A PAC-Bayesian Analysis of Randomized Learning with Application to Stochastic Gradient Descent

Ben London
blondon@amazon.com
Amazon AI

Abstract

We study the generalization error of randomized learning algorithms—focusing on stochastic gradient descent (SGD)—using a novel combination of PAC-Bayes and algorithmic stability. Importantly, our generalization bounds hold for all posterior distributions on an algorithm's random hyperparameters, including distributions that depend on the training data. This inspires an adaptive sampling algorithm for SGD that optimizes the posterior at runtime. We analyze this algorithm in the context of our generalization bounds and evaluate it on a benchmark dataset. Our experiments demonstrate that adaptive sampling can reduce empirical risk faster than uniform sampling while also improving out-of-sample accuracy.

1 Introduction

Randomized algorithms are the workhorses of modern machine learning. One such algorithm is stochastic gradient descent (SGD), a first-order optimization method that approximates the gradient of the learning objective by a random point estimate, thereby making it efficient for large datasets. Recent interest in studying the generalization properties of SGD has led to several breakthroughs. Notably, Hardt et al. [13] showed that SGD is stable with respect to small perturbations of the training data, which let them bound the risk of a learned model. Related studies followed thereafter [18, 21]. Simultaneously, Lin and Rosasco [20] derived risk bounds that show that early stopping acts as a regularizer in multi-pass SGD (echoing studies of incremental gradient descent [25]).

In this paper, we study generalization in randomized learning, with SGD as a motivating example. Using a novel analysis that combines PAC-Bayes with algorithmic stability (reminiscent of [22]), we prove new generalization bounds for randomized learning algorithms, which apply to SGD under various assumptions on the loss function and optimization objective. Our bounds improve on related work in two important ways. While some previous bounds for SGD [2, 13, 18, 21] hold in expectation over draws of the training data, our bounds hold with high probability. Further, existing generalization bounds for randomized learning [8, 9] only apply to algorithms with fixed distributions (such as SGD with uniform sampling); thanks to our PAC-Bayesian treatment, our bounds hold for all posterior distributions, meaning they support data-dependent randomization. The penalty for overfitting the posterior to the data is captured by the posterior's divergence from a fixed prior.

Our generalization bounds suggest a sampling strategy for SGD that adapts to the training data and model, focusing on useful examples while staying close to a uniform prior. We therefore propose an adaptive sampling algorithm that dynamically updates its distribution using multiplicative weight updates (similar to boosting [10, 27], focused online learning [28] and exponentiated gradient dual coordinate ascent [5]). The algorithm requires minimal tuning and works with any stochastic gradient update rule. We analyze the divergence of the adaptive posterior and conduct experiments on a benchmark dataset, using several combinations of update rule and sampling utility function. Our experiments demonstrate that adaptive sampling can reduce empirical risk faster than uniform sampling while also improving out-of-sample accuracy.

2 Preliminaries

Let $\mathcal{X}$ denote a compact domain; let $\mathcal{Y}$ denote a set of labels; and let $\mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y}$ denote their Cartesian product. We assume there exists an unknown, fixed distribution, $\mathbb{D}$, supported on $\mathcal{Z}$. Given a dataset of *examples*, $S \triangleq (z_1, \ldots, z_n) = ((x_1, y_1), \ldots, (x_n, y_n))$, drawn independently and identically from $\mathbb{D}$, we wish to learn the parameters of a predictive model, $\mathcal{X} \mapsto \mathcal{Y}$, from a class of *hypotheses*, $\mathcal{H}$, which we assume is a subset of Euclidean space. We have access to a deterministic *learning algorithm*, $A : \mathcal{Z}^n \times \Theta \to \mathcal{H}$, which, given S, and some *hyperparameters*, $\theta \in \Theta$, produces a hypothesis, $h \in \mathcal{H}$.

We measure the quality of a hypothesis using a *loss* function, $L : \mathcal{H} \times \mathcal{Z} \to [0, M]$, which we assume is M-bounded[1] and λ-Lipschitz (see Appendix A for the definition). Let $L(A(S, \theta), z)$ denote the loss of a hypothesis that was output by $A(S, \theta)$ when applied to example z. Ultimately, we want the learning algorithm to have low expected loss on a random example; i.e., low *risk*, denoted $R(S, \theta) \triangleq \mathbb{E}_{z \sim \mathbb{D}}[L(A(S, \theta), z)]$. (The learning algorithm should always be clear from context.) Since this expectation cannot be computed, we approximate it by the average loss on the training data; i.e., the *empirical risk*, $\hat{R}(S, \theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} L(A(S, \theta), z_i)$, which is what most learning algorithms attempt to minimize. By bounding the difference of the two, $G(S, \theta) \triangleq R(S, \theta) - \hat{R}(S, \theta)$, which we refer to as the *generalization error*, we obtain an upper bound on $R(S, \theta)$.

Throughout this document, we will view a randomized learning algorithm as a deterministic learning algorithm whose hyperparameters are randomized. For instance, *stochastic gradient descent* (SGD) performs a sequence of hypothesis updates, for $t = 1, \ldots, T$, of the form

$$h_t \leftarrow U_t(h_{t-1}, z_{i_t}) \triangleq h_{t-1} - \eta_t \nabla F(h_{t-1}, z_{i_t}),$$

using a sequence of random example indices, $\theta = (i_1, \ldots, i_T)$, sampled according to a distribution, $\mathbb{P}$, on $\Theta = \{1, \ldots, n\}^T$. The *objective* function, $F : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$, may be different from L; it is usually chosen as an optimizable upper bound on L, and need not be bounded. The parameter η_t is a step size for the update at iteration t. SGD can be viewed as taking a dataset, S, drawing $\theta \sim \mathbb{P}$, then running a deterministic algorithm, $A(S, \theta)$, which executes the sequence of hypothesis updates.

Since learning is randomized, we will deal with the expected loss over draws of random hyperparameters. We therefore overload the above notation for a distribution, $\mathbb{P}$, on the hyperparameter space, Θ; let $R(S, \mathbb{P}) \triangleq \mathbb{E}_{\theta \sim \mathbb{P}}[R(S, \theta)]$, $\hat{R}(S, \mathbb{P}) \triangleq \mathbb{E}_{\theta \sim \mathbb{P}}[\hat{R}(S, \theta)]$, and $G(S, \mathbb{P}) \triangleq R(S, \mathbb{P}) - \hat{R}(S, \mathbb{P})$.

2.1 Relationship to PAC-Bayes

Conditioned on the training data, a *posterior* distribution, $\mathbb{Q}$, on the hyperparameter space, Θ, induces a distribution on the hypothesis space, $\mathcal{H}$. If we ignore the learning algorithm altogether and think of $\mathbb{Q}$ as a distribution on $\mathcal{H}$ directly, then $\mathbb{E}_{h \sim \mathbb{Q}}[L(h, z)]$ is the *Gibbs* loss; that is, the expected loss of a random hypothesis. The Gibbs loss has been studied extensively using *PAC-Bayesian* analysis (also known simply as *PAC-Bayes*) [4, 11, 19, 23, 26]. In the PAC-Bayesian learning framework, we fix a *prior* distribution, $\mathbb{P}$, then receive some training data, $S \sim \mathbb{D}^n$, and learn a posterior distribution, $\mathbb{Q}$. PAC-Bayesian bounds frame the generalization error, $G(S, \mathbb{Q})$, as a function of the posterior's divergence from the prior, which penalizes overfitting the posterior to the training data.

In Section 4, we derive new upper bounds on $G(S, \mathbb{Q})$ using a novel PAC-Bayesian treatment. While traditional PAC-Bayes analyzes distributions directly on $\mathcal{H}$, we instead analyze distributions on Θ. Thus, instead of applying the loss directly to a random hypothesis, we apply it to the output of a learning algorithm, whose inputs are a dataset and a random hyperparameter instantiation. This distinction is subtle, but important. In our framework, a random hypothesis is explicitly a function of the learning algorithm, whereas in traditional PAC-Bayes this dependence may only be implicit—for instance, if the posterior is given by random permutations of a learned hypothesis. The advantage of making the learning aspect explicit is that it isolates the source of randomness, which may help in analyzing the distribution of learned hypotheses. Indeed, it may be difficult to map the output of a randomized learning algorithm to a distribution on the hypothesis space. That said, the disadvantage of making learning explicit is that, due to the learning algorithm's dependence on the training data and hyperparameters, the generalization error could be sensitive to certain examples or hyperparameters. This condition is quantified with *algorithmic stability*, which we discuss next.

---

[1] Accommodating unbounded loss functions is possible [15], but requires additional assumptions.

3 Algorithmic Stability

Informally, algorithmic stability measures the change in loss when the inputs to a learning algorithm are perturbed; a learning algorithm is stable if small perturbations lead to proportional changes in the loss. In other words, a learning algorithm should not be overly sensitive to any single input. Stability is crucial for learnability [29], and has also been linked to differentially private learning [30]. In this section, we discuss several notions of stability tailored for randomized learning algorithms. From this point on, let $D_{\mathrm{H}}(\mathbf{v}, \mathbf{v}') \triangleq \sum_{i=1}^{|\mathbf{v}|} \mathbb{1}\{v_i \neq v_i'\}$ denote the Hamming distance.

3.1 Definitions of Stability

The literature traditionally measures stability with respect to perturbations of the training data. We refer to this general property as *data stability*. Data stability has been defined in many ways. The following definitions, originally proposed by Elisseeff et al. [8], are designed to accommodate randomized algorithms via an expectation over the hyperparameters, $\theta \sim \mathbb{P}$.

Definition 1 (Uniform Stability). A randomized learning algorithm, A, is $\beta_{\mathcal{Z}}$-uniformly stable with respect to a loss function, L, and a distribution, $\mathbb{P}$ on Θ, if

$$\sup_{S,S' \in \mathcal{Z}^n : D_{\mathrm{H}}(S,S')=1} \sup_{z \in \mathcal{Z}} \left| \mathbb{E}_{\theta \sim \mathbb{P}} \left[L(A(S,\theta), z) - L(A(S',\theta), z) \right] \right| \leq \beta_{\mathcal{Z}}.$$

Definition 2 (Pointwise Hypothesis Stability). For a given dataset, S, let $S^{i,z}$ denote the result of replacing the ith example with example z. A randomized learning algorithm, A, is $\beta_{\mathcal{Z}}$-pointwise hypothesis stable with respect to a loss function, L, and a distribution, $\mathbb{P}$ on Θ, if

$$\sup_{i \in \{1,\ldots,n\}} \mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{z \sim \mathbb{D}} \mathbb{E}_{\theta \sim \mathbb{P}} \left[\left| L(A(S,\theta), z_i) - L(A(S^{i,z}, \theta), z_i) \right| \right] \leq \beta_{\mathcal{Z}}.$$

Uniform stability measures the maximum change in loss from replacing any single training example, whereas pointwise hypothesis stability measures the expected change in loss on a random example when said example is removed from the training data. Under certain conditions, $\beta_{\mathcal{Z}}$-uniform stability implies $\beta_{\mathcal{Z}}$-pointwise hypothesis stability, but not vice versa. Thus, while uniform stability enables sharper bounds, pointwise hypothesis stability supports a wider range of learning algorithms.

In addition to data stability, we might also require stability with respect to changes in the hyperparameters. From this point forward, we will assume that the hyperparameter space, Θ, decomposes into the product of T subspaces, $\prod_{t=1}^{T} \Theta_t$. For instance, Θ could be the set of all sequences of example indices, $\{1,\ldots,n\}^T$, such as one would sample from in SGD.

