated by a generative neural network and/or an extra data augmentation process. For simplicity, we call this family of methods as Directly Approximately Regularizing Complexity (DARC).

In this paper, we evaluated only a very simple version of the proposed family of methods as a first step. That is, our experiments employed the following simple and easy-to-implement method, called DARC1:

$$\text{loss} = \text{original loss} + \frac{\lambda}{\bar{m}} \left(\max_k \sum_{i=1}^{\bar{m}} |h_k^{(H+1)}(x_i)| \right), \tag{4}$$

where x_i is the i-th sample in the training mini-batch. The additional computational cost and programming effort due to this new regularization is almost negligible because $h_k^{(H+1)}(x_i)$ is already used in computing the original loss. This simplest version was derived by approximating the true distribution of x with the empirical distribution of the training data and the effect of the Rademacher variables via absolute values.

A.3.3 Experimental Results

We evaluated the proposed method (DARC1) by simply adding the new regularization term in equation (4) to existing standard codes for MNIST and CIFAR-10. For all the experiments, we fixed $(\lambda/\bar{m}) = 0.001$ with $\bar{m} = 64$. We used a single model without ensemble methods. The experimental details are in Appendix A.3.4. The source code is available at: http://lis.csail.mit.edu/code/gdl.html

Table 1: Test error (%). A standard variant of LeNet (LeCun et al., 1998) and ResNeXt-29(16×64d) (Xie et al., 2016) are used for MNIST and CIFAR-10, and compared with the addition of the studied regularizer.

Method	MNIST	CIFAR-10
Baseline	0.26	3.52
DARC1	0.20	3.43

Table 2: Test error ratio (DARC1/Base)

	MNIST (ND)		MNIST		CIFAR-10	
	mean	stdv	mean	stdv	mean	stdv
Ratio	0.89	0.61	0.95	0.67	0.97	0.79

Table 3: Values of $\frac{1}{m}\left(\max_k \sum_{i=1}^m |h_k^{(H+1)}(x_i)|\right)$

Method	MNIST (ND)		MNIST		CIFAR-10	
	mean	stdv	mean	stdv	mean	stdv
Base	17.2	2.40	8.85	0.60	12.2	0.32
DARC1	1.30	0.07	1.35	0.02	0.96	0.01

Table 1 shows the error rates comparable with previous results. To the best of our knowledge, the previous state-of-the-art classification error is 0.23% for MNIST with a single model (Sato et al., 2015) (and 0.21% with an ensemble by Wan et al. 2013). To further investigate the improvement, we ran 10 random trials with computationally less expensive settings, to gather mean and standard deviation (stdv). For MNIST, we used fewer epochs with the same model. For CIFAR-10, we used a smaller model class (pre-activation ResNet with only 18 layers). Table 2 summarizes the improvement ratio: the new model's error divided by the base model's error. We observed the improvements for all cases. The test errors (standard deviations) of the base models were 0.53 (0.029) for MNIST (ND), 0.28 (0.024) for MNIST, and 7.11 (0.17) for CIFAR-10 (all in %).

Table 3 summarizes the values of the regularization term $\frac{1}{m}(\max_k \sum_{i=1}^m |h_k^{(H+1)}(x_i)|)$ for each obtained model. The models learned with the proposed method were significantly different from the base models in terms of this value. Interestingly, a comparison of the base cases for MNIST (ND) and MNIST shows that data augmentation by itself *implicitly* regularized what we explicitly regularized in the proposed method.

A.3.4 Experimental detail

For MNIST:

We used the following fixed architecture:

Layer 1 Convolutional layer with 32 filters with filter size of 5 by 5, followed by max pooling of size of 2 by 2 and ReLU.

Layer 2 Convolution layer with 32 filters with filter size of 5 by 5, followed by max pooling of size of 2 by 2 and ReLU.

Layer 3 Fully connected layer with output 1024 units, followed by ReLU and Dropout with its probability being 0.5.

Layer 4 Fully connected layer with output 10 units.

Layer 4 outputs $h^{(H+1)}$ in our notation. For training purpose, we use softmax of $h^{(H+1)}$. Also, $f(x) = \text{argmax}(h^{(H+1)}(x))$ is the label prediction.

