DECISION BOUNDARY VARIABILITY AND GENERALIZATION IN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing works suggest that the generalizability is guaranteed when the margin between data and decision boundaries is sufficiently large. However, the existence of adversarial examples in neural networks shows that excellent generalization and small margin can exist simultaneously, which casts shadows to the current understanding. This paper discovers that the neural network with lower decision boundary (DB) variability has better generalizability. Two new notions, *algorithm DB variability* and (ϵ, η)*-data DB variability*, are proposed to measure the decision boundary variability from the algorithm and data perspectives. Extensive experiments show significant negative correlations between the decision boundary variability and the generalizability. From the theoretical view, we prove two lower bounds and two upper bounds of the generalization error based on the decision boundary variability, which is consistent with our empirical results. Moreover, the bounds do not explicitly depend on the network size, which is usually prohibitively large in deep learning.

1 INTRODUCTION

Neural networks (NNs) have achieved significant success in vast applications (Krizhevsky et al., 2012; Vaswani et al., 2017). However, the advance of NNs is arduous to be characterized by canonical statistical hypothesis complexity, such as VC-dimension (Vapnik et al., 1994) or Rademacher complexity (Bartlett & Mendelson, 2002). Some works attempt to address this challenge from the perspective that inspects the margins from the data to the decision boundaries (DBs). Lyu & Li (2019), Poggio et al. (2020) and Xu et al. (2021) show that over-parameterized NNs tend to locally maximize the margin during training, which coincides with the behaviors of support vector machine (SVM). However, the existence of adversarial examples reveals that the neural networks still hold satisfied generalization ability with surprisingly small margins. Moreover, the performance degradation in adversarial training (Zhang et al., 2019), which enlarges the minimum margins, indicates that larger margins might not imply better generalization of NNs. This casts a shadow on the current explanation via margins.

Unlike inspecting margins in decision boundaries and considering the intimate connection between low variance and significant performance in machine learning, we investigate decision boundaries from the perspective of variability in this paper. For neural networks, the decision boundary variability is largely caused by (1) the randomness from the training algorithm, and (2) the training data. In this paper, they are measured by two new terms, *algorithm DB variability* and (ϵ, η)*-data DB variability*.

Algorithm DB variability measures the variability of DBs in different training repeats. Two major effects are discovered in the algorithm DB variability: (1) *simplicity effect:* the simplicity of data implies the network converges to the specific max-margin solution and hence contributes to small algorithm DB variability; and (2) *constraint effect:* the small inter-class distance constrains the variation of decision boundaries and reduces the algorithm DB variability. Extensive experiments are conducted on the two half moon (Pedregosa et al., 2011), CIFAR-10, and CIFAR-100 (Krizhevsky et al., 2009) datasets. We visualize the trend of the algorithm DB variability with respect to (*w.r.t.*) different data non-linearity levels of training data and model training settings (i.e., model size, training strategy). The empirical results suggest that (1) the simplicity effect is dominant in most cases comparing to the constraint effect; (2) larger model size contributes to smaller

algorithm DB variability of NNs; and (3) lower algorithm DB variability coincides with higher test accuracy. These results suggest that the algorithm decision boundary variability largely indicates the generalization of neural networks. From the theoretical view, two lower bounds and an upper bound of the generalization error are proved based on the algorithm DB variability. The upper bound has a rate of $\mathcal{O}\left(\sqrt{AV\left(f_{\boldsymbol{\theta}}, \mathcal{D}\right)/mk}\right)$, where $AV(f_{\mathbb{Q}}, \mathcal{D})$ is the algorithm DB variability for $f_{\mathbb{Q}}$ on data generation distribution $\mathcal{D}$, m is the sample size, and k is the number of classes.

To present how the training data affects the decision boundary variability, the (ϵ, η)-data DB variability is introduced by employing the η-subset to reconstruct decision boundaries with error ϵ (see Definition 5.1). If a decision boundary can be reconstructed by training on a smaller η-subset with a smaller reconstruction error ϵ, the decision boundary possesses smaller data DB variability. Moreover, an η-ϵ curve can be drawn by varying the value of η and numerically evaluating the label mismatch rate. The area under the η-ϵ curve could be a more elaborate indicator for the generalization of NNs because the curve contains richer information than the algorithm DB variability. An $\mathcal{O}(\frac{1}{\sqrt{m}} + \epsilon + \eta \log \frac{1}{\eta})$ generalization bound based on the (ϵ, η)-data DB variability is established theoretically to enhance the relationship between the generalization of NNs and the variability of DBs. Compared to the algorithm DB variability upper bound, the data DB variability bound does not depend on k, and both upper bounds do not explicitly depend on the network size, which is usually prohibitively large in deep learning.

2 Related works

Deep learning theory. Due to the bias-variance trade-off, model complexity faces a dilemma in conventional wisdom (Mohri et al., 2018). Recently, Zhang et al. (2021) reveal the surprising ability of NNs in fitting noise, but the NNs still have impressive generalization performance in practice. This gap between the well-known bias-variance trade-off and the universal approximation ability for NNs draws attention to numerous researchers (Belkin et al., 2019; Nakkiran et al., 2019). Many works attribute the success of NNs to the effectiveness of the stochastic gradient descent (SGD) algorithm (Bottou, 2010; Hardt et al., 2016; Jin et al., 2017). Some empirical studies also explain the decent performance of networks by uncovering their learning properties (Nakkiran et al., 2020; Jiang et al., 2021; He et al., 2021). For instance, neural networks tend to fit the low-frequency information first (Rahaman et al., 2019; Xu et al., 2019) and gradually learn a more complex function (Kalimeris et al., 2019) during the training procedure.

Decision boundaries and generalization. Recent studies attempt to understand neural networks from the aspect of decision boundaries (He et al., 2018; Karimi et al., 2019). Soudry et al. (2018) show that the over-parameterized networks converge to the max-margin solution when the training data is linear-separable. Some follow-up research has also been conducted along this line (Ji & Telgarsky, 2018b; Lyu & Li, 2019; Chizat & Bach, 2020). More recently, researchers find that NNs only rely on the most discriminative or simplest features to construct the DBs (Ortiz-Jimenez et al., 2020; Shah et al., 2020). Instead, our approach is different from these former methods by considering decision boundary variability, which is empirically and theoretically shown to closely correlate with the generalization in neural networks.

Adversarial training and generalization. It has been shown that the adversarial examples, which are created by adding non-perceivable perturbation on the input data, can completely mislead the NNs (Szegedy et al., 2013; Goodfellow et al., 2014). To tackle this problem, adversarial training is proposed to improve the robustness of the NNs through continuous training on adversarial examples (Madry et al., 2017). Since the adversarial training could enlarge the margin, due to the conventional wisdom, the generalization error should be smaller. Nevertheless, Su et al. (2018) and Zhang et al. (2019) show a trade-off between the robustness and the generalization performance of NNs.

3 Preliminaries

3.1 Notation

We denote the training set by $\mathcal{S} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, m\}$, where $\mathbf{x}_i \in \mathbb{R}^n$, n is the dimension of input data, $y_i \in \{1, \ldots, k\}$, k is the number of classes, and $m = |\mathcal{S}|$ is the training sample size.