Definition 3 (Hyperparameter Stability). A randomized learning algorithm, A, is β_{Θ}-uniformly stable with respect to a loss function, L, if

$$\sup_{S \in \mathcal{Z}^n} \sup_{z \in \mathcal{Z}} \sup_{\theta,\theta' \in \Theta : D_{\mathrm{H}}(\theta,\theta')=1} |L(A(S,\theta), z) - L(A(S,\theta'), z)| \leq \beta_{\Theta}.$$

When A is both $\beta_{\mathcal{Z}}$-uniformly and β_{Θ}-uniformly stable, we say that A is $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniformly stable.

Remark 1. For SGD, Definition 3 can be mapped to Bousquet and Elisseeff's [3] original definition of uniform stability using the resampled example sequence. Yet their generalization bounds would still not apply because the resampled data is not i.i.d. and SGD is not a symmetric learning algorithm.

3.2 Stability of Stochastic Gradient Descent

For non-vacuous generalization bounds, we will need the data stability coefficient, $\beta_{\mathcal{Z}}$, to be of order $\tilde{\mathrm{O}}(n^{-1})$. Additionally, certain results will require the hyperparameter stability coefficient, β_{Θ}, to be of order $\tilde{\mathrm{O}}(1/\sqrt{nT})$. (If $T = \Theta(n)$, as it often is, then $\beta_{\Theta} = \tilde{\mathrm{O}}(T^{-1})$ suffices.) In this section, we review some conditions under which these requirements are satisfied by SGD. We rely on standard characterizations of the objective function—namely, convexity, Lipschitzness and smoothness—the definitions of which are deferred to Appendix A, along with all proofs from this section.

A recent study by Hardt et al. [13] proved that some special cases of SGD—when examples are sampled uniformly, with replacement—satisfy $\beta_{\mathcal{Z}}$-uniform stability (Definition 1) with $\beta_{\mathcal{Z}} = \mathrm{O}(n^{-1})$. We extend their work (specifically, [13, Theorem 3.7]) in the following result for SGD with a convex objective function, when the step size is at most inversely proportional to the current iteration.

Proposition 1. *Assume that the loss function, L, is λ-Lipschitz, and that the objective function, F, is convex, λ-Lipschitz and σ-smooth. Suppose SGD is run for T iterations with a uniform sampling distribution, $\mathbb{P}$, and step sizes $\eta_t \in [0, \eta/t]$, for $\eta \in [0, 2/\sigma]$. Then, SGD is both $\beta_{\mathcal{Z}}$-uniformly stable and $\beta_{\mathcal{Z}}$-pointwise hypothesis stable with respect to L and $\mathbb{P}$, with*

$$\beta_{\mathcal{Z}} \leq \frac{2\lambda^2 \eta \left(\ln T + 1\right)}{n}. \tag{1}$$

When $T = \Theta(n)$, Equation 1 is $\tilde{\mathrm{O}}(n^{-1})$, which is acceptable for proving generalization.

If we do not assume that the objective function is convex, we can borrow a result (with small modification[2]) from Hardt et al. [13, Theorem 3.8].

Proposition 2. *Assume that the loss function, L, is M-bounded and λ-Lipschitz, and that the objective function, F, is λ-Lipschitz and σ-smooth. Suppose SGD is run for T iterations with a uniform sampling distribution, $\mathbb{P}$, and step sizes $\eta_t \in [0, \eta/t]$, for $\eta \geq 0$. Then, SGD is both $\beta_{\mathcal{Z}}$-uniformly stable and $\beta_{\mathcal{Z}}$-pointwise hypothesis stable with respect to L and $\mathbb{P}$, with*

$$\beta_{\mathcal{Z}} \leq \left(\frac{M + (\sigma\eta)^{-1}}{n-1}\right) \left(2\lambda^2\eta\right)^{\frac{1}{\sigma\eta+1}} T^{\frac{\sigma\eta}{\sigma\eta+1}}. \tag{2}$$

Assuming $T = \Theta(n)$, and ignoring constants that depend on M, λ, σ and η, Equation 2 reduces to $\mathrm{O}\left(n^{-\frac{1}{\sigma\eta+1}}\right)$. As $\sigma\eta$ approaches 1, the rate becomes $\mathrm{O}(n^{-1/2})$, which, as will become evident in Section 4, yields generalization bounds that are suboptimal, or even vacuous. However, if $\sigma\eta$ is small—say, $\eta = (10\sigma)^{-1}$—then we get $\mathrm{O}\left(n^{-\frac{10}{11}}\right) \approx \mathrm{O}(n^{-1})$, which suffices for generalization.

We can obtain even tighter bounds for $\beta_{\mathcal{Z}}$-pointwise hypothesis stability (Definition 2) by adopting a data-dependent view. The following result for SGD with a convex objective function is adapted from work by Kuzborskij and Lampert [18, Theorem 3].

Proposition 3. *Assume that the loss function, L, is λ-Lipschitz, and that the objective function, F, is convex, λ-Lipschitz and σ-smooth. Suppose SGD starts from an initial hypothesis, h_0, and is run for T iterations with a uniform sampling distribution, $\mathbb{P}$, and step sizes $\eta_t \in [0, \eta/t]$, for $\eta \in [0, 2/\sigma]$. Then, SGD is $\beta_{\mathcal{Z}}$-pointwise hypothesis stable with respect to L and $\mathbb{P}$, with*

$$\beta_{\mathcal{Z}} \leq \frac{2\lambda\eta \left(\ln T + 1\right) \sqrt{2\sigma\, \mathbb{E}_{z\sim\mathbb{D}}[L(h_0, z)]}}{n}. \tag{3}$$

Importantly, Equation 3 depends on the risk of the initial hypothesis, h_0. If h_0 happens to be close to a global optimum—that is, a good first guess—then Equation 3 could be tighter than Equation 1. Kuzborskij and Lampert also proved a data-dependent bound for non-convex objective functions [18, Theorem 5], which, under certain conditions, might be tighter than Equation 2. Though not presented herein, Kuzborskij and Lampert's bound is worth noting.

As we will later show, we can obtain stronger generalization guarantees by combining $\beta_{\mathcal{Z}}$-uniform stability with β_{Θ}-uniform stability (Definition 3), provided $\beta_{\Theta} = \tilde{\mathrm{O}}(1/\sqrt{nT})$. Prior stability analyses of SGD [13, 18] have not addressed this form of stability. Elisseeff et al. [8] proved $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniform stability for certain bagging algorithms, but did not consider SGD. In light of Remark 1, it is tempting to map β_{Θ}-uniform stability to Bousquet and Elisseeff's [3] uniform stability and thereby leverage their study of various regularized objective functions. However, their analysis crucially relies on exact minimization of the learning objective, whereas SGD with a finite number of steps only finds an approximate minimizer. Thus, to our knowledge, no prior work applies to this problem. As a first step, we prove uniform stability, with respect to both data and hyperparameters, for SGD with a strongly convex objective function and decaying step sizes.

Proposition 4. *Assume that the loss function, L, is λ-Lipschitz, and that the objective function, F, is γ-strongly convex, λ-Lipschitz and σ-smooth. Suppose SGD is run for T iterations with a uniform sampling distribution, $\mathbb{P}$, and step sizes $\eta_t \triangleq (\gamma t + \sigma)^{-1}$. Then, SGD is $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniformly stable with respect to L and $\mathbb{P}$, with*

$$\beta_{\mathcal{Z}} \leq \frac{2\lambda^2}{\gamma n} \quad \text{and} \quad \beta_{\Theta} \leq \frac{2\lambda^2}{\gamma T}. \tag{4}$$

When $T = \Theta(n)$, the β_{Θ} bound in Equation 4 is $\mathrm{O}(1/\sqrt{nT})$, which supports good generalization.

---

[2]Hardt et al.'s definition of stability and theorem statement differ slightly from ours. See Appendix A.1.

4 Generalization Bounds

In this section, we present new generalization bounds for randomized learning algorithms. While prior work [8, 9] has addressed this topic, ours is the first PAC-Bayesian treatment (the benefits of which will be discussed momentarily). Recall that in the PAC-Bayesian framework, we fix a prior distribution, $\mathbb{P}$, on the hypothesis space, $\mathcal{H}$; then, given a sample of training data, $S \sim \mathbb{D}^n$, we learn a posterior distribution, $\mathbb{Q}$, also on $\mathcal{H}$. In our extension for randomized learning algorithms, $\mathbb{P}$ and $\mathbb{Q}$ are instead supported on the hyperparameter space, Θ. Moreover, while traditional PAC-Bayes studies $\mathbb{E}_{h \sim \mathbb{Q}}[L(h, z)]$, we study the expected loss over draws of hyperparameters, $\mathbb{E}_{\theta \sim \mathbb{Q}}[L(A(S, \theta), z)]$. Our goal will be to upper-bound the generalization error of the posterior, $G(S, \mathbb{Q})$, which thereby upper-bounds the risk, $R(S, \mathbb{Q})$, by a function of the empirical risk, $\hat{R}(S, \mathbb{Q})$.

Importantly, our bounds are polynomial in δ^{-1}, for a free parameter $\delta \in (0, 1)$, and hold with probability at least $1 - \delta$ over draws of a finite training dataset. This stands in contrast to related bounds [2, 13, 18, 21] that hold in expectation. While expectation bounds are useful for gaining insight into generalization behavior, high-probability bounds are sometimes preferred. Provided the loss is M-bounded, it is always possible to convert a high-probability bound of the form $\Pr_{S \sim \mathbb{D}^n}\{G(S, \mathbb{Q}) \leq B(\delta)\} \geq 1 - \delta$ to an expectation bound of the form $\mathbb{E}_{S \sim \mathbb{D}^n}[G(S, \mathbb{Q})] \leq B(\delta) + \delta M$.

Another useful property of PAC-Bayesian bounds is that they hold simultaneously for all posteriors, including those that depend on the training data. In Section 3, we assumed that hyperparameters were sampled according to a fixed distribution; for instance, sampling training example indices for SGD uniformly at random. However, in certain situations, it may be advantageous to sample according to a data-dependent distribution. Following the SGD example, suppose most training examples are *easy* to classify (e.g., far from the decision boundary), but some are *difficult* (e.g., near the decision boundary, or noisy). If we sample points uniformly at random, we might encounter mostly easy examples, which could slow progress on difficult examples. If we instead focus training on the difficult set, we might converge more quickly to an optimal hypothesis. Since our PAC-Bayesian bounds hold for all hyperparameter posteriors, we can characterize the generalization error of algorithms that optimize the posterior using the training data. Existing generalization bounds for randomized learning [8, 9], or SGD in particular [2, 13, 18, 20, 21], cannot address such algorithms. Of course, there is a penalty for overfitting the posterior to the data, which is captured by the posterior's divergence from the prior.

Our first PAC-Bayesian theorem requires the weakest stability condition, $\beta_{\mathcal{Z}}$-pointwise hypothesis stability, but the bound is sublinear in δ^{-1}. Our second bound is polylogarithmic in δ^{-1}, but requires the stronger stability conditions, $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniform stability. All proofs are deferred to Appendix B.

Theorem 1. *Suppose a randomized learning algorithm, A, is $\beta_{\mathcal{Z}}$-pointwise hypothesis stable with respect to an M-bounded loss function, L, and a fixed prior, $\mathbb{P}$ on Θ. Then, for any $n \geq 1$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over draws of a dataset, $S \sim \mathbb{D}^n$, every posterior, $\mathbb{Q}$ on Θ, satisfies*

$$G(S, \mathbb{Q}) \leq \sqrt{\left(\frac{\chi^2(\mathbb{Q}\|\mathbb{P}) + 1}{\delta}\right)\left(\frac{2M^2}{n} + 12M\beta_{\mathcal{Z}}\right)}, \tag{5}$$

where $\chi^2(\mathbb{Q}\|\mathbb{P}) \triangleq \mathbb{E}_{\theta \sim \mathbb{P}}\left[\left(\frac{\mathbb{Q}(\theta)}{\mathbb{P}(\theta)}\right)^2 - 1\right]$ is the χ^2 divergence from $\mathbb{P}$ to $\mathbb{Q}$.