We fixed learning rate to be 0.01, momentum coefficient to be 0.5, and optimization algorithm to be (standard) stochastic gradient decent (SGD). We fixed data augmentation process as: random crop with size 24, random rotation up to ± 15 degree, and scaling of 15%. We used 3000 epochs for Table 1, and 1000 epochs for Tables 2 and 3.

For CIFAR-10:

For data augmentation, we used random horizontal flip with probability 0.5 and random crop of size 32 with padding of size 4.

For Table 1, we used ResNeXt-29(16×64d) (Xie et al., 2016). We set initial learning rate to be 0.05 and decreased to 0.005 at 150 epochs, and to 0.0005 at 250 epochs. We fixed momentum coefficient to be 0.9, weight decay coefficient to be 5×10^{-4}, and optimization algorithm to be stochastic gradient decent (SGD) with Nesterov momentum. We stopped training at 300 epochs.

For Tables 2 and 3, we used pre-activation ResNet with only 18 layers (pre-activation ResNet-18) (He et al., 2016). We fixed learning rate to be 0.001 and momentum coefficient to be 0.9, and optimization algorithm to be (standard) stochastic gradient decent (SGD). We used 1000 epochs.

Appendix B. Additional discussions and open problems

Theorem 4 solves Problem 2 with the limited applicability to certain neural networks with squared loss. In contrast, a parallel study (Kawaguchi and Bengio, 2018) presents a novel generic learning theory to solve Problem 2 for general cases in machine learning. It would be beneficial to explore both a generic analysis (Kawaguchi and Bengio, 2018) and a direct analysis in deep learning (this paper) to get tighter results and insights that are tailored for each particular case in deep learning.

Our discussion with Proposition 6 and Remark 7 suggests another open problem: analyzing the role and influence of *human intelligence* in generalization. For example, human intelligence seems to be able to often find good architectures (and other hyper-parameters) that get low validation errors (without non-exponentially large $|F_{\text{val}}|$ in Proposition 6, or a low complexity of $\mathcal{L}_{F_{\text{val}}}$ in Remark 7). A close look at the deep learning literature seems to suggest that this question is fundamentally related to the process of science and engineering, because many successful architectures have been designed based on the physical

properties and engineering priors of the problems at hand (e.g., hierarchical nature, convolution, architecture for motion such as that by Finn et al. 2016, memory networks, and so on). While this is a hard question, it might be beneficial to advance the understanding of human intelligence from this aspect too in order to consider partially automating it.

In previous bounds with a hypothesis space F, if we try different hypothesis spaces F depending on S_m, the basic proof breaks down. An easy recovery at the cost of an extra quantity in a bound is to take a union bound over all possible F_j for $j = 1, 2, \dots$ where we pre-decide $\{F_j\}_j$ without dependence on S_m (or simply consider the "largest" $F \supseteq F_j$, which can result in a very loose bound). Similarly, if we need to try many $w_\sigma := w^{\mathcal{S}_{\alpha m}}$ depending on the whole S_m in Theorem 5, we can take a union bound over $w_\sigma^{(j)}$ for $j = 1, 2, \dots$ in Theorem 5 where we pre-determine $\{w_\sigma^{(j)}\}_j$ without dependence on $S_m \setminus S_{\alpha m}$ but with dependence on $S_{\alpha m}$. We can do the same with Proposition 6 and Remark 7 to use many F_{val} depending on the validation dataset $S_{m_{\text{val}}}^{(\text{val})}$ with a predefined sequence.

B.1 A Relationship to Other Fields

The situation where theoretical studies focus on a set of problems and practical applications care about each element in the set is prevalent in machine learning and computer science literature, not limited to the field of learning theory. For example, for each practical problem instance $q \in Q$, the size of the set Q that had been analyzed in theory for optimal exploration in Markov decision processes (MDPs) were demonstrated to be frequently too pessimistic, and a methodology to partially mitigate the issue was proposed (Kawaguchi, 2016b). Bayesian optimization would suffer from a pessimistic set Q regarding each problem instance $q \in Q$, the issue of which was partially mitigated (Kawaguchi et al., 2015).

Moreover, characterizing a set of problems Q only via a worst-case instance $q' \in Q$ (i.e., worst-case analysis) is known to have several issues in theoretical computer science, and so-called *beyond worst-case analysis* (e.g., smoothed analysis) is an active area of research to mitigate the issues.