We assume that $(\mathbf{x}_i, y_i)$ are independent and identically distributed (i.i.d.) random variables drawn from the data generating distribution $\mathcal{D}$. Denote the classifier as $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$, which is a neural network parameterized by $\boldsymbol{\theta}$. The output of $f_{\boldsymbol{\theta}}(\mathbf{x})$ is a k-dimensional vector and is assumed to be a discrete probability density function. Let $f_{\boldsymbol{\theta}}^{(i)}(\mathbf{x})$ be the i-th component of $f_{\boldsymbol{\theta}}(\mathbf{x})$, hence $\sum_{i=1}^{k} f_{\boldsymbol{\theta}}^{(i)}(\mathbf{x}) = 1$. We define $T(f_{\boldsymbol{\theta}}, \mathbf{x}) = \{i \in \{1, \cdots, k\} | f_{\boldsymbol{\theta}}^{(i)}(\mathbf{x}) = \max_j f_{\boldsymbol{\theta}}^{(j)}(\mathbf{x})\}$ to denote the set of predicted labels by $f_{\boldsymbol{\theta}}$ on $\mathbf{x}$. Due to the randomness of the learning algorithm $\mathcal{A}$, let $\mathbb{Q}(\boldsymbol{\theta}) = \mathcal{A}(\mathcal{S})$ denote the posteriori distribution returned by the training algorithm $\mathcal{A}$ leveraged on the training set $\mathcal{S}$. Hence, we focus on the *Gibbs classifier* (a.k.a. random classifier) $f_{\mathbb{Q}} = \{f_{\boldsymbol{\theta}} | \boldsymbol{\theta} \sim \mathbb{Q}\}$. $0-1$ loss is employed in this paper, and the expected risks in terms of $\boldsymbol{\theta}$ and $\mathbb{Q}$ are defined as:

$$\mathcal{R}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\mathbb{I}\left(y \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] \tag{1}$$

and

$$\mathcal{R}_{\mathcal{D}}(\mathbb{Q}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}} \left[\mathbb{I}\left(y \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right], \tag{2}$$

respectively. Here $\mathbb{I}(\cdot)$ is the indicator function. Since the data generating distribution $\mathcal{D}$ is unknown, evaluating the expected risk $\mathcal{R}_{\mathcal{D}}$ is not practical. Therefore, it is a practical way to estimate the expected risk by the empirical risk $\mathcal{R}_{\mathcal{S}}$, which is defined as:

$$\mathcal{R}_{\mathcal{S}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}} \left[\mathbb{I}\left(y \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left(y_i \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}_i\right)\right) \tag{3}$$

$$\mathcal{R}_{\mathcal{S}}(\mathbb{Q}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}} \left[\mathbb{I}\left(y \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}} \left[\mathbb{I}\left(y_i \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}_i\right)\right)\right], \tag{4}$$

where $(\mathbf{x}_i, y_i) \in \mathcal{S}$ and $m = |\mathcal{S}|$.

3.2 Decision boundary

If the output k-dimensional vector $f_{\boldsymbol{\theta}}(\mathbf{x})$ on the input example $\mathbf{x}$ has a tie, *i.e.*, the maximum value of the vector is not unique, then $\mathbf{x}$ is considered to locate on the decision boundary of $f_{\boldsymbol{\theta}}$.

Definition 3.1 (decision boundary)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$. Then the *decision boundary* of $f_{\boldsymbol{\theta}}$ is defined by

$$\{\mathbf{x} \in \mathbb{R}^n | \exists i, j \in \{1, \cdots, k\}, i \neq j, f_{\boldsymbol{\theta}}^{(i)}(\mathbf{x}) = f_{\boldsymbol{\theta}}^{(j)}(\mathbf{x}) = \max_q f_{\boldsymbol{\theta}}^{(q)}(\mathbf{x})\} \tag{5}$$

After defining the decision boundary, we have the following remark:

Remark 1. (1) If a input example $(\mathbf{x}, y)$ is not located on the decision boundary of $f_{\boldsymbol{\theta}}$, $T(f_{\boldsymbol{\theta}}, \mathbf{x})$ is a singleton, and we have

$$\mathbb{I}\left(y \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right) = \mathbb{I}\left(y \neq T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right). \tag{6}$$

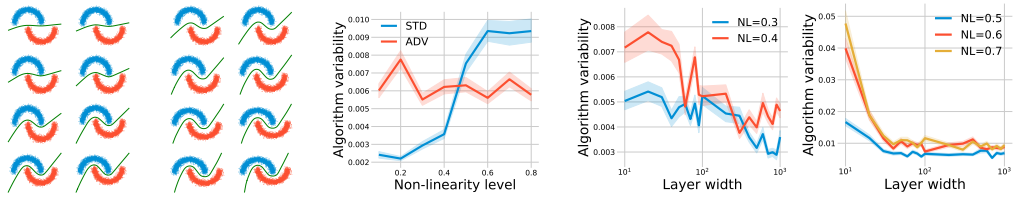
(2) If the input $\mathbf{x}$ is a boundary point, in practice, we randomly draw a label from the set $T(f_{\boldsymbol{\theta}}, \mathbf{x})$ as the prediction of $f_{\boldsymbol{\theta}}$ on $\mathbf{x}$.

3.3 Adversarial training

Adversarial training (AT) is a popular strategy to enhance the adversarial robustness of neural networks against adversarial examples, which is generated through projected gradient descent (PGD) (Madry et al., 2018) in our empirical studies. More specifically, adversarial training can be formulated as solving the minimax-loss problem as follows:

$$\min_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} \max_{\|\mathbf{x}_i' - \mathbf{x}_i\| \leq \gamma} \ell\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i'\right), y_i\right), \tag{7}$$

where γ is the *radius* to limit the distance between adversarial examples and original examples. Adversarial training can be considered to enlarge the minimum margins from training examples to decision boundaries to at least γ.



(a) STD (b) ADV (c) AV vs. S complexity (d) AV vs. layer width

Figure 1: Algorithm decision boundary variability on two half moon datasets. (a) Decision boundaries of moon datasets with standard training (STD). (b) Decision boundaries of two half moon datasets with adversarial training (ADV). (c) Plot of algorithm DB variability (AV) as a function of sample complexity (S complexity) on two half moon datasets. (d) Plots of algorithm DB variability (AV) as a function of layer widths on different non-linear (NL) two half moon datasets. The darker lines show the average over seeds and the shaded area shows the standard deviations

4 ALGORITHM DECISION BOUNDARY VARIABILITY

Due to the randomness of learning algorithms, there is no doubt that the parameters have a substantial variation when the network is repeatedly trained. However, *do the decision boundaries in these repeated-training networks have a large discrepancy?* With this question, we define the algorithm decision boundary variability (AV) to measure the variability of DBs caused by the randomness of algorithms in different training repeats.

Definition 4.1 (algorithm decision boundary variability)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$ and let $\mathbb{Q}(\boldsymbol{\theta})$ be the distribution over $\boldsymbol{\theta}$. Then, the algorithm decision boundary variability for $f_{\mathbb{Q}}$ on $\mathcal{D}$ is

$$AV(f_{\mathbb{Q}}, \mathcal{D}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}} \left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}}, \mathbf{x}) \neq T(f_{\boldsymbol{\theta}'}, \mathbf{x}) \right) \right], \tag{8}$$

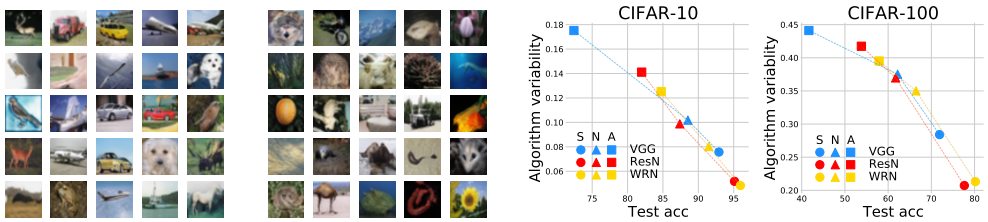
where $T(f_{\boldsymbol{\theta}}, \mathbf{x}) = \{i \in \{1, \cdots, k\} | f_{\boldsymbol{\theta}}^{(i)}(\mathbf{x}) = \max_j f_{\boldsymbol{\theta}}^{(j)}(\mathbf{x})\}$.

In the light of Definition 4.1, smaller algorithm DB variability $AV(f_{\mathbb{Q}}, \mathcal{D})$ means that the network produces more stable DBs or converges to more similar DBs during different training repeats. The algorithm DB variability is also a good notion for the "entropy" of decision boundaries because it measures the decision uncertainty over the whole data generating distribution; we provide a detailed discussion in Appendix B.1 due to space limitation.

4.1 ALGORITHM DB VARIABILITY ABOUT SAMPLE COMPLEXITY

According to Soudry et al. (2018); Ji & Telgarsky (2018a), neural networks converge to the max-margin solutions on the linearly separable data, which coincides with the behaviors of SVM. In this case, the decision boundaries are identical when we repeatedly train the network, and hence the algorithm DB variability is zero. The linearly separable scenario indicates that simple data correlates with the low algorithm DB variability, and the decline in the algorithm DB variability caused by the simplicity of data is termed as *simplicity effect*. On the other hand, when this training set possesses tiny distances between examples of different categories, the small inter-class distances could constrain the variation of DBs and hence also lead to a tiny algorithm DB variability. We term the drop of algorithm DB variability brought by the small inter-class distances as *constraint effect*.