Theorem 2. *Suppose a randomized learning algorithm, A, is $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniformly stable with respect to an M-bounded loss function, L, and a fixed product measure, $\mathbb{P}$ on $\Theta = \prod_{t=1}^{T} \Theta_t$. Then, for any $n \geq 1$, $T \geq 1$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over draws of a dataset, $S \sim \mathbb{D}^n$, every posterior, $\mathbb{Q}$ on Θ, satisfies*

$$G(S, \mathbb{Q}) \leq \beta_{\mathcal{Z}} + \sqrt{2\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta}\right)\left(\frac{(M + 2n\beta_{\mathcal{Z}})^2}{n} + 4T\beta_{\Theta}^2\right)}, \tag{6}$$

where $D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) \triangleq \mathbb{E}_{\theta \sim \mathbb{Q}}\left[\ln\left(\frac{\mathbb{Q}(\theta)}{\mathbb{P}(\theta)}\right)\right]$ is the KL divergence from $\mathbb{P}$ to $\mathbb{Q}$.

Since Theorems 1 and 2 hold simultaneously for all hyperparameter posteriors, they provide generalization guarantees for SGD with any sampling distribution. Note that the stability requirements only need to be satisfied by a fixed product measure, such as a uniform distribution. This simple

sampling distribution can have $\left(\mathrm{O}(n^{-1}), \mathrm{O}(T^{-1})\right)$-uniform stability under certain conditions, as demonstrated in Section 3.2. In the following, we apply Theorem 2 to SGD with a strongly convex objective function, leveraging Proposition 4 to upper-bound the stability coefficients.

Corollary 1. *Assume that the loss function, L, is M-bounded and λ-Lipschitz, and that the objective function, F, is γ-strongly convex, λ-Lipschitz and σ-smooth. Let $\mathbb{P}$ denote a uniform prior on $\{1, \ldots, n\}^T$. Then, for any $n \geq 1$, $T \geq 1$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over draws of a dataset, $S \sim \mathbb{D}^n$, SGD with step sizes $\eta_t \triangleq (\gamma t + \sigma)^{-1}$ and any posterior sampling distribution, $\mathbb{Q}$ on $\{1, \ldots, n\}^T$, satisfies*

$$G(S, \mathbb{Q}) \leq \frac{2\lambda^2}{\gamma n} + \sqrt{2\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta}\right)\left(\frac{(M + 4\lambda^2/\gamma)^2}{n} + \frac{16\lambda^4}{\gamma^2 T}\right)}.$$

When the divergence is polylogarithmic in n, and $T = \Theta(n)$, the generalization bound is $\tilde{\mathrm{O}}(n^{-1/2})$. In the special case of uniform sampling, the KL divergence is zero, yielding a $\mathrm{O}(n^{-1/2})$ bound.

Importantly, Theorem 1 does not require hyperparameter stability, and is therefore of interest for analyzing non-convex objective functions, since it is not known whether uniform hyperparameter stability can be satisfied without (strong) convexity. One can use Equation 2 (or [18, Theorem 5]) to upper-bound $\beta_{\mathcal{Z}}$ in Equation 5 and thereby obtain a generalization bound for SGD with a non-convex objective function, such as neural network training. We leave this substitution to the reader.

Equation 6 holds with high probability over draws of a dataset, but the generalization error is an expected value over draws of hyperparameters. To obtain a bound that holds with high probability over draws of both data and hyperparameters, we consider posteriors that are product measures.

Theorem 3. *Suppose a randomized learning algorithm, A, is $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniformly stable with respect to an M-bounded loss function, L, and a fixed product measure, $\mathbb{P}$ on $\Theta = \prod_{t=1}^{T} \Theta_t$. Then, for any $n \geq 1$, $T \geq 1$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over draws of a dataset, $S \sim \mathbb{D}^n$, and hyperparameters, $\theta \sim \mathbb{Q}$, from any posterior product measure, $\mathbb{Q}$ on Θ,*

$$G(S, \theta) \leq \beta_{\mathcal{Z}} + \beta_{\Theta}\sqrt{2T\ln\frac{2}{\delta}} + \sqrt{2\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{4}{\delta}\right)\left(\frac{(M + 2n\beta_{\mathcal{Z}})^2}{n} + 4T\beta_{\Theta}^2\right)}. \quad (7)$$

If $\beta_{\Theta} = \tilde{\mathrm{O}}(1/\sqrt{nT})$, then $\beta_{\Theta}\sqrt{2T\ln\frac{2}{\delta}}$ vanishes at a rate of $\tilde{\mathrm{O}}(n^{-1/2})$. We can apply Theorem 3 to SGD in the same way we applied Theorem 2 in Corollary 1. Further, note that a uniform distribution is a product distribution. Thus, if we eschew optimizing the posterior, then the KL divergence disappears, leaving a $\mathrm{O}(n^{-1/2})$ derandomized generalization bound for SGD with uniform sampling.[3]

5 Adaptive Sampling for Stochastic Gradient Descent

The PAC-Bayesian theorems in Section 4 motivate data-dependent posterior distributions on the hyperparameter space. Intuitively, certain posteriors may improve, or speed up, learning from a given dataset. For instance, suppose certain training examples are considered valuable for reducing empirical risk; then, a sampling posterior for SGD should weight those examples more heavily than others, so that the learning algorithm can, probabilistically, focus its attention on the valuable examples. However, a posterior should also try to stay close to the prior, to control the divergence penalty in the generalization bounds.

Based on this idea, we propose a sampling procedure for SGD (or any variant thereof) that constructs a posterior based on the training data, balancing the utility of the sampling distribution with its divergence from a uniform prior. The algorithm operates alongside the learning algorithm, iteratively generating the posterior as a sequence of conditional distributions on the training data. Each iteration of training generates a new distribution conditioned on the previous iterations, so the posterior dynamically adapts to training. We therefore call our algorithm *adaptive sampling SGD*.

---

[3]We can achieve the same result by pairing Proposition 4 with Elisseeff et al.'s generalization bound for algorithms with $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniform stability [8, Theorem 15]. However, Elisseeff et al.'s bound only applies to fixed product measures on Θ, whereas Theorem 3 applies more generally to any posterior product measure, and when $\mathbb{P} = \mathbb{Q}$, Equation 7 is within a constant factor of Elisseeff et al.'s bound.

Algorithm 1 Adaptive Sampling SGD

Require: Examples, $(z_1, \ldots, z_n) \in \mathcal{Z}^n$; initial hypothesis, $h_0 \in \mathcal{H}$; update rule, $U_t : \mathcal{H} \times \mathcal{Z} \to \mathcal{H}$; utility function, $f : \mathcal{Z} \times \mathcal{H} \to \mathbb{R}$; amplitude, $\alpha \geq 0$; decay, $\tau \in (0, 1)$.

1: $(q_1, \ldots, q_n) \leftarrow \mathbf{1}$ ▷ Initialize sampling weights uniformly
2: **for** $t = 1, \ldots, T$ **do**
3:     $i_t \sim \mathbb{Q}_t \propto (q_1, \ldots, q_n)$ ▷ Draw index i_t proportional to sampling weights
4:     $h_t \leftarrow U_t(h_{t-1}, z_{i_t})$ ▷ Update hypothesis
5:     $q_{i_t} \leftarrow q_{i_t}^\tau \exp\left(\alpha\, f(z_{i_t}, h_t)\right)$ ▷ Update sampling weight for i_t
6: **return** h_T

Algorithm 1 maintains a set of nonnegative sampling weights, $(q_1, \ldots, q_n)$, which define a distribution on the dataset. The posterior probability of the ith example in the tth iteration, given the previous iterations, is proportional to the ith weight: $\mathbb{Q}_t(i) \triangleq \mathbb{Q}(i_t = i \,|\, i_1, \ldots, i_{t-1}) \propto q_i$. The sampling weights are initialized to 1, thereby inducing a uniform distribution. At each iteration, we draw an index, $i_t \sim \mathbb{Q}_t$, and use example z_{i_t} to update the hypothesis. We then update the weight for i_t multiplicatively as $q_{i_t} \leftarrow q_{i_t}^\tau \exp\left(\alpha\, f(z_{i_t}, h_t)\right)$, where: $f(z_{i_t}, h_t)$ is a *utility* function of the chosen example and current hypothesis; $\alpha \geq 0$ is an *amplitude* parameter, which controls the aggressiveness of the update; and $\tau \in (0, 1)$ is a *decay* parameter, which lets q_i gradually forget past updates.

The multiplicative weight update (line 5) can be derived by choosing a sampling distribution for the next iteration, $t + 1$, that maximizes the expected utility while staying close to a reference distribution. Consider the following constrained optimization problem:

$$\max_{\mathbb{Q}_{t+1} \in \Delta^n} \sum_{i=1}^n \mathbb{Q}_{t+1}(i) f(z_i, h_t) - \frac{1}{\alpha}\, D_{\text{KL}}(\mathbb{Q}_{t+1} \| \mathbb{Q}_t^\tau). \tag{8}$$

The term $\sum_{i=1}^n \mathbb{Q}_{t+1}(i) f(z_i, h_t)$ is the expected utility under the new distribution, $\mathbb{Q}_{t+1}$. This is offset by the KL divergence, which acts as a regularizer, penalizing $\mathbb{Q}_{t+1}$ for diverging from a reference distribution, $\mathbb{Q}_t^\tau$, where $\mathbb{Q}_t^\tau(i) \propto q_i^\tau$. The decay parameter, τ, controls the *temperature* of the reference distribution, allowing it to interpolate between the current distribution ($\tau = 1$) and a uniform distribution ($\tau = 0$). The amplitude parameter, α, scales the influence of the regularizer relative to the expected utility. We can solve Equation 8 analytically using the method of Lagrange multipliers, which yields

$$\mathbb{Q}_{t+1}^\star(i) \propto \mathbb{Q}_t^\tau(i) \exp\left(\alpha\, f(z_{i_t}, h_t) - 1\right) \propto q_i^\tau \exp\left(\alpha\, f(z_{i_t}, h_t)\right).$$

Updating q_i for all $i = 1, \ldots, n$ is impractical for large n, so we approximate the above solution by only updating the weight for the last sampled index, i_t, effectively performing coordinate ascent.

The idea of tuning the empirical data distribution through multiplicative weight updates is reminiscent of AdaBoost [10] and focused online learning [28], but note that Algorithm 1 learns a single hypothesis, not an ensemble. In this respect, it is similar to SelfieBoost [27]. One could also draw parallels to exponentiated gradient dual coordinate ascent [5]. Finally, note that when the gradient estimate is unbiased (i.e., weighted by the inverse sampling probability), we obtain a variant of importance sampling SGD [31], though we do not necessarily need unbiased gradient estimates.

It is important to note that we do not actually need to compute the full posterior distribution—which would take $\mathrm{O}(n)$ time per iteration—in order to sample from it. Indeed, using an algorithm and data structure described in Appendix C, we can sample from and update the distribution in $\mathrm{O}(\log n)$ time, using $\mathrm{O}(n)$ space. Thus, the additional iteration complexity of adaptive sampling is logarithmic in the size of the dataset, which suitably efficient for learning from large datasets.

In practice, SGD is typically applied with *mini-batching*, whereby multiple examples are drawn at each iteration, instead of just one. Given the massive parallelism of today's computing hardware, mini-batching is simply a more efficient way to process a dataset, and can result in more accurate gradient estimates than single-example updates. Though Algorithm 1 is stated for single-example updates, it can be modified for mini-batching by replacing line 3 with multiple independent draws from $\mathbb{Q}_t$, and line 5 with sampling weight updates for each unique[4] example in the mini-batch.