B.2 Consistency of Theory

Statistical learning theory can be considered to provide two types of statements relevant to the scope of this paper. The first type (which comes from upper bounds) is logically in the form of "p implies q," where $p :=$ "the hypothesis-space complexity is small" (or another statement about stability, robustness, or flat minima), and $q :=$ "the generalization gap is small." Notice that "p implies q" does not imply "q implies p." Thus, based on statements of this type, it is entirely possible that the generalization gap is small even when the hypothesis-space complexity is large or the learning mechanism is unstable, non-robust, or subject to sharp minima.

The second type (which comes from lower bounds) is logically in the following form: in a set U_{all} of all possible problem configurations, there exists a subset $U \subseteq U_{\text{all}}$ such that "q implies p" in U (with the same definitions of p and q as in the previous paragraph). For example, Mohri et al. (2012, Section 3.4) derived lower bounds on the generalization gap by showing the existence of a "bad" distribution that characterizes U. Similarly, the classical *no free lunch* theorems are the results with the existence of a worst-case distribution for each algorithm. However, if the problem instance at hand (e.g., object classification with

MNIST or CIFAR-10) is not in such a U in the proofs (e.g., if the data distribution is not among the "bad" ones considered in the proofs), q does not necessarily imply p. Thus, it is still naturally possible that the generalization gap is small with large hypothesis-space complexity, instability, non-robustness, and sharp minima. Therefore, there is no contradiction or paradox.

B.3 Practical Role of Generalization Theory

From the discussions above, we can see that there is a *logically expected* difference between the scope in theory and the focus in practice; it is logically expected that there are problem instances where theoretical bounds are pessimistic. In order for generalization theory to have maximal impact in practice, we must be clear on a set of different roles it can play regarding practice, and then work to extend and strengthen it in each of these roles. We have identified the following practical roles for theory:

Role 1 Provide guarantees on expected risk.

Role 2 Guarantee generalization gap

- Role 2.1 to be small for a given fixed S_m, and/or
- Role 2.2 to approach zero with *a fixed model class* as m increases.

Role 3 Provide theoretical insights to guide the search over model classes.

Appendix C. Appendix for Section 3

C.1 Proof of Theorem 1

Proof. For any matrix M, let $\text{Col}(M)$ and $\text{Null}(M)$ be the column space and null space of M. Since $\text{rank}(\Phi) \geq m$ and $\Phi \in \mathbb{R}^{m \times n}$, the set of its columns span $\mathbb{R}^m$, which proves statement (i). Let $w^* = w_1^* + w_2^*$ where $\text{Col}(w_1^*) \subseteq \text{Col}(M^T)$ and $\text{Col}(w_2^*) \subseteq \text{Null}(M)$. For statement (ii), set the parameter as $w := w_1^* + \epsilon C_1 + \alpha C_2$ where $\text{Col}(C_1) \subseteq \text{Col}(M^T)$, $\text{Col}(C_2) \subseteq \text{Null}(M)$, $\alpha \geq 0$ and $C_2 = \frac{1}{\alpha} w_2^* + \bar{C}_2$. Since $\text{rank}(M) < n$, $\text{Null}(M) \neq \{0\}$ and there exist non-zero $\bar{C}_2$. Then,

$$\hat{Y}(w) = Y + \epsilon \Phi C_1,$$

and

$$\hat{Y}_{\text{test}}(w) = Y_{\text{test}} + \epsilon \Phi_{\text{test}} C_1.$$

By setting $A = \Phi C_1$ and $B = \Phi_{\text{test}} C_1$ with a proper normalization of C_1 yields (a) and (b) in statement (ii) (note that C_1 has an arbitrary freedom in the bound on its scale because its only condition is $\text{Col}(C_1) \subseteq \text{Col}(M^T)$). At the same time with the same parameter, since $\text{Col}(w_1^* + \epsilon C_1) \perp \text{Col}(C_2)$,

$$\|w\|_F^2 = \|w_1^* + \epsilon C_1\|_F^2 + \alpha^2 \|C_2\|_F^2,$$

and

$$\|w - w^*\|_F^2 = \|\epsilon C_1\|_F^2 + \alpha^2 \|\bar{C}_2\|_F^2,$$

which grows unboundedly as $\alpha \to \infty$ without changing A and B, proving (c) in statement (ii). □

C.2 Proof of Corollary 2

Proof. It follows the fact that the proof in Theorem 1 uses the assumption of $n > m$ and $\text{rank}(\Phi) \geq m$ only for statement (i). □

Appendix D. Appendix for Section 4

We use the following lemma in the proof of Theorem 4.