To investigate how the sample complexity affects the algorithm DB variability through simplicity effect and constraint effect, we manufacture eight *two half moon datasets* with different non-linearity levels to simulate the variation of sample complexities (please see the generalization of the datasets in Appendix A.1). For each two half moon dataset, 20 one-hidden-layer multilayer perceptrons (MLPs) are *standardly* or *adversarially* trained on it, respectively, to estimate the parameter distribution $\mathbb{Q}(\boldsymbol{\theta})$. The related decision boundaries are shown in Figure 1(a) and 1(b) for a more intuitive comparison between DBs with standard and with adversarial training. The data generating distribution $\mathcal{D}$ is simulated by the rectangular grid containing 1000×1000 data points. Then, we can empirically estimate the algorithm DB variability on the two half moon dataset according to Eq. 8;



(a) Fake CIFAR-10 (b) Fake CIFAR-100 (c) Algorithm DB variability vs. test accuracy

Figure 2: Algorithm decision boundary variability on CIFAR-10 and CIFAR-100. (a) Examples of fake CIFAR-10 images generated by conditional BigGAN. (b) Examples of fake CIFAR-100 images generated by conditional BigGAN. (c) Scatter plots of algorithm DB variability to accuracy on test set with different architectures and training strategies on CIFAR-10 and CIFAR-100. The colors of blue, red, and yellow points denote the architectures of VGG-16 (VGG), ResNet-18 (ResN), and WideResNet-28 (WRN), respectively. The shapes of ●, ▲, and ■ designate the training strategies of standard training (S), non-data-augmentation training (N), and adversarial training (A), respectively. Each point is calculated and then averaged on 10 trials.

see Figure 1(c). The plot suggests that (1) the algorithm DB variability increases as the two half moon dataset becomes more complex with the standard training setting; and (2) adversarial training improves the algorithm DB variability on less non-linear datasets, while it lowers the algorithm DB variability when the two half moon dataset becomes more complex, compared to standard training.

We explain these observations through the simplicity effect and constraint effect: when the training data is less complex, simplicity effect is dominant for the algorithm DB variability. Therefore, when the two half moon dataset becomes more complex or more non-linear, the simplicity effect declines, and thus, the algorithm DB variability increases under the standard training setting. As for the adversarial training setting, each data point is converted to a "data ball" with the radius of the adversarial perturbation, which substantially decreases the inter-class distances and makes the training set more complex. Hence, adversarial training undermines the simplicity effect and improves the algorithm DB variability when the two half moon dataset is less non-linear. However, when adversarial training is applied on large non-linear two half moon datasets, their inter-class distances will become extremely tiny. In this scenario, constraint effect becomes an unneglectable factor and will rapidly overwhelm simplicity effect, which causes the drop of algorithm DB variability.

4.2 Algorithm DB variability about model size

In this section, we explore the impact of model size on the algorithm DB variability. To achieve this, we standardly train 1900 one-hidden-layer MLPs on five different non-linear two half moon datasets with different widths (please see the full list of widths involved in our experiments in Appendix A.3.3), while all irrelative variables are strictly controlled. The experiments are repeated for 10 trials on each two half moon dataset. Then the algorithm DB variability is estimated as the same as we do in Section 4.1 in all cases. We employ two plots in Figure 1(d) to present the results due to the different scales on the y-axis. From Figure 1(d), we can observe clear correlations between the algorithm DB variability and the width: *the algorithm DB variability decreases as the layer width grows.* Therefore, albeit more parameters generally represents larger model complexity and hurt the generalization according to canonical learning theory, it could inhibit the variation of decision boundaries. This constraint on the "decision boundary space" can be considered as a manifestation of Occam's razor (Blumer et al., 1987), and hence helps reconcile the mysterious conflict between networks' immense capacity and excellent generalization ability in the over-parameterized regime.

4.3 Algorithm DB variability on real-world data

To further concrete the relationship between the algorithm DB variability and generalization in neural networks, we conduct experiments with different network architectures and training strategies on CIFAR-10 and CIFAR-100. In detail, VGG-16 (Simonyan & Zisserman, 2014), ResNet-18 (He et al., 2016), and WideResNet-28 (Zagoruyko & Komodakis, 2016b) are optimized by standard, non-

data-augmentation and adversarial training, respectively, until the training procedure converges. We clarify that basic data augmentation (crop and flip) (Zagoruyko & Komodakis, 2016a) is adopted in both standard and adversarial training, and only the basic data augmentation is considered in our experiments and analysis. Each training setting (dataset, architecture, training strategy) is repeated for 10 trials with different random seeds to estimate the parameter distribution $\mathbb{Q}(\boldsymbol{\theta})$. As for simulating the data generating distribution, we trained two conditional BigGAN (Zhao et al., 2020), an advanced generative network architecture, to produce $100,000$ fake images for CIFAR-10 and CIFAR-100, respectively. Some examples of fake images have been shown in Figure 2(a) and 2(b).

Through these generative fake images, we can estimate the algorithm DB variability *w.r.t.* these well-trained models. For each training setting, we plot both the average test accuracy and algorithm DB variability; see Figure 2(c). From the plots, we have several observations: (1) adversarial training dramatically decreases the test accuracy and promotes the algorithm DB variability compared to standard training. The vast improvement in the algorithm DB variability brought by AT suggests that *simplicity effect could be dominant in the real scenarios*. This is because that the images have large inter-class distances due to their high dimensions and hence has a looser tolerance to the variation of DBs, which significantly undermines the constraint effect; (2) data augmentation decreases the algorithm decision boundary variability by comparing the standard training and non-data-augmentation training. The reason is that the augmented images by cropping or flipping are still located on the data generating distribution, so data augmentation is considered to expand the size of the training set. Hence, the expanded training set can characterize longer or larger decision boundaries on the data generating distribution; (3) WideResNet has better test accuracy and lower algorithm DB variability than ResNet and VGG; and (4) there is a negative correlation between the test accuracy and the algorithm DB variability. Therefore, based on these findings, we propose the following conjecture:

Hypothesis 1. *Neural networks with smaller algorithm decision boundary variability on the data generating distributions possess better generalization performance.*

4.4 Theoretical Evidence

In this section, we explore and develop the theoretical foundations for the algorithm decision boundary variability on data generating distributions.

Assumption 1. The decision boundary of the classifier network $f_{\boldsymbol{\theta}}$ on data generation distribution $\mathcal{D}$ is a set with measure zero.

Corollary 1. Let $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$. If Assumption 1 holds for all $\boldsymbol{\theta} \sim \mathbb{Q}$, then, for all $i \in \{1, \cdots, k\}$, we have

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{I}(i \in T(f_{\boldsymbol{\theta}}, \mathbf{x}))\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{I}(T(f_{\boldsymbol{\theta}}, \mathbf{x}) = i)\right] \tag{9}$$

and

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{I}(i \notin T(f_{\boldsymbol{\theta}}, \mathbf{x}))\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{I}(T(f_{\boldsymbol{\theta}}, \mathbf{x}) \neq i)\right]. \tag{10}$$

4.4.1 Algorithm DB variability-based lower bounds

Theorem 1 (lower bound on expected risk)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$ and let $\mathbb{Q}(\boldsymbol{\theta})$ be the distribution over $\boldsymbol{\theta}$. Then, if Assumption 1 holds for all $\boldsymbol{\theta} \sim \mathbb{Q}$, we have

$$\mathcal{R}_{\mathcal{D}}(\mathbb{Q}) \geq \frac{AV(f_{\mathbb{Q}}, \mathcal{D})}{2}, \tag{11}$$

where $AV(f_{\mathbb{Q}}, \mathcal{D}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}}, \mathbf{x}) \neq T(f_{\boldsymbol{\theta}'}, \mathbf{x})\right)\right]$ is the algorithm DB variability for $f_{\mathbb{Q}}$ on data generating distribution $\mathcal{D}$.