---

[4]If an example is drawn multiple times in a mini-batch, its sampling weight is only updated once.

5.1 Divergence Analysis

Recall that our generalization bounds use the posterior's divergence from a fixed prior to penalize the posterior for overfitting the training data. Thus, to connect Algorithm 1 to our bounds, we analyze the adaptive posterior's divergence from a uniform prior on $\{1, \ldots, n\}^T$. This quantity reflects the potential cost, in generalization performance, of adaptive sampling. The goal of this section is to upper-bound the KL divergence resulting from Algorithm 1 in terms of interpretable, data-dependent quantities. All proofs are deferred to Appendix D.

Our analysis requires introducing some notation. Given a sequence of sampled indices, $(i_1, \ldots, i_t)$, let $N_{i,t} \triangleq |\{t' : t' < t, i_{t'} = i\}|$ denote the number of times that index i was chosen *before* iteration t. Let $O_{i,j}$ denote the jth iteration in which i was chosen; for instance, if i was chosen at iterations 13 and 47, then $O_{i,1} = 13$ and $O_{i,2} = 47$. With these definitions, we can state the following bound, which exposes the influences of the utility function, amplitude and decay on the KL divergence.

Theorem 4. *Fix a uniform prior, $\mathbb{P}$, a utility function, $f : \mathcal{Z} \times \mathcal{H} \to \mathbb{R}$, an amplitude, $\alpha \geq 0$, and a decay, $\tau \in (0, 1)$. If Algorithm 1 is run for T iterations, then its posterior, $\mathbb{Q}$, satisfies*

$$D_{\text{KL}}(\mathbb{Q}\|\mathbb{P}) \leq \sum_{t=2}^{T} \mathop{\mathbb{E}}_{(i_1,\ldots,i_t)\sim\mathbb{Q}} \frac{\alpha}{n} \sum_{i=1}^{n} \left[\sum_{j=1}^{N_{i_t,t}} f(z_{i_t}, h_{O_{i_t,j}}) \tau^{N_{i_t,t}-j} - \sum_{k=1}^{N_{i,t}} f(z_i, h_{O_{i,k}}) \tau^{N_{i,t}-k} \right]. \quad (9)$$

Equation 9 can be interpreted as measuring, on average, how the cumulative past utilities of each sampled index, i_t, differ from the cumulative utilities of any other index, i.[5] When the posterior becomes too focused on certain examples, this difference is large. The accumulated utilities decay exponentially, with the rate of decay controlled by τ. The amplitude, α, scales the entire bound, which means that aggressive posterior updates may adversely affect generalization.

An interesting special case of Theorem 4 is when the utility function is nonnegative, which results in a simpler, more interpretable bound.

Theorem 5. *Fix a uniform prior, $\mathbb{P}$, a nonnegative utility function, $f : \mathcal{Z} \times \mathcal{H} \to \mathbb{R}_+$, an amplitude, $\alpha \geq 0$, and a decay, $\tau \in (0, 1)$. If Algorithm 1 is run for T iterations, then its posterior, $\mathbb{Q}$, satisfies*

$$D_{\text{KL}}(\mathbb{Q}\|\mathbb{P}) \leq \frac{\alpha}{1-\tau} \sum_{t=1}^{T-1} \mathop{\mathbb{E}}_{(i_1,\ldots,i_t)\sim\mathbb{Q}} \left[f(z_{i_t}, h_t) \right]. \quad (10)$$

Equation 10 is simply the sum of expected utilities computed over $T-1$ iterations of training, scaled by $\alpha/(1-\tau)$. The implications of this bound are interesting when the utility function is defined as the loss, $f(z, h) \triangleq L(h, z)$; then, if SGD quickly converges to a hypothesis with low *maximal* loss on the training data, it can reduce the generalization error.[6] The caveat is that tuning the amplitude or decay to speed up convergence may actually counteract this effect.

It is worth noting that similar guarantees hold for a mini-batch variant of Algorithm 1. The bounds are essentially unchanged, modulo notational intricacies.

6 Experiments

To demonstrate the effectiveness of Algorithm 1, we conducted several experiments with the CIFAR-10 dataset [16]. This benchmark dataset contains 60,000 (32×32)-pixel RGB images from 10 object classes, with a standard, static partitioning into 50,000 training examples and 10,000 test examples.

We specified the hypothesis class as the following convolutional neural network architecture: 32 (3×3) filters with rectified linear unit (ReLU) activations in the first and second layers, followed by (2×2) max-pooling and 0.25 dropout[7]; 64 (3×3) filters with ReLU activations in the third and fourth layers, again followed by (2×2) max-pooling and 0.25 dropout; finally, a fully-connected, 512-unit layer with ReLU activations and 0.5 dropout, followed by a fully-connected, 10-output softmax layer. We trained the network using the cross-entropy loss. We emphasize that our goal was

---

[5]When $N_{i,t} = 0$ (i.e., i has not yet been sampled), a summation over $j = 1, \ldots, N_{i,t}$ evaluates to zero.

[6]This interpretation concurs with ideas in [13, 28].

[7]It can be shown that dropout improves data stability [13, Lemma 4.4].

not to achieve state-of-the-art results on the dataset; rather, to evaluate Algorithm 1 in a simple, yet realistic, application.

Following the intuition that sampling should focus on difficult examples, we experimented with two utility functions for Algorithm 1 based on common loss functions. For an example $z = (x, y)$, with $h(x, y)$ denoting the predicted probability of label y given input x under hypothesis h, let

$$f_0(z, h) \triangleq \mathbb{1}\{\arg\max_{y' \in \mathcal{Y}} h(x, y') \neq y\} \quad \text{and} \quad f_1(z, h) \triangleq 1 - h(x, y).$$

The first utility function, f_0, is the 0-1 loss; the second, f_1, is the L_1 loss, which accounts for uncertainty in the most likely label. We combined these utility functions with two parameter update rules: standard SGD with decreasing step sizes, $\eta_t \triangleq \eta/(1+\nu t) \leq \eta/(\nu t)$, for $\eta > 0$ and $\nu > 0$; and AdaGrad [7], a variant of SGD that automatically tunes a separate step size for each parameter. We used mini-batches of 100 examples per update. The combination of utility functions and update rules yields four adaptive sampling algorithms: AdaSamp-01-SGD, AdaSamp-01-AdaGrad, AdaSamp-L1-SGD and AdaSamp-L1-AdaGrad. We compared these to their uniform sampling counterparts, Unif-SGD and Unif-AdaGrad.

We tuned all hyperparameters using random subsets of the training data for cross-validation. We then ran 10 trials of training and testing, using different seeds for the pseudorandom number generator at each trial to generate different random initializations[8] and training sequences. Figures 1a and 1b plot learning curves of the average cross-entropy and accuracy, respectively, on the training data; Figure 1c plots the average accuracy on the test data. We found that all adaptive sampling variants reduced empirical risk (increased training accuracy) faster than their uniform sampling counterparts. Further, AdaGrad with adaptive sampling exhibited modest, yet consistent, improvements in test accuracy in early iterations of training. Figure 1d illustrates the effect of varying the amplitude parameter, α. Higher values of α led to faster empirical risk reduction, but lower test accuracy—a sign of overfitting the posterior to the data, which concurs with Theorems 4 and 5 regarding the influence of α on the KL divergence. Figure 1e plots the KL divergence from the conditional prior, $\mathbb{P}_t$, to the conditional posterior, $\mathbb{Q}_t$, given sampled indices $(i_1, \ldots, i_{t-1})$; i.e., $D_{\mathrm{KL}}(\mathbb{Q}_t \| \mathbb{P}_t)$. The sampling distribution quickly diverged in early iterations, to focus on examples where the model erred, then gradually converged to a uniform distribution as the empirical risk converged.

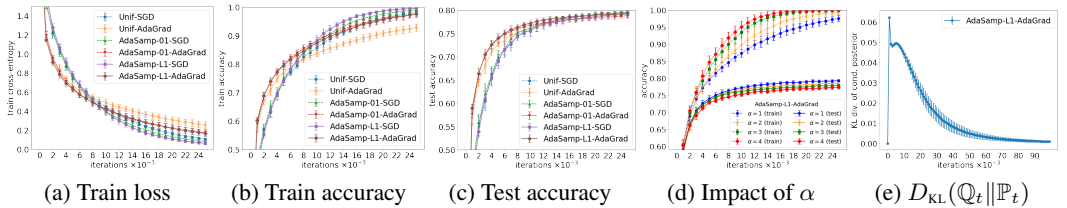| (a) Train loss | (b) Train accuracy | (c) Test accuracy | (d) Impact of α | (e) $D_{\mathrm{KL}}(\mathbb{Q}_t \| \mathbb{P}_t)$ |

Figure 1: Experimental results on CIFAR-10, averaged over 10 random initializations and training runs. (Best viewed in color.) Figure 1a plots learning curves of training cross-entropy (lower is better). Figures 1b and 1c, respectively, plot train and test accuracies (higher is better). Figure 1d highlights the impact of the amplitude parameter, α, on accuracy. Figure 1e plots the KL divergence from the conditional prior, $\mathbb{P}_t$, to the conditional posterior, $\mathbb{Q}_t$, given sampled indices $(i_1, \ldots, i_{t-1})$.

7 Conclusions and Future Work

We presented new generalization bounds for randomized learning algorithms, using a novel combination of PAC-Bayes and algorithmic stability. The bounds inspired an adaptive sampling algorithm for SGD that dynamically updates the sampling distribution based on the training data and model. Experimental results with this algorithm indicate that it can reduce empirical risk faster than uniform sampling while also improving out-of-sample accuracy. Future research could investigate different utility functions and distribution updates, or explore the connections to related algorithms. We are also interested in providing stronger generalization guarantees, with polylogarithmic dependence on δ^{-1}, for non-convex objective functions, but proving $\tilde{O}(1/\sqrt{nT})$-uniform hyperparameter stability without (strong) convexity is difficult. We hope to address this problem in future work.

---

[8]Each training algorithm started from the same initial hypothesis.

A Proofs from Section 3

The stability bounds in Section 3 require several characterizations of a loss or objective function. In the following definitions, we consider generic functions of the form $\varphi : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$. Since we are only interested in how a function behaves with respect to $\mathcal{H}$, we specify the definitions accordingly.

Definition 4 (Convexity). A differentiable function, $\varphi : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$, is convex (in $\mathcal{H}$) if

$$\forall h, h' \in \mathcal{H}, \ \forall z \in \mathcal{Z}, \ \ \langle \nabla \varphi(h, z), h' - h \rangle \le \varphi(h', z) - \varphi(h, z).$$

Further, φ is γ-strongly convex (with respect to the 2-norm) if

$$\frac{\gamma}{2} \|h' - h\|^2 + \langle \nabla \varphi(h, z), h' - h \rangle \le \varphi(h', z) - \varphi(h, z).$$

Definition 5 (Lipschitzness). A function, $\varphi : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$, is λ-Lipschitz (in $\mathcal{H}$) if

$$\sup_{h, h' \in \mathcal{H}} \sup_{z \in \mathcal{Z}} \frac{|\varphi(h, z) - \varphi(h', z)|}{\|h - h'\|} \le \lambda. \tag{11}$$

If φ is differentiable, then Equation 11 is equivalent to

$$\sup_{h \in \mathcal{H}} \sup_{z \in \mathcal{Z}} \|\nabla \varphi(h, z)\| \le \lambda.$$

Definition 6 (Smoothness). A differentiable function, $\varphi : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$, is σ-smooth (in $\mathcal{H}$) if

$$\sup_{h, h' \in \mathcal{H}} \sup_{z \in \mathcal{Z}} \frac{\|\nabla \varphi(h, z) - \nabla \varphi(h', z)\|}{\|h - h'\|} \le \sigma.$$

Smoothness is a form of Lipschitzness; a function is σ-smooth if its gradient is σ-Lipschitz.