Lemma 10. (Matrix Bernstein inequality: corollary to theorem 1.4 in Tropp 2012) *Consider a finite sequence $\{M_i\}$ of independent, random, self-adjoint matrices with dimension d. Assume that each random matrix satisfies that $\mathbb{E}[M_i] = 0$ and $\lambda_{\max}(M_i) \leq R$ almost surely. Let $\gamma^2 = \| \sum_i \mathbb{E}[M_i^2] \|_2$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\lambda_{\max}\left(\sum_i M_i\right) \leq \frac{2R}{3} \ln \frac{d}{\delta} + \sqrt{2\gamma^2 \ln \frac{d}{\delta}}.$$

Proof. Theorem 1.4 by Tropp (2012) states that for all $t \geq 0$,

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_i M_i\right) \geq t\right] \leq d \cdot \exp\left(\frac{-t^2/2}{\gamma^2 + Rt/3}\right).$$

Setting $\delta = d \exp\left(-\frac{t^2/2}{\gamma^2 + Rt/3}\right)$ implies

$$-t^2 + \frac{2}{3} R(\ln d/\delta)t + 2\gamma^2 \ln d/\delta = 0.$$

Solving for t with the quadratic formula and bounding the solution with the subadditivity of square root on non-negative terms (i.e., $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for all $a, b \geq 0$),

$$t \leq \frac{2}{3} R(\ln d/\delta) + 2\gamma^2 \ln d/\delta.$$

□

D.1 Proof of Theorem 4

Proof. From Equation (1), the squared loss of deep models for each point (x, y) can be rewritten as

$$\sum_{k=1}^{d_y} (z^\top \bar{w}_k - y_k)^2 = \sum_{k=1}^{d_y} \bar{w}_k^\top (zz^\top) \bar{w}_k - 2y_k z^\top \bar{w}_k + y_k^2.$$

Thus, from Equation (1) with the squared loss, we can decompose the generalization gap into three terms as

$$\begin{aligned} R[w^{\mathcal{S}_m}] - \hat{R}_{S_m}[w^{\mathcal{S}_m}] = &\sum_{k=1}^{d_y} \left[(\bar{w}_k^{\mathcal{S}_m})^\top \left(\mathbb{E}[zz^\top] - \frac{1}{m} \sum_{i=1}^{m} z_i z_i^\top \right) \bar{w}_k^{\mathcal{S}_m} \right] \\ &+ 2 \sum_{k=1}^{d_y} \left[\left(\frac{1}{m} \sum_{i=1}^{m} y_{ik} z_i^\top - \mathbb{E}[y_k z^\top] \right) \bar{w}_k^{\mathcal{S}_m} \right] \\ &+ \mathbb{E}[y^\top y] - \frac{1}{m} \sum_{i=1}^{m} y_i^\top y_i. \end{aligned}$$

As G is a real symmetric matrix, we denote an eigendecomposition of G as $G = U \Lambda U^\top$ where the diagonal matrix Λ contains eigenvalues as $\Lambda_{jj} = \lambda_j$ with the corresponding orthogonal eigenvector matrix U; u_j is the j-th column of U. Then,

$$(\bar{w}_k^{\mathcal{S}_m})^\top G \bar{w}_k^{\mathcal{S}_m} = \sum_j \lambda_j (u_j^\top \bar{w}_k^{\mathcal{S}_m})^2 = \|\bar{w}_k^{\mathcal{S}_m}\|_2^2 \sum_j \lambda_j \cos^2 \theta^{(1)}_{\bar{w}_k^{\mathcal{S}_m}, j},$$

and

$$\sum_j \lambda_j (u_j^\top \bar{w}_k^{\mathcal{S}_m})^2 \leq \lambda_{\max}(G) \|U^\top \bar{w}_k^{\mathcal{S}_m}\|_2^2 = \lambda_{\max}(G) \|\bar{w}_k^{\mathcal{S}_m}\|_2^2.$$