The proof is given in Appendix C.1. Theorem 1 provides a lower bound on the expected risk $\mathcal{R}_{\mathcal{D}}(\mathbb{Q})$ based on the algorithm DB variability $AV(f_{\mathbb{Q}}, \mathcal{D})$, and the lower bound presents that the Gibbs classifier $f_{\mathbb{Q}}$ possesses a significant expected risk when its algorithm DB variability is large. Moreover, when we consider the binary classification, *i.e.*, $k = 2$, there is a tighter lower bound:

Theorem 2 (lower bound for binary case)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^2$ be a binary classifier network parameterized by $\boldsymbol{\theta}$ and let $\mathbb{Q}(\boldsymbol{\theta})$ be the distribution over $\boldsymbol{\theta}$. Suppose the expected risk $\mathcal{R}_{\mathcal{D}}(\mathbb{Q}) \leq \frac{1}{2}$ and Assumption 1 hold for all $\boldsymbol{\theta} \sim \mathbb{Q}$, then we have

$$\mathcal{R}_{\mathcal{D}}(\mathbb{Q}) \geq \frac{1 - \sqrt{1 - 2AV(f_{\mathbb{Q}}, \mathcal{D})}}{2}. \tag{12}$$

4.4.2 AN ALGORITHM DB VARIABILITY-BASED UPPER BOUND

Before we present the upper bound based on the algorithm DB variability, let us introduce two assumptions used in this section. The first assumption is related to control the covariance between $\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]$ and $\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=j\right)\right]$ for $i,j\neq y$:

Assumption 2. For any $i,j\in\{1,2,\cdots,k\}$ and $i\neq j$, the covariance between the variables $\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\mathbb{I}\left(y\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right]$ and $\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\mathbb{I}\left(y\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right]$ is non-positive:

$$\mathrm{Cov}\left[\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\mathbb{I}\left(y\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right],\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=j\right)\mathbb{I}\left(y\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right]\right]\leq 0. \tag{13}$$

Assumption 2 shows the non-positive correlation between classifying the input example to a wrong class i and classifying it to another wrong class j. Another assumption is related to control the expected risk on the single point:

Assumption 3. For any $(\mathbf{x},y)\sim\mathcal{D}$, it satisfies

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\mathbb{I}\left(y\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right]\leq\frac{2}{k+2}, \tag{14}$$

where $i\in\{1,2,\cdots,k\}$ and k is the number of potential categories.

With the above assumptions, we can derive the upper bound on the variance of risk:

Lemma 1 (upper bound on risk variance)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}):\mathbb{R}^n\rightarrow\mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$ and let $\mathbb{Q}(\boldsymbol{\theta})$ be the distribution over $\boldsymbol{\theta}$. Suppose Assumption 1, 2 and 3 hold, then we have

$$\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right]\right]\leq\frac{AV(f_{\mathbb{Q}},\mathcal{D})}{k}, \tag{15}$$

where $AV(f_{\mathbb{Q}},\mathcal{D})=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\neq T\left(f_{\boldsymbol{\theta}'},\mathbf{x}\right)\right)\right]$ is the algorithm DB variability of $f_{\mathbb{Q}}$ on data generating distribution $\mathcal{D}$.

The proof is given in Appendix C.3. Hence, the next theorem is a direct consequence of the *one-sided Chebyshev inequality*: $\Pr\left[\mathbb{E}[Z_1]-\frac{1}{m}\sum_{i=1}^{m}Z_i>a\right]\leq\frac{\mathrm{Var}[Z_1]}{2ma^2}$ for any $a>0$.

Theorem 3 (PAC upper bound on expected risk)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}):\mathbb{R}^n\rightarrow\mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$ and let $\mathbb{Q}(\boldsymbol{\theta})$ be the distribution over $\boldsymbol{\theta}$ *w.r.t.* the training set $\mathcal{S}$. Suppose Assumption 1, 2 and 3 hold. Then, with the probability of at least $1-\delta$ over a sample of size m, we have

$$\mathcal{R}_{\mathcal{D}}(\mathbb{Q})\leq\mathcal{R}_{S}(\mathbb{Q})+\sqrt{\frac{AV(f_{\mathbb{Q}},\mathcal{D})}{2mk\delta}}. \tag{16}$$

The above theorem presents that a smaller algorithm DB variability $AV(f_{\mathbb{Q}},\mathcal{D})$ contributes to a tighter upper bound on the true risk, which theoretically verifies Hypothesis 1 that neural networks with smaller algorithm decision boundary variability possess better generalization performance.

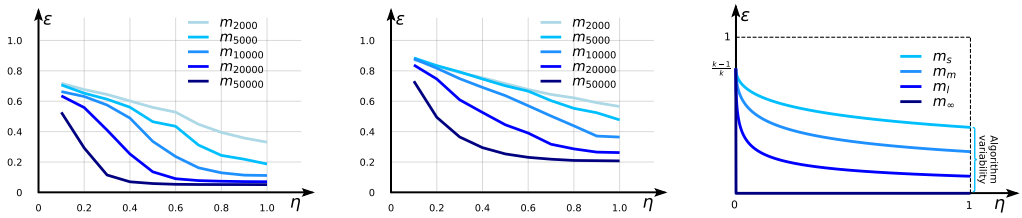
5 DATA DECISION BOUNDARY VARIABILITY

In the previous sections, we introduced the algorithm DB variability, which measures the decision boundary variability caused by the randomness of learning algorithms. However, the algorithm DB variability hardly shows the decision boundary variability caused by changes in training data. To remedy the problem and complete our theorem about decision boundary variability, we define the data DB variability as below:

Definition 5.1 (data decision boundary variability)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}):\mathbb{R}^n\rightarrow\mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$, where $\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S})$ is returned by leveraging the stochastic learning algorithm $\mathcal{A}$ on the training set $\mathcal{S}$, which is sampled from the data generating distribution $\mathcal{D}$. We term $\mathcal{S}_\eta\subset\mathcal{S}$ as a *η-subset* of $\mathcal{S}$ if $\frac{|\mathcal{S}_\eta|}{|\mathcal{S}|}=\eta$. Then, if we fixed η and

$$\inf_{\mathcal{S}_\eta\subset\mathcal{S}}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}},\mathbf{x})\neq T(f_{\boldsymbol{\theta}'},\mathbf{x})\right)\right]=\epsilon, \tag{17}$$

the decision boundary of $f_{\mathcal{A}(\mathcal{S})}$ is said to possess a (ϵ,η)*-data decision boundary variability*.



(a) η-ϵ curves on CIFAR-10 (b) η-ϵ curves on CIFAR-100 (c) Schematic diagram

Figure 3: (a) The η-ϵ curves on CIFAR-10 with different training sample sizes 2000 (m_{2000}), 2000 (m_{2000}), 10000 (m_{10000}), 20000 (m_{20000}), and 50000 (m_{50000}), respectively. (b) The η-ϵ curves on CIFAR-100 with different training sample sizes. (c) The schematic diagram of complexity curves w.r.t. small (m_s), medium (m_m), large (m_l), and infinite (m_∞) sample sizes, respectively.

The data decision boundary variability contains two parameters of ϵ and η, respectively. That Gibbs classifier $f_{\mathcal{A}(\mathcal{S})}$ has a (ϵ, η)-data DB variability means that only the proportion of η of $\mathcal{S}$, *i.e.*, $\mathcal{S}_\eta$, which can be considered as "support vector set", is enough to reconstruct a similar decision boundary with the reconstruction error ϵ. The data DB variability can also be connected with the complexity of decision boundaries if we assume that simpler decision boundaries rely on fewer "support vectors"; we provide a detailed discussion in Appendix B.2 due to space limitation.

5.1 η-ϵ CURVES ABOUT DATA DB VARIABILITY

According to Definition 5.1, the data DB variability degrades to the algorithm DB variability $AV(f_{\mathbb{Q}}, \mathcal{D})$ when $\mathcal{S}_\eta = \mathcal{S}$. In other words, the algorithm DB variability is a special case of the data DB variability with $\eta = 1$. Therefore, the data DB variability could present more detailed information on reflecting how the decision boundary variability depends on the training set, especially when we observe the variation of the reconstruction error ϵ *w.r.t.* different η.