A.1 Proof of Propositions 1 to 3

Propositions 1 to 3 extend work by Hardt et al. [13] and Kuzborskij and Lampert [18], whose definitions of data stability differ slightly from ours (which are taken from [8]). To reconcile our definition of $\beta_{\mathcal{Z}}$-uniform stability with Hardt et al.'s, which does not involve an absolute value, observe that

$$\sup_{S, S', z} \left| \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [L(A(S, \theta), z) - L(A(S', \theta), z)] \right| = \sup_{S, S', z} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [L(A(S, \theta), z) - L(A(S', \theta), z)]$$

by the symmetry of the supremum over S and S'. Kuzborskij and Lampert's definition of hypothesis stability equates to our *pointwise* hypothesis stability, though they do not include an absolute value inside the expectation over $\theta \sim \mathbb{P}$. Nonetheless, since the loss function is always assumed to be λ-Lipschitz, this distinction does not matter. Indeed, all existing stability proofs for SGD implicitly leverage the following upper bound:

$$\mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [L(A(S, \theta), z) - L(A(S', \theta), z)] \le \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [|L(A(S, \theta), z) - L(A(S', \theta), z)|]$$

$$\le \lambda \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [\|A(S, \theta) - A(S', \theta)\|]. \tag{12}$$

Equation 12 implies that Kuzborskij and Lampert's proofs hold for our definition of pointwise hypothesis stability; we simply start the proof from the right-hand side of the first inequality. By the same logic, we can convert existing proofs of $\beta_{\mathcal{Z}}$-uniform stability to proofs of $\beta_{\mathcal{Z}}$-pointwise hypothesis stability. Moreover, Equation 12 lets us distinguish between the loss function, L, and the objective function, F, which is optimized by A; though [13, 18] do not make this distinction, their results hold when $L \ne F$ because they assume Lipschitzness.

Using the above reasoning, we therefore arrive at the following adaptations, which will be used to prove Propositions 1 and 3.

Lemma 1 (adapted from [13, Theorem 3.7]). *Assume that the loss function, L, is λ-Lipschitz, and that the objective function, F, is convex, λ-Lipschitz and σ-smooth. Suppose SGD is run for T iterations with a uniform sampling distribution, $\mathbb{P}$, and step sizes $\eta_t \in [0, 2/\sigma]$. Then, SGD is both $\beta_{\mathcal{Z}}$-uniformly stable and $\beta_{\mathcal{Z}}$-pointwise hypothesis stable with respect to L and $\mathbb{P}$, with*

$$\beta_{\mathcal{Z}} \le \frac{2\lambda^2}{n} \sum_{t=1}^{T} \eta_t.$$

Lemma 2 (adapted from [18, Theorem 3]). *Assume that the loss function, L, is λ-Lipschitz, and that the objective function, F, is convex, λ-Lipschitz and σ-smooth. Suppose SGD starts from an initial hypothesis, h_0, and is run for T iterations with a uniform sampling distribution, $\mathbb{P}$, and step sizes $\eta_t \in [0, 2/\sigma]$. Then, SGD is $\beta_{\mathcal{Z}}$-pointwise hypothesis stable with respect to L and $\mathbb{P}$, with*

$$\beta_{\mathcal{Z}} \leq \frac{2\lambda\sqrt{2\sigma\, \mathbb{E}_{z\sim\mathbb{D}}[L(h_0, z)]}}{n} \sum_{t=1}^{T} \eta_t.$$

If $\eta \leq 2/\sigma$, then $\eta_t \leq \eta/t \leq 2/\sigma$ for all $t \geq 1$. We thus have from Lemma 1 that

$$\beta_{\mathcal{Z}} \leq \frac{2\lambda^2\eta}{n} \sum_{t=1}^{T} \frac{1}{t} \leq \frac{2\lambda^2\eta}{n} \left(\ln T + 1\right),$$

which proves Proposition 1. The last inequality follows from the fact that the Tth harmonic number, $\sum_{t=1}^{T} \frac{1}{t}$, is upper-bounded by $\ln T + 1$. We obtain Proposition 3 from Lemma 2 using an identical proof.

Proposition 2 follows from [13, Theorem 3.8] with a few small modifications. As previously mentioned, we can use Equation 12 to reconcile definitional differences, distinguish L from F, and adapt the proof for pointwise hypothesis complexity. We also assume that L is M-bounded instead of 1-bounded, so we use $\sup_{h,z} L(h, z) \leq M$ in the proof (see [12, Lemma 3.11]).

A.2 Proof of Proposition 4

We characterize SGD updates using the following definition, borrowed from Hardt et al. [13].

Definition 7 (Expansivity). An update rule, $U : \mathcal{H} \times \mathcal{Z} \to \mathcal{H}$, is α-expansive if

$$\sup_{h,h'\in\mathcal{H}} \sup_{z\in\mathcal{Z}} \frac{\|U(h, z) - U(h', z)\|}{\|h - h'\|} \leq \alpha.$$

We say that U is contractive if $\alpha \leq 1$.

Expansivity is yet another form of Lipschitzness; an update rule is α-expansive if it is α-Lipschitz.

We begin our proof with a fundamental technical lemma.

Lemma 3. *Assume that the objective function, F, is λ-Lipschitz. Further, assume that each SGD update, U_t, is α_t-expansive. If SGD is run for T iterations on two sequences of examples that differ at a single iteration, k, then the resulting learned hypotheses, h_T and h'_T, satisfy*

$$\|h_T - h'_T\| \leq 2\lambda\eta_k \prod_{t=k+1}^{T} \alpha_t.$$

Proof. For the first $k - 1$ iterations of SGD, the example sequences are the same; therefore, so are the learned weights. On processing the kth example, the weights may diverge, but we will show that the divergence is bounded, due to the Lipschitz property. For every iteration after k, the weights may continue to follow different trajectories, but the expansivity property lets us bound the difference of the final, learned weights.

Starting at T and recursing backward, we have that

$$\|h_T - h'_T\| \leq \|h_{T-1} - h'_{T-1}\| \, \alpha_T \leq \ldots \leq \|h_k - h'_k\| \prod_{t=k+1}^{T} \alpha_t. \tag{13}$$

Then, expanding the kth update,

$$\begin{aligned}
\|h_k - h'_k\| &= \|h_{k-1} - \eta_k \nabla F(h_{k-1}, z_k) - h_{k-1} + \eta_k \nabla F(h_{k-1}, z'_k)\| \\
&\leq \|\eta_k \nabla F(h_{k-1}, z_k)\| + \|\eta_k \nabla F(h_{k-1}, z'_k)\| \\
&\leq 2\eta_k \lambda. \tag{14}
\end{aligned}$$

Combining Equations 13 and 14 completes the proof. $\qquad\square$

We can now prove Proposition 4. First, note that $\eta_t \leq 1/\sigma$ for all $t \geq 1$. As noted by Hardt et al. [12, proof of Theorem 3.9], due to the strong convexity of the objective function, this step size guarantees that each update is contractive with coefficient $1 - \eta_t\gamma = 1 - (t + \sigma/\gamma)^{-1}$. Moreover [12, proof of Theorem 3.10],

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [\|h_T - h'_T\|] &\leq \sum_{k=1}^{T} \left(\prod_{t=k+1}^{T} (1 - \eta_t\gamma) \right) \eta_k \cdot \frac{2\lambda}{n} \\
&= \sum_{k=1}^{T} \left(\prod_{t=k+1}^{T} \left(1 - \frac{1}{t + \sigma/\gamma} \right) \right) \frac{1}{k + \sigma/\gamma} \cdot \frac{2\lambda}{\gamma n} \\
&= \sum_{k=1}^{T} \frac{k + \sigma/\gamma}{T + \sigma/\gamma} \cdot \frac{1}{k + \sigma/\gamma} \cdot \frac{2\lambda}{\gamma n} \\
&= \frac{T}{T + \sigma/\gamma} \cdot \frac{2\lambda}{\gamma n} \leq \frac{2\lambda}{\gamma n}.
\end{aligned}
\tag{15}
$$

Combining Equations 12 and 15 yields an upper bound on the data stability coefficient, $\beta_{\mathcal{Z}} \leq \frac{2\lambda^2}{\gamma n}$.

Now, suppose the example sequence is perturbed at any index k. Via Lemma 3, we have that

$$
\begin{aligned}
\|h_T - h'_T\| &\leq 2\lambda\eta_k \prod_{t=k+1}^{T} (1 - \eta_t\gamma) \\
&= \frac{2\lambda}{\gamma} \cdot \frac{1}{k + \sigma/\gamma} \prod_{t=k+1}^{T} \left(1 - \frac{1}{t + \sigma/\gamma} \right) \\
&= \frac{2\lambda}{\gamma} \cdot \frac{1}{k + \sigma/\gamma} \cdot \frac{k + \sigma/\gamma}{T + \sigma/\gamma} \leq \frac{2\lambda}{\gamma T},
\end{aligned}
$$

which we combine with the Lipschitz property (Equation 11) to obtain $\beta_{\Theta} \leq \frac{2\lambda^2}{\gamma T}$.

B   Proofs from Section 4

B.1   Stability of the Generalization Error

Our analysis in Section 4 uses stability to bound the moments and moment-generating function of the generalization error. To enable these proofs, we first derive some technical lemmas that relate stability in the loss to the stability in the generalization error. The first lemma applies to data stability; the second, to hyperparameter stability.

Lemma 4. *If A is $\beta_{\mathcal{Z}}$-uniformly stable with respect to an M-bounded loss function, L, and a distribution, $\mathbb{P}$, then, for any $S, S' \in \mathcal{Z}^n : D_{\mathrm{H}}(S, S') = 1$,*

$$
G(S, \mathbb{P}) - G(S', \mathbb{P}) \leq 2\beta_{\mathcal{Z}} + \frac{M}{n}.
$$

Proof. Observe that the difference of generalization errors decomposes as

$$
G(S, \mathbb{P}) - G(S', \mathbb{P}) = (R(S, \mathbb{P}) - R(S', \mathbb{P})) + (\hat{R}(S', \mathbb{P}) - \hat{R}(S, \mathbb{P})).
\tag{16}
$$

We will upper-bound each difference separately. First, using linearity of expectation and $\beta_{\mathcal{Z}}$-uniform stability, we have that

$$
R(S, \mathbb{P}) - R(S', \mathbb{P}) = \mathop{\mathbb{E}}_{z \sim \mathbb{D}} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [L(A(S, \theta), z) - L(A(S', \theta), z)] \leq \beta_{\mathcal{Z}}.
\tag{17}
$$

Then, without loss of generality, assume that S' differs from S at the ith example, denoted z_i'. Using $\beta_{\mathcal{Z}}$-uniform stability again,

$$
\begin{aligned}
\hat{R}(S', \mathbb{P}) - \hat{R}(S, \mathbb{P}) &= \frac{1}{n} \sum_{j \neq i} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \left[L(A(S', \theta), z_j) - L(A(S, \theta), z_j) \right] \\
&\quad + \frac{1}{n} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \left[L(A(S', \theta), z_i') - L(A(S, \theta), z_i) \right] \\
&\leq \frac{1}{n} \sum_{j \neq i} \beta_{\mathcal{Z}} + \frac{M}{n} \leq \beta_{\mathcal{Z}} + \frac{M}{n}.
\end{aligned}
\tag{18}
$$

Combining Equations 16 to 18 completes the proof. $\qquad\square$

Lemma 5. *If A is β_{Θ}-uniformly stable with respect to a loss function, L, then, for any $S \in \mathcal{Z}^n$ and $\theta, \theta' \in \Theta : D_{\mathrm{H}}(\theta, \theta') = 1$,*
$$
G(S, \theta) - G(S, \theta') \leq 2\beta_{\Theta}.
$$

Proof. The proof is almost identical to that of Lemma 4. First, we decompose the generalization error:
$$
G(S, \theta) - G(S, \theta') = (R(S, \theta) - R(S, \theta')) + (\hat{R}(S, \theta') - \hat{R}(S, \theta)).
\tag{19}
$$
Then, we upper-bound the difference of risk terms:
$$
R(S, \theta) - R(S, \theta') = \mathop{\mathbb{E}}_{z \sim \mathbb{D}} \left[L(A(S, \theta), z) - L(A(S, \theta'), z) \right] \leq \beta_{\Theta}.
\tag{20}
$$

Then, we upper-bound the difference of empirical risk terms:
$$
\hat{R}(S, \theta') - \hat{R}(S, \theta) = \frac{1}{n} \sum_{i=1}^{n} L(A(S, \theta'), z_i) - L(A(S, \theta), z_i) \leq \beta_{\Theta}.
\tag{21}
$$

Combining Equations 19 to 21 completes the proof. $\qquad\square$

Note that it is unnecessary to upper-bound the absolute difference in generalization error when using uniform stability, since it follows from the symmetry of the supremum over $S, S' \in \mathcal{Z}^n$ or $\theta, \theta' \in \Theta$.