Also,

$$v^\top \bar{w}_k^{\mathcal{S}_m} = \|v\|_2 \|\bar{w}_k^{\mathcal{S}_m}\|_2 \cos \theta^{(2)}_{\bar{w}_k^{\mathcal{S}_m}} \leq \|v\|_2 \|\bar{w}_k^{\mathcal{S}_m}\|_2.$$

By using these,

$$\begin{aligned} R[w^{\mathcal{S}_m}] - \hat{R}_{S_m}[w^{\mathcal{S}_m}] - c_y &= \textstyle\sum_{k=1}^{d_y} \left(2\|v\|_2 \|\bar{w}_k^{\mathcal{S}_m}\|_2 \cos \theta^{(2)}_{\bar{w}_k^{\mathcal{S}_m}} + \|\bar{w}_k^{\mathcal{S}_m}\|_2^2 \sum_j \lambda_j \cos^2 \theta^{(1)}_{\bar{w}_k^{\mathcal{S}_m}, j} \right) \\ &\leq \sum_{k=1}^{d_y} \left(2\|v\|_2 \|\bar{w}_k^{\mathcal{S}_m}\|_2 + \lambda_{\max}(G) \|\bar{w}_k^{\mathcal{S}_m}\|_2^2 \right). \end{aligned}$$

□

D.2 SGD Chooses Direction only in terms of $\bar{w}$ but not in terms of w in z

Recall that

$$h_k^{(H+1)}(x, w) = z^\top \bar{w} = [\bar{x} \circ \bar{\sigma}(x, w)]^\top \bar{w}.$$

Note that $\sigma(x, w)$ is 0 or 1 for max-pooling and/or ReLU nonlinearity. Thus, the derivative of $z = [\bar{x} \circ \bar{\sigma}(x, w)]$ with respect to w is zero everywhere (except at the measure zero set where the derivative does not exists). Thus, by the chain rule (and power rule), the gradient of the loss with respect to w only contain the contribution from the derivative of $h_k^{(H+1)}$ with respect to $\bar{w}$, but not with respect to w in z.

D.3 Simple Implementation of Two-Phase Training Procedure

Directly implementing Equation (3) requires the summation over all paths, which can be computationally expensive. To avoid it, we implemented it by creating two deep neural networks, one of which defines $\bar{w}$ paths hierarchically, and another of which defines w_σ paths hierarchically, resulting in the computational cost at most (approximately) twice as much as the original cost of training standard deep learning models. We tied w_σ and $\bar{w}$ in the two networks during standard phase, and untied them during freeze phase.

Our source code is available at:

http://lis.csail.mit.edu/code/gdl.html

The computation of the standard network without skip connection can be re-written as:

$$\begin{aligned} h^{(l)}(x,w) &= \sigma^{(l)}(W^{(l)}h^{(l-1)}(x,w)) \\ &= \dot{\sigma}^{(l)}(W^{(l)}h^{(l-1)}(x,w)) \circ W^{(l)}h^{(l-1)}(x,w) \\ &= \dot{\sigma}^{(l)}(W_\sigma^{(l)}h_\sigma^{(l-1)}(x,w)) \circ W^{(l)}h^{(l-1)}(x,w) \end{aligned}$$

where $W_\sigma^{(l)} := W^{(l)}$, $h_\sigma^{(l-1)} := \sigma(W_\sigma^{(l)}h_\sigma^{(l-1)}(x,w))$ and $\dot{\sigma}_j^{(l)}(W^{(l)}h^{(l-1)}(x,w)) = 1$ if the j-th unit at the l-th layer is active, $\dot{\sigma}_j^{(l)}(W^{(l)}h^{(l-1)}(x,w)) = 0$ otherwise. Note that because $W_\sigma^{(l)} = W^{(l)}$, we have that $h_\sigma^{(l-1)} = h^{(l)}$ in standard phase and standard models.

In the two-phase training procedure, we created two networks for $W_\sigma^{(l)}h_\sigma^{(l-1)}(x,w)$ and $W^{(l)}h^{(l-1)}(x,w)$ separately. We then set $W_\sigma^{(l)} = W^{(l)}$ during standard phase, and frozen $W_\sigma^{(l)}$ and only trained $W^{(l)}$ during freeze phase. By following the same derivation of Equation (1), we can see that this defines the desired computation without explicitly computing the summation over all paths. By the same token, this applies to DAGs.