To explore the relationship between the reconstruction error ϵ and the proportion of subset η, we train 1000 networks of ResNet-18 on CIFAR-10 and CIFAR-100 of different sample sizes m, respectively. Albeit finding the most suitable η-subset is intractable, we adopt a coreset selection approach named *selection via proxy* (Coleman et al., 2020), which can rank the importance of training examples, to estimate the η-subset for a given training set $\mathcal{S}$ and proportion η (full list of η involved in our experiments is presented in Appendix A.3.5). Then, through repeatedly training the network on $\mathcal{S}_\eta$, we can estimate the reconstruction error ϵ. The η-ϵ curves of CIFAR-10 and CIFAR-100 are presented in Figure 3(a) and 3(b), respectively. From the plots, we have an observation that *there is a more rapid decline in ϵ along with small η* and also a smaller algorithm DB variability when the training sample size m is larger. Furthermore, we plot the schematic diagram of η-ϵ curves *w.r.t.* different sample size m, as shown in Figure 3(c). When $\eta = 0$, $f_{\mathcal{A}(\mathcal{S}_n)}$ cannot be better than random guess, and hence $\epsilon = \frac{k-1}{k}$, where k is the number of potential categories. We have noticed that ϵ has a sharper drop along with η when the sample size m is larger. Therefore, we rationally propose the following assumption, which is also shown by the right angle with m_∞ in Figure 3(c).

Assumption 4. If $m \to \infty$, then we have $\epsilon \to 0$ when $\eta \to 0$.

These plots indicate that the area under the η-ϵ curve could be a more meticulous predictor for the generalization ability of neural networks compared to the algorithm DB variability, which is only a point on the η-ϵ curve when $\eta = 1$. Hence, the area under the η-ϵ curve can also be considered as an extension of the algorithm DB variability: if the Gibbs classifier $f_{\mathcal{A}(\mathcal{S})}$ possesses a smaller area under the η-ϵ curve, it produces more stable decision boundaries with varying training subsets. In the following section, we will theoretically establish the connection between the parameters of η and ϵ and the generalization ability of neural networks.

5.2 THEORETICAL EVIDENCE

In this section, we develop the theoretical foundations for the data decision boundary variability, and present that *neural networks with better data DB variability possess better generalization*.

Lemma 2. If the decision boundaries of $f_{\mathcal{A}(\mathcal{S})}$ possess a (ϵ, η)-data DB variability, then we have

$$|\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S})) - \mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}_\eta))| \leq \epsilon. \tag{18}$$

Lemma 2 shows that the difference between the expected risk of $\mathcal{A}(\mathcal{S})$ and $\mathcal{A}(\mathcal{S}_\eta)$ can be bounded by their difference in decision boundaries. Then, by applying the concentration inequality, we can bound the difference between their empirical risk on the same training set $\mathcal{S}$ with the probability $1-\delta$:

Lemma 3. If the decision boundaries of $f_{\mathcal{A}(\mathcal{S})}$ possess a (ϵ, η)-data DB variability, then, with probability of at least $1-\delta$ over a sample of size m, we have

$$\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}_\eta)) \leq \mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S})) + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}} + \epsilon, \tag{19}$$

where $m = |\mathcal{S}|$ is the training sample size. If we further assume the training error $\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S})) = 0$, then, with the probability of at least $1-\delta$ over a sample of size m, we have

$$\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}_\eta)) \leq \sqrt{\frac{1}{2m}\log\frac{1}{\delta}} + \epsilon \tag{20}$$

Theorem 4 (data DB variability-based upper bound on expected risk)**.** If the decision boundaries of $f_{\mathcal{A}(\mathcal{S})}$ possess a (ϵ, η)-data DB variability on the data generating distribution $\mathcal{D}$, and assume $\eta \leq 0.5$ and the empirical risks $\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S})) = \mathcal{R}_{\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) = 0$, then, with the probability of at least $1-\delta$ over a sample of size m, we have

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S})) \leq \Omega + \sqrt{4\Omega\Delta} + 8\Delta + \epsilon, \tag{21}$$

where

$$\Omega = \frac{1}{1-\eta}\left(\sqrt{\frac{1}{2m}\log\frac{2}{\delta}} + \epsilon\right), \tag{22}$$

$$\Delta = \eta\log\frac{e}{\eta} + \frac{1}{m}\log\frac{2}{\delta}. \tag{23}$$

Moreover, for sufficient large m, we have

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S})) \leq \mathcal{O}(\frac{1}{\sqrt{m}} + \epsilon + \eta\log\frac{1}{\eta}). \tag{24}$$

The proof is omitted here and is given in Appendix C.7. According to Assumption 4, when $m \to \infty$, then $\eta \to 0$ and $\epsilon \to 0$, and according to Eq. 24, $\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S})) \to 0$. Therefore, the generalization bound is asymptotically converged. Theorem 4 presents that smaller data DB variability, *i.e.*, smaller ϵ and η, contributes to a tighter upper bound on the expected risk, and also theoretically verifies the relationship between the data DB variability and the generalization ability of neural networks.

6 CONCLUSION AND DISCUSSION

In this paper, we empirically and theoretically explored the relationship between decision boundary variability and generalization in neural networks, through the algorithm DB variability and data DB variability. Albeit the algorithm DB variability is easier to estimate, the data DB variability could be a more elaborate predictor on the generalization ability through the area under the η-ϵ curve. As for the theoretical results, compared to the algorithm DB variability upper bound, the data DB variability bound does not depend on the number of classes k. Besides, the analysis in Section 5.1 mentioned that the algorithm DB variability is a special case of the data DB variability. Hence, unifying the theoretical works about these variabilities will be a promising direction in future works.

REFERENCES

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pp. 1305–1338. PMLR, 2020.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJg2b0VYDr`.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Shuyue Guan and Murray Loew. Analysis of generalizability of deep neural networks based on the complexity of decision boundary. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 101–106. IEEE, 2020.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234. PMLR, 2016.

Fengxiang He, Shiye Lei, Jianmin Ji, and Dacheng Tao. Neural networks behave as hash encoders: An empirical study. *arXiv preprint arXiv:2101.05490*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Warren He, Bo Li, and Dawn Song. Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*, 2018.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018a.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018b.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*, 2021.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. *Advances in Neural Information Processing Systems*, 32:3496–3506, 2019.

Minguk Kang and Jaesik Park. ContraGAN: Contrastive Learning for Conditional Image Generation. 2020.

Hamid Karimi, Tyler Derr, and Jiliang Tang. Characterizing the decision boundary of deep neural networks. *arXiv preprint arXiv:1912.11460*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020.

Guillermo Ortiz-Jimenez, Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Hold me tight! influence of discriminative features on deep network boundaries. *arXiv preprint arXiv:2002.06349*, 2020.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Tomaso Poggio, Qianli Liao, and Andrzej Banburski. Complexity control by gradient descent in deep networks. *Nature communications*, 11(1):1–5, 2020.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19 (1):2822–2878, 2018.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 631–648, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the vc-dimension of a learning machine. *Neural computation*, 6(5):851–876, 1994.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Tengyu Xu, Yi Zhou, Kaiyi Ji, and Yingbin Liang. When will gradient methods converge to max-margin classifier under relu models? *Stat*, 10(1):e354, 2021.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016a.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016b.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020.

A ADDITIONAL EXPERIMENTS DETAILS

This section provides all the additional details of our experiments.

A.1 DATASETS

Our experiments are conducted on two public datasets (CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009)) and three manufactured datasets (two half moon datasets, fake CIFAR-10, and fake CIFAR-100). The detail of these datasets are shown as follows.

- **CIFAR-10** consists of $50,000$ training images and $10,000$ test images from 10 different classes, and **CIFAR-100** consists of $50,000$ training images and $10,000$ test images from 100 different classes. One can download CIFAR-10 and CIFAR-100 from `https://www.cs.toronto.edu/~kriz/cifar.html`.
- **Two half moon dataset** consists of $5,000$ training points from two different classes. The two half moon dataset is generated through Scikit-learn module (Pedregosa et al., 2011), and the related Python codes is presented as below:

```
from sklearn import datasets
x, y = datasets.make_moons(n_samples=5000, shuffle=False, noise
    =0.1, random_state=9)
x[2500:, 1] = x[y==1][:,1] - (0.9 - nonlinearity)
```

  `x` and `y` in the above code block denote input examples and labels, respectively. The variable `nonlinearity` is used to set the non-linearity level of the generative two half moon dataset. In this paper, we generated eight two half moon dataset with different non-linearity levels $[0.1, 0.2, 0.3,$
  $0.4, 0.5, 0.6, 0.7, 0.8]$.
- **Fake CIFAR-10** consists of $100,000$ test images from 10 different classes. The fake CIFAR-10 is generated by conditional BigGAN (Zhao et al., 2020). The pre-trained BigGAN is provided by Kang & Park (2020) and can be download from `https://drive.google.com/drive/folders/1xVN7dQPWMLi8gDZEb5FThkjbFtIdzb6b`.
- **Fake CIFAR-100** consists of $100,000$ test images from 100 different classes. The generation of fake CIFAR-100 is similar to it of fake CIFAR-10, while the training set for the BigGAN is replaced from CIFAR-10 to CIFAR-100.