B.2 Proof of Theorem 1

PAC-Bayesian analysis typically requires a key step known as *change of measure*. For our first bound, we use a change of measure inequality based on the *Rényi divergence*,
$$
D_{\alpha}(\mathbb{Q} \| \mathbb{P}) \triangleq \frac{1}{\alpha - 1} \ln \mathop{\mathbb{E}}_{X \sim \mathbb{P}} \left[\left(\frac{\mathbb{Q}(X)}{\mathbb{P}(X)} \right)^{\alpha} \right].
$$

Lemma 6 ([1, Theorem 8])**.** *Let X denote a random variable taking values in Ω, and let $\varphi : \Omega \to \mathbb{R}$ denote a measurable function. Then, for any $\alpha > 1$, and any two distributions, $\mathbb{P}$ and $\mathbb{Q}$, on Ω,*
$$
\frac{\alpha}{\alpha - 1} \ln \mathop{\mathbb{E}}_{X \sim \mathbb{Q}} [\varphi(X)] \leq D_{\alpha}(\mathbb{Q} \| \mathbb{P}) + \ln \mathop{\mathbb{E}}_{X \sim \mathbb{P}} \left[\varphi(X)^{\frac{\alpha}{\alpha - 1}} \right].
\tag{22}
$$

An important special case of Lemma 6 is $\alpha = 2$, in which case
$$
D_2(\mathbb{Q} \| \mathbb{P}) = \ln \mathop{\mathbb{E}}_{X \sim \mathbb{P}} \left[\left(\frac{\mathbb{Q}(X)}{\mathbb{P}(X)} \right)^2 \right] = \ln \left(\chi^2(\mathbb{Q} \| \mathbb{P}) + 1 \right),
$$

and, taking the exponent of Equation 22,
$$
\mathop{\mathbb{E}}_{X \sim \mathbb{Q}} [\varphi(X)] \leq \sqrt{(\chi^2(\mathbb{Q} \| \mathbb{P}) + 1) \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\varphi(X)^2]}.
$$

Thus, with $X \triangleq \theta$ and $\varphi(X) \triangleq G(S, \theta)$,
$$
G(S, \mathbb{Q}) = \mathop{\mathbb{E}}_{\theta \sim \mathbb{Q}} [G(S, \theta)] \leq \sqrt{(\chi^2(\mathbb{Q} \| \mathbb{P}) + 1) \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [G(S, \theta)^2]}.
$$

Further, since $\mathbb{E}_{\theta \sim \mathbb{P}}[G(S, \theta)^2]$ is a nonnegative function of $S \sim \mathbb{D}^n$, Markov's inequality says that

$$\Pr_{S \sim \mathbb{D}^n} \left\{ \mathbb{E}_{\theta \sim \mathbb{P}} \left[G(S, \theta)^2 \right] \geq \frac{1}{\delta} \mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{\theta \sim \mathbb{P}} \left[G(S, \theta)^2 \right] \right\} \leq \delta.$$

We therefore have that with probability at least $1 - \delta$ over draws of $S \sim \mathbb{D}^n$,

$$G(S, \mathbb{Q}) \leq \sqrt{(\chi^2(\mathbb{Q}\|\mathbb{P}) + 1) \frac{1}{\delta} \mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{\theta \sim \mathbb{P}} [G(S, \theta)^2]}. \tag{23}$$

All that remains is to upper-bound $\mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{\theta \sim \mathbb{P}} \left[G(S, \theta)^2 \right]$, which can be accomplished via pointwise hypothesis stability.

Lemma 7 ([8, Lemma 11])**.** *For any (randomized) learning algorithm, A, and M-bounded loss function, L,*

$$\mathbb{E}_{S \sim \mathbb{D}^n} \left[G(S, \theta)^2 \right] \leq \frac{2M^2}{n} + \frac{12M}{n} \sum_{i=1}^n \mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{z \sim \mathbb{D}} \left[\left| L(A(S, \theta), z_i) - L(A(S^{i,z}, \theta), z_i) \right| \right]. \tag{24}$$

Taking the expectation over $\theta \sim \mathbb{P}$ on both sides of Equation 24, and using the linearity of expectation, we have that

$$\mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{\theta \sim \mathbb{P}} \left[G(S, \theta)^2 \right] \leq \frac{2M^2}{n} + \frac{12M}{n} \sum_{i=1}^n \mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{z \sim \mathbb{D}} \mathbb{E}_{\theta \sim \mathbb{P}} \left[\left| L(A(S, \theta), z_i) - L(A(S^{i,z}, \theta), z_i) \right| \right]$$

$$\leq \frac{2M^2}{n} + \frac{12M}{n} \sum_{i=1}^n \beta_{\mathcal{Z}} = \frac{2M^2}{n} + 12M\beta_{\mathcal{Z}}. \tag{25}$$

The last inequality follows directly from Definition 2. Combining Equations 23 and 25, we obtain Equation 5.

B.3   Proof of Theorem 2

The proof of Theorem 2 requires two technical lemmas: the first is a change of measure inequality based on the KL divergence, attributed to Donsker and Varadhan [6]; the second is an upper bound on the moment-generating function of the generalization error, which we prove herein.

Lemma 8 ([6])**.** *Let X denote a random variable taking values in Ω, and let $\varphi : \Omega \to \mathbb{R}$ denote a measurable function. Then, for any two distributions, $\mathbb{P}$ and $\mathbb{Q}$, on Ω,*

$$\mathbb{E}_{X \sim \mathbb{Q}} [\varphi(X)] \leq D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln \mathbb{E}_{X \sim \mathbb{P}} [\exp(\varphi(X))].$$

Lemma 9. *Fix a product measure, $\mathbb{P}$, on $\Theta = \prod_{t=1}^T \Theta_t$, and suppose A is a $(\beta_{\mathcal{Z}}, \beta_{\Theta})$-uniformly stable with respect to L and $\mathbb{P}$. Then, with*

$$\bar{\beta}_{\mathcal{Z}} = 2\beta_{\mathcal{Z}} + \frac{M}{n}, \tag{26}$$

for any $\epsilon > 0$, the moment-generating function (MGF) of $G(S, \theta)$ satisfies

$$\mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{\theta \sim \mathbb{P}} [\exp(\epsilon\, G(S, \theta))] \leq \exp \left(\frac{\epsilon^2}{8} \left(n\bar{\beta}_{\mathcal{Z}}^2 + 4T\beta_{\Theta}^2 \right) + \epsilon\, \beta_{\mathcal{Z}} \right). \tag{27}$$

Proof. Before we begin, let us pause to recognize that the random variable $G(S, \theta)$ has nonzero mean. This is because the learning algorithm—hence, the loss composed with the learning algorithm—is a non-decomposable function of the training data and hyperparameters. We therefore start by defining a zero-mean random variable,

$$\Phi(S, \theta) \triangleq G(S, \theta) - G(\mathbb{D}, \mathbb{P}),$$

where

$$G(\mathbb{D}, \mathbb{P}) \triangleq \mathbb{E}_{S \sim \mathbb{D}^n} \mathbb{E}_{\theta \sim \mathbb{P}} [G(S, \theta)]$$

14

denotes the expected generalization error over draws of both $S \sim \mathbb{D}^n$ and $\theta \sim \mathbb{P}$. These definitions let us decompose the MGF of $G(S, \theta)$ as

$$\mathop{\mathbb{E}}_{S \sim \mathbb{D}^n} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \left[\exp\left(\epsilon\, G(S, \theta) \right) \right] = \mathop{\mathbb{E}}_{S \sim \mathbb{D}^n} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \left[\exp\left(\epsilon\, \Phi(S, \theta) + \epsilon\, G(\mathbb{D}, \mathbb{P}) \right) \right]$$
$$= \mathop{\mathbb{E}}_{S \sim \mathbb{D}^n} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \left[\exp\left(\epsilon\, \Phi(S, \theta) \right) \right] \exp\left(\epsilon\, G(\mathbb{D}, \mathbb{P}) \right). \tag{28}$$

The second equality uses the fact that $G(\mathbb{D}, \mathbb{P})$ is constant with respect to the outer expectations. We now have that the MGF of $G(S, \theta)$ is the product of two factors: the MGF of $\Phi(S, \theta)$, and a monotonic function of $G(\mathbb{D}, \mathbb{P})$. We will bound these terms separately.

First, we upper-bound $G(\mathbb{D}, \mathbb{P})$ as follows:

$$G(\mathbb{D}, \mathbb{P}) = \mathop{\mathbb{E}}_{S \sim \mathbb{D}^n} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \left[\mathop{\mathbb{E}}_{z \sim \mathbb{D}} [L(A(S, \theta), z)] - \frac{1}{n} \sum_{i=1}^{n} L(A(S, \theta), z_i) \right]$$

$$= \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{S \sim \mathbb{D}^n} \mathop{\mathbb{E}}_{z \sim \mathbb{D}} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} [L(A(S, \theta), z) - L(A(S, \theta), z_i)]$$

$$\leq \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{S \sim \mathbb{D}^n} \mathop{\mathbb{E}}_{z \sim \mathbb{D}} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \left[L(A(S^{i,z}, \theta), z) - L(A(S, \theta), z_i) \right] + \beta_{\mathcal{Z}}$$

$$= 0 + \beta_{\mathcal{Z}}.$$

In the second line, we rearrange the expectations using the linearity of expectation. In the third line, we form a new dataset, $S^{i,z}$, by replacing z_i with z; via Definition 1, the expected difference of losses due to replacement, $\mathbb{E}_{\theta \sim \mathbb{P}}[L(A(S, \theta), z) - L(A(S^{i,z}, \theta), z)]$, is upper-bounded by $\beta_{\mathcal{Z}}$-uniform stability. The last line follows from the fact that each example is i.i.d.; since both S and $S^{i,z}$ are distributed according to $\mathbb{D}^n$, and θ is independent of the datasets, the losses cancel out in expectation. Therefore, using the monotonicity of the exponent, and the fact that ϵ is positive, we have that

$$\exp\left(\epsilon\, G(\mathbb{D}, \mathbb{P}) \right) \leq \exp\left(\epsilon\, \beta_{\mathcal{Z}} \right). \tag{29}$$