D.4 Experimental detail in Section 4.3.1

For all MNIST(ND), MNIST and CIFAR-10, we used the same settings as those for Tables 2 and 3 in Section A.3.3. That is, for all experiments, we still used the same fixed value of $(\lambda/\bar{m}) = 0.001$ with $\bar{m} = 64$. Other experimental detail is also identical to what is described in Appendix A.3.4, except that we used 1000 epochs for each standard and freeze phase.

D.5 Proof of Theorem 5

Proof. We do *not* require the independence over the coordinates of $\tilde{z}_i$ and the entries of random matrices $\tilde{z}_i\tilde{z}_i^\top$ because of the definition of independence required for matrix Bernstein inequality (for $\frac{1}{m_\sigma}\sum_{i=1}^{m_\sigma}\tilde{z}_i\tilde{z}_i^\top$) (e.g., see section 2.2.3 of Tropp et al. 2015) and because of a union bound over coordinates (for $\frac{1}{m_\sigma}\sum_{i=1}^{m_\sigma}y_{ik}\tilde{z}_i$). We use the fact that $\tilde{z}_{\alpha m+1},\ldots,\tilde{z}_m$ are independent random variables *over the sample index (although dependent over the coordinates)*, because a $w_\sigma := w^{\mathcal{S}_{\alpha m}}$ is fixed and independent from $x_{\alpha m+1},\ldots,x_m$.

From Equation (2), with the definition of induced matrix norm and the Cauchy-Schwarz inequality,

$$R[f_{\mathcal{A}(S_m)}] - \hat{R}_{S_m \setminus S_{\alpha m}}[f_{\mathcal{A}(S_m)}] \leq \sum_{k=1}^{d_y} \left\|\bar{w}_k^{S_m}\right\|_2^2 \lambda_{\max}\left(\mathbb{E}[\tilde{z}\tilde{z}^\top] - \frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} \tilde{z}_i\tilde{z}_i^\top\right) \tag{5}$$
$$+ 2\sum_{k=1}^{d_y} \|\bar{w}_k^{S_m}\|_1 \left\|\frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} y_{ik}\tilde{z}_i - \mathbb{E}[y_k\tilde{z}]\right\|_\infty$$
$$+ \mathbb{E}[y^\top y] - \frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} y_i^\top y_i.$$

In the below, we bound each term of the right-hand side of the above with concentration inequalities.

For the first term: Matrix Bernstein inequality (Lemma 10) states that for any $\delta > 0$, with probability at least $1 - \delta/3$,

$$\lambda_{\max}\left(\mathbb{E}[\tilde{z}\tilde{z}^\top] - \frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} \tilde{z}_i\tilde{z}_i^\top\right) \leq \frac{2C_{zz}}{3m_\sigma}\ln\frac{3d_z}{\delta} + \sqrt{\frac{2\gamma_{zz}^2}{m_\sigma}\ln\frac{3d_z}{\delta}}.$$

Here, Matrix Bernstein inequality was applied as follows. Let $M_i = (\frac{1}{m_\sigma}G^{(i)})$. Then, $\sum_{i=\alpha m+1}^{m} M_i = \mathbb{E}[\tilde{z}\tilde{z}^\top] - \frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m}\tilde{z}_i\tilde{z}_i^\top$. We have that $\mathbb{E}[M_i] = 0$ for all i. Also, $\lambda_{\max}(M_i) \leq \frac{1}{m_\sigma}C_{zz}$ and $\|\sum_i \mathbb{E}[M_i^2]\|_2 \leq \frac{1}{m_\sigma}\gamma_{zz}^2$.

For the second term: We apply Bernstein inequality to each $(k, k') \in \{1, \ldots, d_y\} \times \{1, \ldots, d_z\}$ and take union bound over $d_y d_z$ events, obtaining that for any $\delta > 0$, with probability at least $1 - \delta/3$, for all $k \in \{1, 2, \ldots, d_y\}$,

$$\left\|\frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} y_{ik}\tilde{z}_i - \mathbb{E}[y_k\tilde{z}]\right\|_\infty \leq \frac{2C_{yz}}{3m_\sigma}\ln\frac{6d_y d_z}{\delta} + \sqrt{\frac{\gamma_{yz}^2}{m_\sigma}\ln\frac{6d_y d_z}{\delta}}$$