A.2 MODEL ARCHITECTURES

We use different neural network architectures in our experiments, including MLPs, VGG-16, ResNet-18, and WideResNet-28. The detailed architectures of MLPs have been described in the main text. Other architectures are presented in Table 1.

A.3 IMPLEMENTATION DETAILS

This section provides all the additional implementation details for our experiments.

A.3.1 MODEL TRAINING

We employ SGD to optimize all the models and the momentum factor is 0.9. The weight decay factor is set to 5e-4, and the learning rate is decayed by 0.2 every 50 epochs.

A.3.2 ADDITIONAL DETAILS IN SECTION 4.1

We train one-hidden-layer MLPs with width of 1000 on eight two half moon datasets with different non-linearities of $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$. For each two half moon dataset, 20 MLPs are independently trained for 100 epochs, in which the batch size is set as 64, and the learning rate is initialized as 0.01. For the adversarial training, the radius of the adversarial perturbation is set as 0.2 and l_2 distance is selected, while all other settings are the same as the standard training.

Table 1: Detailed model architectures for CIFAR-10/100

VGG-16	ResNet-18	WideResNet-28-10
$(3\times3,32)\times2$ maxpool, 2×2	3×3, 64	3×3, 16
$(3\times3,128)\times2$ maxpool, 2×2	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,160\\3\times3,160\end{bmatrix}\times4$
$(3\times3,256)\times3$ maxpool, 2×2	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,320\\3\times3,320\end{bmatrix}\times4$
$(3\times3,512)\times3$ maxpool, 2×2	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,640\\3\times3,640\end{bmatrix}\times4$
$(3\times3,512)\times3$ maxpool, 2×2	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$	
fc-4096 fc-4096	avgpool	avgpool
fc-10/100	fc-10/100	fc-10/100

To estimate the consistency of decision boundary, we sample 1000×1000 points from the 2-dimensional grid of $[-1.5,2.5]\times[-1.5,1.5]$.

A.3.3 Additional details in Section 4.2

We train one-hidden-layer MLPs with widths of $[10,20,30,40,50,60,70,80,90,100,200,300,400,500,600,700,800,900,1000]$ on five two half moon datasets with different non-linearity levels of $[0.3,0.4,0.5,0.6,0.7]$. 20 MLPs are independently trained for each (width, non-linearity) setting.

A.3.4 Additional details in Section 4.3

We train VGG-16, ResNet-18, Wide-ResNet-28 on CIFAR-10 and CIFAR-100. In the training procedure, the model is trained for 200 epochs, in which the batch size is set to 128, and the learning rate is initialized as 0.1. There are three training strategies included in this experiment: standard training, non-data-augmentation training, and adversarial training. In the adversarial training, the radius of the adversarial perturbation is set as 10 and l_∞ distance is selected. The basic data augmentation (cropping and flipping) in the standard training and adversarial training is achieved by the following Pytorch code:

```
transforms.RandomCrop(32, padding=4)
transforms.RandomHorizontalFlip()
```

The experiment is repeated for 10 trials for each (dataset, architecture, training strategy) setting.

A.3.5 Additional details in Section 5.1

We randomly sample examples from the training set of CIFAR-10 and CIFAR-100 to form five datasets with different sizes of $[2000,5000,10000,20000,50000]$, respectively. For each dataset, we obtain 10 η-subsets with different η of $[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]$ via a coreset selection approach named *selection via proxy* (Coleman et al., 2020). The related code can be downloaded from `https://github.com/stanford-futuredata/selection-via-proxy`. The ResNet-18 is repeatedly trained for 10 trials to estimate the complexity of decision boundaries for each η-subset.

B ADDITIONAL DISCUSSION ON DECISION BOUNDARY VARIABILITY

B.1 ALGORITHM DB VARIABILITY AND THE ENTROPY OF DECISION BOUNDARIES

If Assumption 1 holds for all $\boldsymbol{\theta} \sim \mathbb{Q}$, $1 - AV(f_{\mathbb{Q}}, \mathcal{D})$ can be rewritten as

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right]=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\sum_{i=1}^{k}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]. \tag{25}$$

The term $\sum_{i=1}^{k}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i)\right]$ can be considered to measure the degree of prediction uncertainty for the given voxel $\mathbf{x}$ in the input space $\mathbb{R}^n$. If we leverage $-\log(\cdot)$ on the term $\sum_{i=1}^{k}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i)\right]$, then $-\log\sum_{i=1}^{k}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i)\right]$ denotes the collision entropy of prediction made by the Gibbs classifier $f_{\mathbb{Q}}$ on $\mathbf{x}$. We can also replace the collision entropy with canonical Shannon entropy $-\sum_{i=1}^{k}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i)\right]\log\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i)\right]$ in the future research. Hence, the algorithm DB variability is closely related to the "entropy of decision boundary", and the uncanny generalization in neural networks might be further uncovered by investigating this low entropy of decision boundary.

B.2 DATA DB VARIABILITY AND THE COMPLEXITY OF DBS

According to Guan & Loew (2020), a complex decision boundary is considered to own large curvatures and conjectured to indicate inferior generalization. Nevertheless, from the perspective of causality, we argue that the large curvature or non-linearity of DBs *is the result other than the cause* for the classification task, and the primary reason for shaping a complex DB during the training procedure should be the significant non-linearity of the training data. If only the geometric properties of decision boundaries are analysed without investigating the data, the results might be incomplete and even misleading. Another obstacle for describing the complexity of DBs via its geometric properties is the huge dimensional input space, which makes the geometric properties of DBs hard to quantify and estimate. Therefore, defining the complexity of DBs based on its curvature is not rational and impractical.

Here we consider the complexity of DBs from the perspective of the training set. During the training procedure, a small part of training examples, considered as "support vectors", play a more critical role in supervising the formation of decision boundaries and compelling the DB to be gradually more complicated. If the construction of decision boundaries relies on fewer "support vectors", the decision boundary should be simpler. In other words, if these "support vectors" are excluded from the training sample, the DB will be notably dissimilar when the network is retrained on the modified training set. Hence, the complexity of DBs can be also defined with the notion of the data DB variability:

Definition B.1 (complexity of decision boundaries)**.** Let $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$ be a classifier network parameterized by $\boldsymbol{\theta}$, where $\boldsymbol{\theta} \sim \mathcal{A}(\mathcal{S})$ is returned by leveraging the stochastic learning algorithm $\mathcal{A}$ on the training set $\mathcal{S}$, which is sampled from the data generating distribution $\mathcal{D}$. We term $\mathcal{S}_\eta \subset \mathcal{S}$ as a η-*subset* of $\mathcal{S}$ if $\frac{|\mathcal{S}_\eta|}{|\mathcal{S}|} = \eta$. Then, if we fixed η and

$$\inf_{\mathcal{S}_\eta\subset\mathcal{S}}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}},\mathbf{x})\neq T(f_{\boldsymbol{\theta}'},\mathbf{x})\right)\right]=\epsilon, \tag{26}$$

the decision boundary of $f_{\mathcal{A}(\mathcal{S})}$ is said to possess a (ϵ, η)-*complexity*.

Implications about the complexity of DB. With Definition B.1, we could more intuitively understand the relationship between the complexity of DBs and the generalization ability in deep learning: (1) difficult tasks generally have more complex decision boundaries, since their datum are more non-linear and contain more "support vectors"; (2) in adversarial training, each data point is converted to a "data ball" with the radius of the adversarial perturbation and has more impact on forming the DBs. Hence, adversarial training contributes to a more complex decision boundary by enlarging the "support vector set"; (3) for data augmentation, generated images are also considered to obey the data generation distribution $\mathcal{D}$. Hence, data augmentation decreases the complexity of decision boundaries by greatly expanding the training set $\mathcal{S}$, while $|\mathcal{S}_\eta|$ has only a slight growth. Besides, according to our theoretical results in Section 5.2, it can be verified that *neural networks with simpler decision boundaries possess better generalization performance*.