We now upper-bound the MGF of $\Phi(S, \theta)$, which involves a somewhat technical proof. To reduce notation, we omit the subscript on expectations. Further, we use the shorthand $z_{i:j} \triangleq (z_i, \ldots, z_j)$ and $\theta_{i:j} \triangleq (\theta_i, \ldots, \theta_j)$ to denote subsequences. (Interpret $z_{1:0}$ and $\theta_{1:0}$ as the empty set.) We start by constructing a Doob martingale as follows:

$$V_i \triangleq \begin{cases} \mathbb{E}[G(S, \theta) \mid z_{1:i}] - \mathbb{E}[G(S, \theta) \mid z_{1:i-1}] & \text{for } i = 1, \ldots, n; \\ \mathbb{E}[G(S, \theta) \mid S, \theta_{1:t}] - \mathbb{E}[G(S, \theta) \mid S, \theta_{1:t-1}] & \text{for } i = n + t, \ t = 1, \ldots, T. \end{cases}$$

Observe that $\mathbb{E}[V_i] = 0$ and $\sum_{i=1}^{n+T} V_i = \Phi(S, \theta)$. Thus, using the *law of total expectation* (alternatively, *law of iterated expectations*, or *tower rule*),

$$\mathbb{E}\left[\exp\left(\epsilon\, \Phi(S, \theta) \right) \right] \leq \left(\prod_{i=1}^{n} \sup_{z_{1:i-1}} \mathbb{E}\left[e^{\epsilon V_i} \mid z_{1:i-1} \right] \right) \left(\prod_{t=1}^{T} \sup_{S, \theta_{1:t-1}} \mathbb{E}\left[e^{\epsilon V_{n+t}} \mid S, \theta_{1:t-1} \right] \right). \tag{30}$$

Each iterate of Equation 30 is the supremum of the MGF for the corresponding martingale variable. We will use *Hoeffding's lemma* [14] to uniformly upper-bound each MGF. Hoeffding's lemma states that, if X is a zero-mean random variable, such that $a \leq X \leq b$ almost surely, then for all $\epsilon \in \mathbb{R}$,

$$\mathbb{E}\left[e^{\epsilon X} \right] \leq \exp\left(\frac{\epsilon^2 (b-a)^2}{8} \right). \tag{31}$$

To apply Hoeffding's lemma to each iterate of Equation 30, it suffices to show that

$$\forall i \in \{1, \ldots, n\}, \exists c_i : \sup V_i - \inf V_i$$
$$= \sup_{\substack{z_{1:i}, z'_{1:i}: \\ z_{1:i-1} = z'_{1:i-1}}} \mathbb{E}[G(S, \theta) \mid z_{1:i}] - \mathbb{E}[G(S', \theta) \mid z'_{1:i}] \leq c_i; \tag{32}$$

$$\text{and} \quad \forall t \in \{1, \ldots, T\}, \exists c_t : \sup V_{n+t} - \inf V_{n+t}$$
$$= \sup_{S} \sup_{\substack{\theta_{1:t}, \theta'_{1:t}: \\ \theta_{1:t-1} = \theta'_{1:t-1}}} \mathbb{E}[G(S, \theta) \mid S, \theta_{1:t}] - \mathbb{E}[G(S, \theta') \mid S, \theta'_{1:t}] \leq c_t. \tag{33}$$

15

The constants c_i and c_t replace $a - b$ in Equation 31.

To prove Equation 32, we use Lemma 4 (since A is $\beta_{\mathcal{Z}}$-uniformly stable) and the independence of examples and hyperparameters. For any $z_{1:i}, z'_{1:i} \in \mathcal{Z}^i : z_{1:i-1} = z'_{1:i-1}$,

$$\mathbb{E}[G(S,\theta)\,|\,z_{1:i}] - \mathbb{E}[G(S',\theta)\,|\,z'_{1:i}] = \sum_{z_{i+1:n}} (G(S,\mathbb{P}) - G(S',\mathbb{P})) \prod_{j=1}^{n-i} \mathbb{D}(z_{i+j}) \leq \bar{\beta}_{\mathcal{Z}}.$$

(For notational simplicity, the expectation over $z_{i+1:n}$ is written as a summation, though $\mathcal{Z}$ need not be a finite set.) To prove Equation 33, we use Lemma 5 (since A is β_{Θ}-uniformly stable) and the independence of hyperparameters. For any $S \in \mathcal{Z}^n$ and $\theta_{1:t}, \theta'_{1:t} \in \prod_{j=1}^t \Theta_j : \theta_{1:t-1} = \theta'_{1:t-1}$,

$$\mathbb{E}[G(S,\theta)\,|\,S,\theta_{1:t}] - \mathbb{E}[G(S,\theta')\,|\,S,\theta'_{1:t}] = \sum_{\theta_{t+1:T}} (G(S,\theta) - G(S,\theta')) \prod_{j=1}^{T-t} \mathbb{P}(\theta_{t+j}) \leq 2\beta_{\Theta}.$$

Thus, applying Hoeffding's lemma (Equation 31) to each iterate of Equation 30—using $c_i = \bar{\beta}_{\mathcal{Z}}$ in Equation 32, and $c_t = 2\beta_{\Theta}$ in Equation 33—we have that

$$\mathbb{E}\left[\exp\left(\epsilon\,\Phi(S,\theta)\right)\right] \leq \left(\prod_{i=1}^n \exp\left(\frac{\epsilon^2 \bar{\beta}_{\mathcal{Z}}^2}{8}\right)\right)\left(\prod_{t=1}^T \exp\left(\frac{\epsilon^2 (2\beta_{\Theta})^2}{8}\right)\right)$$

$$= \exp\left(\frac{\epsilon^2}{8}\left(n\bar{\beta}_{\mathcal{Z}}^2 + 4T\beta_{\Theta}^2\right)\right). \tag{34}$$

Finally, by combining Equations 28, 29 and 34, we establish Equation 27. $\qquad\square$

We are now ready to prove Theorem 2. Let $\epsilon > 0$ denote a free parameter, which we will define later. Via Lemma 8 (with $X \triangleq \theta$ and $\varphi(X) \triangleq \epsilon\,G(S,\theta)$), we have that

$$G(S,\mathbb{Q}) = \frac{1}{\epsilon} \mathbb{E}_{\theta\sim\mathbb{Q}}[\epsilon\,G(S,\theta)] \leq \frac{1}{\epsilon}\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln \mathbb{E}_{\theta\sim\mathbb{P}}\left[\exp\left(\epsilon\,G(S,\theta)\right)\right]\right). \tag{35}$$

By Markov's inequality, with probability at least $1 - \delta$ over draws of $S \sim \mathbb{D}^n$,

$$\mathbb{E}_{\theta\sim\mathbb{P}}\left[\exp\left(\epsilon\,G(S,\theta)\right)\right] \leq \frac{1}{\delta} \mathbb{E}_{S\sim\mathbb{D}^n} \mathbb{E}_{\theta\sim\mathbb{P}}\left[\exp\left(\epsilon\,G(S,\theta)\right)\right]$$

$$\leq \frac{1}{\delta} \exp\left(\frac{\epsilon^2}{8}\left(n\bar{\beta}_{\mathcal{Z}}^2 + 4T\beta_{\Theta}^2\right) + \epsilon\beta_{\mathcal{Z}}\right). \tag{36}$$

The second inequality uses Lemma 9 to upper-bound the MGF of $G(S,\theta)$, with $\bar{\beta}_{\mathcal{Z}}$ defined in Equation 26. Combining Equations 35 and 36, we thus have that with probability at least $1 - \delta$,

$$G(S,\mathbb{Q}) \leq \beta_{\mathcal{Z}} + \frac{1}{\epsilon}\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{1}{\delta}\right) + \frac{\epsilon}{8}\left(n\bar{\beta}_{\mathcal{Z}}^2 + 4T\beta_{\Theta}^2\right). \tag{37}$$

What remains is to optimize ϵ to minimize the bound. Minimizing an expression of the form $a/\epsilon + b\epsilon$ is fairly straightforward; the optimal value for ϵ is $\sqrt{a/b}$. However, if we were to apply this formula to Equation 37, the optimal ϵ would depend on $\mathbb{Q}$ via the KL divergence term. Since we want the bound to hold simultaneously for all $\mathbb{Q}$, we need to define ϵ such that it does not depend on $\mathbb{Q}$. To do so, we construct an infinite sequence of ϵ values; for $i = 0, 1, 2, \ldots$, let

$$\epsilon_i \triangleq 2^i \sqrt{\frac{8\ln\frac{2}{\delta}}{n\bar{\beta}_{\mathcal{Z}}^2 + 4T\beta_{\Theta}^2}}. \tag{38}$$

For each ϵ_i, we assign $\delta_i \triangleq \delta 2^{-(i+1)}$ mass to the probability that Equation 37 does not hold, substituting ϵ_i and δ_i for ϵ and δ, respectively. Thus, by the union bound, with probability at least $1 - \sum_{i=0}^\infty \delta_i = 1 - \delta\sum_{i=0}^\infty 2^{-(i+1)} = 1 - \delta$, all $i = 0, 1, 2, \ldots$ satisfy

$$G(S,\mathbb{Q}) \leq \beta_{\mathcal{Z}} + \frac{1}{\epsilon_i}\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{1}{\delta_i}\right) + \frac{\epsilon_i}{8}\left(n\bar{\beta}_{\mathcal{Z}}^2 + 4T\beta_{\Theta}^2\right).$$

For any $\mathbb{Q}$, we select the optimal index, $i^\star$, as

$$i^\star = \left\lfloor \frac{1}{2\ln 2} \ln\left(\frac{D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P})}{\ln(2/\delta)} + 1 \right) \right\rfloor .$$

Since, with a bit of arithmetic,

$$\frac{1}{2}\sqrt{\frac{D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P})}{\ln(2/\delta)} + 1} \le 2^{i^\star} \le \sqrt{\frac{D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P})}{\ln(2/\delta)} + 1}, \tag{39}$$

combining Equations 38 and 39, we have that

$$\sqrt{\frac{2(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta})}{n\bar\beta_{\mathcal{Z}}^2 + 4T\beta_\Theta^2}} \le \epsilon_{i^\star} \le \sqrt{\frac{8(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta})}{n\bar\beta_{\mathcal{Z}}^2 + 4T\beta_\Theta^2}}.$$

It can also be shown [22] that

$$D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{1}{\delta_{i^\star}} \le \frac{3}{2}\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta} \right).$$

Therefore, with probability at least $1 - \delta$ over draws of $S \sim \mathbb{D}^n$, every posterior, $\mathbb{Q}$, satisfies

$$G(S, \mathbb{Q}) \le \beta_{\mathcal{Z}} + \frac{1}{\epsilon_{i^\star}}\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{1}{\delta_{i^\star}} \right) + \frac{\epsilon_{i^\star}}{8}\left(n\bar\beta_{\mathcal{Z}}^2 + 4T\beta_\Theta^2 \right)$$

$$\le \beta_{\mathcal{Z}} + \sqrt{\frac{n\bar\beta_{\mathcal{Z}}^2 + 4T\beta_\Theta^2}{2(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta})} \cdot \frac{3}{2}\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta} \right)}$$

$$+ \sqrt{\frac{8(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta})}{n\bar\beta_{\mathcal{Z}}^2 + 4T\beta_\Theta^2} \cdot \frac{n\bar\beta_{\mathcal{Z}}^2 + 4T\beta_\Theta^2}{8}}$$

$$= \beta_{\mathcal{Z}} + \sqrt{2\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta} \right)\left(n\bar\beta_{\mathcal{Z}}^2 + 4T\beta_\Theta^2 \right)}.$$

Substituting Equation 26 for $\bar\beta_{\mathcal{Z}}$, we obtain Equation 6.