For the third term: From Bernstein inequality, with probability at least $1 - \delta/3$,

$$\mathbb{E}[y^\top y] - \frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} y_i^\top y_i \leq \frac{2C_y}{3m}\ln\frac{3}{\delta} + \sqrt{\frac{2\gamma_y^2}{m}\ln\frac{3}{\delta}}.$$

Putting together: Putting together, for a fixed (or frozen) w_σ, with probability at least $1 - \delta$ (probability over $S_m \setminus S_{\alpha m} = \{(x_{\alpha m+1}, y_{\alpha m+1}), \ldots, (x_m, y_m)\}$), we have that $\lambda_{\max}\left(\mathbb{E}[\tilde{z}\tilde{z}^\top] - \frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m}\tilde{z}_i\tilde{z}_i^\top\right) \leq \beta_1$, $\left\|\frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} y_{ik}\tilde{z}_i - \mathbb{E}[y_k\tilde{z}]\right\|_\infty \leq \beta_2$ (for all k), and $\mathbb{E}[y^\top y] - \frac{1}{m_\sigma}\sum_{i=\alpha m+1}^{m} y_i^\top y_i \leq \beta_3$. Since Equation (5) always hold deterministically (with or without such a dataset), the desired statement of this theorem follows. □

Appendix E. Appendix for Section 5

E.1 Proof of Proposition 6

Proof. Consider a single fixed $f \in F_{\text{val}}$. Since F_{val} is independent from the validation dataset, $\kappa_{f,1}, \ldots, \kappa_{f,m_{\text{val}}}$ are independent zero-mean random variables, given a fixed $f \in F_{\text{val}}$. Thus, we can apply Bernstein inequality, yielding

$$\mathbb{P}\left(\frac{1}{m_{\text{val}}}\sum_{i=1}^{m_{\text{val}}}\kappa_{f,i} > \epsilon\right) \leq \exp\left(-\frac{\epsilon^2 m_{\text{val}}/2}{\gamma^2 + \epsilon C/3}\right).$$

By taking union bound over all elements in F_{val},

$$\mathbb{P}\left(\cup_{f\in F_{\text{val}}}\left\{\frac{1}{m_{\text{val}}}\sum_{i=1}^{m_{\text{val}}}\kappa_{f,i} > \epsilon\right\}\right) \leq |F_{\text{val}}|\exp\left(-\frac{\epsilon^2 m_{\text{val}}/2}{\gamma^2 + \epsilon C/3}\right).$$

By setting $\delta = |F_{\text{val}}|\exp\left(-\frac{\epsilon^2 m_{\text{val}}/2}{\gamma^2+\epsilon C/3}\right)$ and solving for ϵ (via quadratic formula),

$$\epsilon = \frac{2C\ln(\frac{|F_{\text{val}}|}{\delta})}{6m_{\text{val}}} \pm \frac{1}{2}\sqrt{\left(\frac{2C\ln(\frac{|F_{\text{val}}|}{\delta})}{3m_{\text{val}}}\right)^2 + \frac{8\gamma^2\ln(\frac{|F_{\text{val}}|}{\delta})}{m_{\text{val}}}}.$$

By noticing that the solution of ϵ with the minus sign results in $\epsilon < 0$, which is invalid for Bernstein inequality, we obtain the valid solution with the plus sign. Then, we have

$$\epsilon \leq \frac{2C\ln(\frac{|F_{\text{val}}|}{\delta})}{3m_{\text{val}}} + \sqrt{\frac{2\gamma^2\ln(\frac{|F_{\text{val}}|}{\delta})}{m_{\text{val}}}},$$

where we used that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. By tanking the negation of the statement, we obtain that for any $\delta > 0$, with probability at least $1-\delta$, for all $f \in F_{\text{val}}$,

$$\frac{1}{m_{\text{val}}}\sum_{i=1}^{m_{\text{val}}}\kappa_{f,i} \leq \frac{2C\ln(\frac{|F_{\text{val}}|}{\delta})}{3m_{\text{val}}} + \sqrt{\frac{2\gamma^2\ln(\frac{|F_{\text{val}}|}{\delta})}{m}},$$

where $\frac{1}{m_{\text{val}}}\sum_{i=1}^{m_{\text{val}}}\kappa_{f,i} = R[f] - \hat{R}_{S^{(\text{val})}_{m_{\text{val}}}}[f]$. □