C PROOFS

To avoid technicalities, the measurability/integrability issues are ignored throughout this paper. Moreover, Fubini's theorem is assumed to be applicable for any integration *w.r.t.* multiple variables. In other words, the order of integrations is exchangeable.

C.1 PROOF OF THEOREM 1

Proof. If Assumption 1 holds for all $\boldsymbol{\theta} \sim \mathbb{Q}$, we have

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}'}, \mathbf{x}\right)\right)\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) \neq T\left(f_{\boldsymbol{\theta}'}, \mathbf{x}\right)\right)\right] = 0 \tag{27}$$

Hence,

$$\begin{aligned}
AV(f_{\mathbb{Q}}, \mathcal{D}) =& \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}}, \mathbf{x}) \neq T(f_{\boldsymbol{\theta}'}, \mathbf{x})\right)\right] && (28)\\
=& \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T(f_{\boldsymbol{\theta}}, \mathbf{x})\right)\mathbb{I}\left(y \notin T(f_{\boldsymbol{\theta}'}, \mathbf{x})\right)\right. && (29)\\
&\left. + \mathbb{I}\left(y \notin T(f_{\boldsymbol{\theta}}, \mathbf{x})\right)\mathbb{I}\left(T(f_{\boldsymbol{\theta}}, \mathbf{x}) \neq T(f_{\boldsymbol{\theta}'}, \mathbf{x})\right)\right] && (30)\\
\leq& 2\mathcal{R}_{\mathcal{D}}(\mathbb{Q}) && (31)
\end{aligned}$$

The proof is completed. □

C.2 PROOF OF THEOREM 2

Proof. When the classification is binary, *i.e.*, $k = 2$, with Assumption 1, we have

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) = T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] && (32)\\
=& \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] + \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] && (33)\\
=& \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] + \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[1 - \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right]\right]^2 && (34)\\
=& 2\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] + 2\mathcal{R}_{\mathcal{D}}(\mathbb{Q}) - 1 && (35)
\end{aligned}$$

Plugging in $\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right]\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] - (1 - \mathcal{R}_{\mathcal{D}}(\mathbb{Q}))^2$ yields

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) = T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right] && (36)\\
=& 2\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \in T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right]\right] + \mathcal{R}^2_{\mathcal{D}}(\mathbb{Q}) + (1 - \mathcal{R}_{\mathcal{D}}(\mathbb{Q}))^2 && (37)\\
\geq& \mathcal{R}^2_{\mathcal{D}}(\mathbb{Q}) + (1 - \mathcal{R}_{\mathcal{D}}(\mathbb{Q}))^2 && (38)
\end{aligned}$$

Plugging in $AV(f_{\mathbb{Q}}, \mathcal{D}) = 1 - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}}, \mathbf{x}) = T(f_{\boldsymbol{\theta}'}, \mathbf{x})\right)\right]$ and solving the inequality yield the desired inequality and finishes the proof of Theorem 2. □

C.3 PROOF OF LEMMA 1

Proof. With Assumption 1, the variance of $\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y \notin T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right)\right)\right]$ can be written as

$$\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) \neq y\right)\right]\right] = \mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\sum_{i=1}^{k}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) = i\right)\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) \neq y\right)\right]\right] \tag{39}$$

Without loss of generality, we assume that all examples are belong to the first class, *i.e.*, $y = 1$, which doesn't change the variance. Hence, we have

$$\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\sum_{i=1}^{k}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) = i\right)\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) \neq y\right)\right]\right] = \mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\sum_{i=2}^{k}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}}, \mathbf{x}\right) = i\right)\right]\right] \tag{40}$$

With the non-positive covariantion assumption (Assumption 2), the following inequality can be derived:

$$\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\sum_{i=2}^{k}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right] \tag{41}$$

$$\leq\sum_{i=2}^{k}\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left(\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right)$$

$$=\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]-\sum_{i=2}^{k}\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right)^2 \tag{42}$$

For $1-AV(f_{\mathbb{Q}},\mathcal{D})=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=T\left(f_{\boldsymbol{\theta}'},\mathbf{x}\right)\right)\right]$, we have

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=T\left(f_{\boldsymbol{\theta}'},\mathbf{x}\right)\right)\right] \tag{43}$$

$$=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\sum_{i=1}^{k}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right] \tag{44}$$

$$=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=y\right)\right]+\sum_{i=2}^{k}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right] \tag{45}$$

$$=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\left(1-\sum_{i=2}^{k}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right)^2+\sum_{i=2}^{k}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right] \tag{46}$$

$$\leq 1+2\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]-2\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]$$
$$+2\sum_{i>j>1}\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=j\right)\right]\right) \tag{47}$$

$$\leq 1+2\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]-2\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]$$
$$+(k-2)\sum_{i=2}^{k}\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right)^2 \tag{48}$$

$$\leq 1+2\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]-2\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]$$
$$+k\sum_{i=2}^{k}\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right)^2 \tag{49}$$

The last inequality holds because $2p_ip_j\leq p_i^2+p_j^2$. Moreover, with Assumption 3 that $\forall(\mathbf{x},y)\sim\mathcal{D}$ and $i\in[k]$, $\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(f_{\boldsymbol{\theta}}(\mathbf{x})=i\right)\mathbb{I}\left(f_{\boldsymbol{\theta}}(\mathbf{x})\neq y\right)\right]\leq\frac{2}{k+2}$, we have

$$\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]-\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right] \tag{50}$$

$$\leq-\frac{k}{2}\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right] \tag{51}$$

Plugging the inequality into Eq. 49 yields

$$\sum_{i=2}^{k}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}^2_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]-\sum_{i=2}^{k}\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)=i\right)\right]\right)^2\leq\frac{AV(f_{\mathbb{Q}},\mathcal{D})}{k}. \tag{52}$$

By combining the above equality and Eq. 42, we get the following inequality *w.r.t.* the variance:

$$\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\neq y\right)\right]\right]\leq\frac{AV(f_{\mathbb{Q}},\mathcal{D})}{k}. \tag{53}$$

The proof is completed. □

C.4 Proof of Theorem 3

Proof. For the random variable Z_i, recall the *one-sided* Chebyshev inequality that

$$\Pr\left[\mathbb{E}[Z_1]-\frac{1}{m}\sum_{i=1}^{m}Z_i>a\right]\leq\frac{\mathrm{Var}[Z_1]}{2ma^2}. \tag{54}$$

Let $Z_i=\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y_i\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}_i\right)\right)\right]$, so we have

$$\Pr\left[\mathcal{R}_{\mathcal{D}}(\mathbb{Q})-\mathcal{R}_S(\mathbb{Q})>a\right]\leq\frac{\mathrm{Var}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{Q}}\left[\mathbb{I}\left(y\notin T\left(f_{\boldsymbol{\theta}},\mathbf{x}\right)\right)\right]\right]}{2ma^2}. \tag{55}$$

Plugging Lemma 1 into the above inequality yields

$$\Pr\left[\mathcal{R}_{\mathcal{D}}(\mathbb{Q})-\mathcal{R}_S(\mathbb{Q})>a\right]\leq\frac{AV(f_{\mathbb{Q}},\mathcal{D})}{2mka^2}. \tag{56}$$

Plugging in $\delta=\frac{AV(f_{\mathbb{Q}},\mathcal{D})}{2mka^2}$ yields the desired inequality and completes the proof. □

C.5 Proof of Lemma 2

Proof. From the definition of the complexity of DB, there exists a η-subset $\mathcal{S}_\eta$ s.t.

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}},\mathbf{x})\neq T(f_{\boldsymbol{\theta}'},\mathbf{x})\right)\right]=\epsilon. \tag{57}$$

Recall the LHS $=|\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}))-\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}_\eta))|$, then

$$\begin{aligned}
\text{LHS}&=\left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S})}\left[\mathbb{I}\left(y\notin T(f_{\boldsymbol{\theta}},\mathbf{x})\right)\right]-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(y\notin T(f_{\boldsymbol{\theta}},\mathbf{x})\right)\right]\right| && (58)\\
&=\left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(y\notin T(f_{\boldsymbol{\theta}},\mathbf{x})\right)-\mathbb{I}\left(y\notin T(f_{\boldsymbol{\theta}'},\mathbf{x})\right)\right]\right| && (59)\\
&\leq\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\left|\mathbb{I}\left(y\notin T(f_{\boldsymbol{\theta}},\mathbf{x})\right)-\mathbb{I}\left(y\notin T(f_{\boldsymbol{\theta}'},\mathbf{x})\right)\right|\right] && (60)\\
&\leq\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}},\mathbf{x})\neq T(f_{\boldsymbol{\theta}'},\mathbf{x})\right)\right]=\epsilon && (61)
\end{aligned}$$

The proof is completed. □

C.6 Proof of Lemma 3

Proof. Let

$$\mu=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}},\mathbf{x})\neq T(f_{\boldsymbol{\theta}'},\mathbf{x})\right)\right] \tag{62}$$

$$\hat{\mu}=\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{A}(\mathcal{S}),\boldsymbol{\theta}'\sim\mathcal{A}(\mathcal{S}_\eta)}\left[\mathbb{I}\left(T(f_{\boldsymbol{\theta}},\mathbf{x}_i)\neq T(f_{\boldsymbol{\theta}'},\mathbf{x}_i)\right)\right], \tag{63}$$

By applying Hoeffding's Inequality, we have

$$\Pr\left[\hat{\mu}-\mu\geq t\right]\leq\exp(-2mt^2). \tag{64}$$

Plug in $\delta=\exp(-2mt^2)$ into Eq. 64, thus, with the probability of at least $1-\delta$, we have

$$\hat{\mu}\leq\sqrt{\frac{1}{2m}\log\frac{1}{\delta}}+\mu\leq\sqrt{\frac{1}{2m}\log\frac{1}{\delta}}+\epsilon. \tag{65}$$

Following the proof of Lemma 2, we can derive the inequality of $|\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}_\eta))-\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}))|\leq\hat{\mu}$. Plugging this into Eq. 65 yields the desired inequality and concludes the proof of Lemma 3. □

C.7 PROOF OF THEOREM 4

We first introduce Lemma 4 and Lemma 5 as below.

Lemma 4 (Lemma 30.1 in (Shalev-Shwartz & Ben-David, 2014))**.** Assume T and V are two datasets independently sampled from the data generating distribution $\mathcal{D}$, then, with the probability of at least $1-\delta$, we have

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(T)) \leq \mathcal{R}_V(\mathcal{A}(T)) + \sqrt{\frac{2\mathcal{R}_V(\mathcal{A}(T))\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}. \tag{66}$$

Lemma 5 (Theorem 30.2 in (Shalev-Shwartz & Ben-David, 2014))**.** Let $\mathcal{S}_\eta$ be a η-subnet of the dataset $\mathcal{S}$, which is sampled from the data generation distribution $\mathcal{D}$ and the sample size $|\mathcal{S}| = m$. Let $\mathcal{S}\backslash\mathcal{S}_\eta = \mathcal{S} - \mathcal{S}_\eta$ and assume $\eta \leq 0.5$. Then, with the probability of at least $1-\delta$ over a sample of size m, we have

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}_\eta)) \leq \mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) + \sqrt{4\mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta))\Delta} + 8\Delta, \tag{67}$$

where

$$\Delta = \eta\log\frac{e}{\eta} + \frac{1}{m}\log\frac{1}{\delta'}. \tag{68}$$

Proof of Lemma 5.

$$\Pr\left[\exists\mathcal{S}_\eta \subseteq \mathcal{S} \text{ s.t. } \mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}_\eta)) \leq \mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) + \sqrt{\frac{2\mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta))\log(1/\delta)}{|\mathcal{S}\backslash\mathcal{S}_\eta|}} + \frac{4\log(1/\delta)}{|\mathcal{S}\backslash\mathcal{S}_\eta|}\right] \tag{69}$$

$$\leq \sum_{\mathcal{S}_\eta\subseteq\mathcal{S}} \Pr\left[\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}_\eta)) \leq \mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) + \sqrt{\frac{2\mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta))\log(1/\delta)}{|\mathcal{S}\backslash\mathcal{S}_\eta|}} + \frac{4\log(1/\delta)}{|\mathcal{S}\backslash\mathcal{S}_\eta|}\right] \tag{70}$$

$$= \binom{m}{\eta m}\delta \leq \left(\frac{e}{\eta}\right)^{\eta m}\delta \tag{71}$$

Plug in $\delta' = \left(\frac{e}{\eta}\right)^{\eta m}\delta$, and use the assumption $\eta \leq \frac{1}{2}$, which implies $|\mathcal{S}\backslash\mathcal{S}_\eta| \geq \frac{m}{2}$, then, with the probability of at least $1-\delta'$, we have that

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}_\eta)) \leq \mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) + \sqrt{4\mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta))\left(\eta\log\frac{e}{\eta} + \frac{1}{m}\log\frac{1}{\delta'}\right)} + 8\left(\eta\log\frac{e}{\eta} + \frac{1}{m}\log\frac{1}{\delta'}\right), \tag{72}$$

which concludes the proof. □

With the above lemmas, we can derive the generalization bound based on the complexity of decision boundary.

Proof of Theorem 4. With the assumption of zero-training error that $\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S})) = \mathcal{R}_{\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) = 0$, we can derive the follow inequality:

$$\mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) = \frac{1}{1-\eta}\left(\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}_\eta)) - \mathcal{R}_{\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta))\right) = \frac{1}{1-\eta}\mathcal{R}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}_\eta)) \tag{73}$$

Then, by plugging the above equation into to Lemma 3, with the probability of at least $1-\delta$, we have

$$\mathcal{R}_{\mathcal{S}\backslash\mathcal{S}_\eta}(\mathcal{A}(\mathcal{S}_\eta)) \leq \frac{1}{1-\eta}\left(\sqrt{\frac{1}{2m}\log\frac{1}{\delta}} + \epsilon\right) \tag{74}$$

Through combining this with Lemma 5, with the probability of at least $1-2\delta$, we have

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}_\eta)) \leq \Omega + \sqrt{4\Omega\Delta} + 8\Delta, \tag{75}$$

where

$$\Omega = \frac{1}{1-\eta}\left(\sqrt{\frac{1}{2m}\log\frac{1}{\delta}} + \epsilon\right), \tag{76}$$

$$\Delta = \eta \log\frac{e}{\eta} + \frac{1}{m}\log\frac{1}{\delta}. \tag{77}$$

Plugging the equality of $\mathcal{R}_{\mathcal{D}}\left(\mathcal{A}\left(\mathcal{S}\right)\right) \leq \mathcal{R}_{\mathcal{D}}\left(\mathcal{A}\left(\mathcal{S}_{\eta}\right)\right) + \epsilon$ in Lemma 2 into Eq. 75, with the probability of at least $1 - 2\delta$, we have

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S})) \leq \Omega + \sqrt{4\Omega\Delta} + 8\Delta + \epsilon. \tag{78}$$

Plugging in $\delta' = 2\delta$ yields the desired inequality of Eq. 21.

When m is sufficient large, $\sqrt{4\Omega\Delta}$ can be dropped due to $\sqrt{4\Omega\Delta} \leq \Omega + \Delta$. Considering $\eta \leq 0.5$, $\Omega \leq 2\left(\sqrt{\frac{1}{2m}\log\frac{1}{\delta}} + \epsilon\right) = \mathcal{O}(\frac{1}{\sqrt{m}} + \epsilon)$. The term $\frac{1}{m}\log\frac{1}{\delta}$ in Δ can be dropped because it has a faster convergence speed compared to $\sqrt{\frac{1}{2m}\log\frac{1}{\delta}}$ in Ω. Because $\log\frac{1}{\delta}$ is considered as a constant, we have

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\mathcal{S})) \leq \mathcal{O}(\frac{1}{\sqrt{m}} + \epsilon + \eta\log\frac{1}{\eta}). \tag{79}$$

The proof of Theorem 4 is finished. □