B.4 Proof of Theorem 3

To accommodate all posteriors that might arise from drawing $S \sim \mathbb{D}^n$, it helps to consider $\mathbb{Q}$ as a function of S. Accordingly, we let $\mathbb{Q}_S$ denote the distribution induced by S. With $\delta_1 \triangleq \delta/2$, let

$$E_1(S) \triangleq \mathbb{1}\left\{ \exists \mathbb{Q} : G(S, \mathbb{Q}) \ge +\beta_{\mathcal{Z}} + \sqrt{2\left(D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) + \ln\frac{2}{\delta_1} \right)\left(\frac{(M + 2n\beta_{\mathcal{Z}})^2}{n} + 4T\beta_\Theta^2 \right)} \right\}$$

denote the event that there exists a posterior for which Equation 6 does not hold. With $\delta_2 \triangleq \delta/2$, let

$$E_2(S, \theta) \triangleq \mathbb{1}\left\{ G(S, \theta) \ge G(S, \mathbb{Q}_S) + \beta_\Theta \sqrt{2T\ln\frac{1}{\delta_2}} \right\}$$

denote the event that the generalization error for a given θ exceeds the expected generalization error under the posterior $\mathbb{Q}_S$ by more than $\beta_\Theta \sqrt{2T\ln\frac{1}{\delta_2}}$.

The probability we want to upper-bound is

$$\Pr_{\substack{S\sim\mathbb{D}^n \\ \theta\sim\mathbb{Q}_S}} \{ E_1(S) \vee E_2(S, \theta) \} \le \Pr_{S\sim\mathbb{D}^n} \{ E_1(S) \} + \Pr_{\substack{S\sim\mathbb{D}^n \\ \theta\sim\mathbb{Q}_S}} \{ E_2(S, \theta) \}$$

$$\le \Pr_{S\sim\mathbb{D}^n} \{ E_1(S) \} + \sup_{S\in\mathcal{Z}^n} \Pr_{\theta\sim\mathbb{Q}_S} \{ E_2(S, \theta) \,|\, S \}.$$

The first inequality follows from the union bound; the second inequality follows from probability theory. By Theorem 2, $\Pr_{S\sim\mathbb{D}^n}\{E_1(S)\} \le \delta_1$. To upper-bound $\Pr_{\theta\sim\mathbb{Q}_S}\{E_2(S, \theta)\,|\,S\}$, it suffices to show that $G(S, \theta)$ concentrates tightly around $G(S, \mathbb{Q}_S)$. We will do so with *McDiarmid's inequality* [24]. The following is a specialized version of the theorem.

Lemma 10 ([24]). *Let $X_1, \ldots, X_n$ denote i.i.d. random variables, each taking values in Ω. Suppose $\varphi : \Omega^n \to \mathbb{R}$ is a measurable function for which there exists a constant, β, such that*

$$\sup_{\omega_1, \ldots, \omega_n \in \Omega^n} \sup_{\omega_i' \in \Omega} |\varphi(\omega_1, \ldots, \omega_i, \ldots, \omega_n) - \varphi(\omega_1, \ldots, \omega_i', \ldots, \omega_n)| \leq \beta. \tag{40}$$

Then, for any $\epsilon > 0$,

$$\Pr\{\varphi(X) - \mathbb{E}\,\varphi(X) \geq \epsilon\} \leq \exp\left(\frac{-2\epsilon^2}{n\beta^2}\right). \tag{41}$$

An important special case is when $\beta = \Theta(n^{-1})$, in which case Equation 41 is $\mathrm{O}(\exp(-2n\epsilon^2))$, which decays rapidly.

Recall that A is β_Θ-uniformly stable with respect to L, independent of the posterior. Remember also that, by Lemma 5, G satisfies McDiarmid's stability condition (Equation 40) with $\beta \triangleq 2\beta_\Theta$. Since $\mathbb{Q}_S$ is a product measure, we can therefore apply McDiarmid's inequality; with $\epsilon \triangleq \beta_\Theta \sqrt{2\,T \ln \frac{1}{\delta_2}}$,

$$\Pr_{\theta \sim \mathbb{Q}_S}\{E_2(S, \theta) \,|\, S\} \leq \exp\left(\frac{-2\left(\beta_\Theta \sqrt{2\,T \ln \frac{1}{\delta_2}}\right)^2}{T\,(2\beta_\Theta)^2}\right) = \delta_2.$$

Thus,

$$\Pr_{\substack{S \sim \mathbb{D}^n \\ \theta \sim \mathbb{Q}_S}}\{E_1(S) \vee E_2(S, \theta)\} \leq \delta_1 + \delta_2 = \delta;$$

so, with probability at least $1 - \delta$,

$$G(S, \theta) \leq \beta_\Theta \sqrt{2\,T \ln \frac{1}{\delta_2}} + G(S, \mathbb{Q}_S)$$

$$\leq \beta_\Theta \sqrt{2\,T \ln \frac{1}{\delta_2}} + \beta_{\mathcal{Z}} + \sqrt{2\left(D_{\mathrm{KL}}(\mathbb{Q}_S \| \mathbb{P}) + \ln \frac{2}{\delta_1}\right)\left(\frac{(M + 2n\beta_{\mathcal{Z}})^2}{n} + 4T\beta_\Theta^2\right)}.$$

Replacing δ_1 and δ_2 with $\delta/2$ yields Equation 7.

C Efficient Iteratively Re-weighted Sampling

At each iteration of Algorithm 1, we sample from a categorical distribution on $\{1, \ldots, n\}$, then re-weight the distribution. While sampling from a uniform distribution is trivial, sampling from a nonuniform distribution is complicated. If the distribution is static, sampling can be performed in constant time, with $\mathrm{O}(n)$ initialization time and $\mathrm{O}(n)$ space, using the *alias method* [17]. However, the data structure that enables the alias method cannot be updated in sublinear time, which makes the alias method inefficient for iterative re-weighting when n is large.

In this section, we describe an algorithm for iteratively re-weighted sampling that balances sampling efficiency with re-weighting efficiency. Like the alias method, the algorithm requires $\mathrm{O}(n)$ initialization time and $\mathrm{O}(n)$ space, but the cost of sampling and re-weighting is $\mathrm{O}(\log n)$-time. Even for very large n, logarithmic time is an acceptable iteration complexity—especially since it may pale in comparison to the complexity of updating the hypothesis.

Before training, we initialize a full binary tree of depth $\lceil \log n \rceil$. We label the first n leaves with the initial sampling weights (e.g., for uniform initialization, n^{-1}) and label the remaining $2^{\lceil \log n \rceil} - n$ leaves with 0. We then label each internal node with the sum of its children. During training, we sample from the distribution by performing a random tree traversal: at each internal node visited, we flip a biased coin, whose outcome probabilities are proportional to the labels of the node's children, then move to the corresponding child; the index of the leaf node we arrive at is the sampled value. It is easy to verify that this procedure results in a sample from the distribution. To modify the weight for a given index, we add the change in weight to each node in the path from the root to the associated leaf node. Pseudocode for these procedures is given in Algorithm 2.

Algorithm 2 Efficient Iteratively Re-weighted Sampling

1: **procedure** INITIALIZE($q_1, \ldots, q_n$)
2:     Initialize a full binary tree, $\mathcal{T}$, of depth $\lceil \log n \rceil$
3:     For $i = 1, \ldots, n$, label the ith leaf node with q_i; label the remaining leaf nodes with 0
4:     Label each internal node with the sum of its children's labels.
5: **procedure** SAMPLE($\mathcal{T}$)
6:     $v \leftarrow$ ROOT($\mathcal{T}$)
7:     **while** v is not a leaf **do**
8:         Flip a biased coin, c, with outcome probabilities proportional to the labels of v's children
9:         **if** $c =$ HEADS **then**
10:             $v \leftarrow$ LEFTCHILD
11:         **else**
12:             $v \leftarrow$ RIGHTCHILD
13:     **return** index of leaf node v
14: **procedure** UPDATE($\mathcal{T}, i, q$)
15:     $\Delta \leftarrow q - q_i$
16:     **for** node v on the path from the root to the ith leaf node **do**
17:         Add Δ to the label of v

D Proofs from Section 5

D.1 Proof of Theorem 4

Observe that the KL divergence decomposes as

$$D_{\text{KL}}(\mathbb{Q}\|\mathbb{P}) = \mathop{\mathbb{E}}_{(i_1,\ldots,i_T)\sim\mathbb{Q}} \left[\ln \left(\frac{\mathbb{Q}(i_1,\ldots,i_T)}{\mathbb{P}(i_1,\ldots,i_T)} \right) \right] = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{(i_1,\ldots,i_t)\sim\mathbb{Q}} \left[\ln \left(\frac{\mathbb{Q}_t(i_t)}{\mathbb{P}_t(i_t)} \right) \right], \qquad (42)$$

where $\mathbb{Q}_t(i) = \mathbb{Q}(i_t = i \,|\, i_1, \ldots, i_{t-1})$ is the conditional posterior at iteration t, and $\mathbb{P}_t(i)$, the conditional prior, is simply a uniform distribution on $\{1, \ldots, n\}$. In the first iteration, $\mathbb{Q}_1(i) = \mathbb{P}_1(i)$, since the sampling weights are initialized uniformly to 1. Then, for every $t \geq 2$,

$$\ln \left(\frac{\mathbb{Q}_t(i_t)}{\mathbb{P}_t(i_t)} \right) = \ln \left(\frac{q_{i_t}^{(t)} / \sum_{i=1}^{n} q_i^{(t)}}{n^{-1}} \right) = \ln q_{i_t}^{(t)} - \ln \left(\frac{1}{n} \sum_{i=1}^{n} q_i^{(t)} \right), \qquad (43)$$

where $q_i^{(t)}$ denotes the state of q_i at the *start* of the tth iteration. Unrolling the recursive definition of $q_i^{(t)}$, we have

$$\ln q_i^{(t)} = \ln \prod_{j=1}^{N_{i,t}} \exp \left(\alpha\, f(z_i, h_{O_{i,j}})\, \tau^{N_{i,t}-j} \right) = \alpha \sum_{j=1}^{N_{i,t}} f(z_i, h_{O_{i,j}})\, \tau^{N_{i,t}-j}. \qquad (44)$$

Further, using Jensen's inequality and the concavity of the logarithm,

$$\ln \left(\frac{1}{n} \sum_{i=1}^{n} q_i^{(t)} \right) = \ln \left(\frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{N_{i,t}} \exp \left(\alpha\, f(z_i, h_{O_{i,j}})\, \tau^{N_{i,t}-j} \right) \right)$$

$$\geq \frac{1}{n} \sum_{i=1}^{n} \ln \left(\prod_{j=1}^{N_{i,t}} \exp \left(\alpha\, f(z_i, h_{O_{i,j}})\, \tau^{N_{i,t}-j} \right) \right)$$

$$= \frac{\alpha}{n} \sum_{i=1}^{n} \sum_{j=1}^{N_{i,t}} f(z_i, h_{O_{i,j}})\, \tau^{N_{i,t}-j}. \qquad (45)$$

Combining Equations 42 to 45, we have

$$D_{\text{KL}}(\mathbb{Q}\|\mathbb{P}) \leq \sum_{t=2}^{T} \mathop{\mathbb{E}}_{(i_1,\ldots,i_t)\sim\mathbb{Q}} \left[\alpha \sum_{j=1}^{N_{i_t,t}} f(z_{i_t}, h_{O_{i_t,j}})\, \tau^{N_{i_t,t}-j} - \frac{\alpha}{n} \sum_{i=1}^{n} \sum_{k=1}^{N_{i,t}} f(z_i, h_{O_{i,k}})\, \tau^{N_{i,t}-k} \right].$$

We then reorder the summations to obtain Equation 9.

D.2 Proof of Theorem 5

First, observe that the lower bound in Equation 45 is nonnegative, due to the nonnegativity of the utility function, amplitude and decay. We can therefore drop $\ln\left(\frac{1}{n}\sum_{i=1}^{n} q_i^{(t)}\right)$ from Equation 43, which yields the following upper bound:

$$D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) \leq \mathop{\mathbb{E}}_{(i_1,\ldots,i_T)\sim\mathbb{Q}}\left[\sum_{t=2}^{T}\ln q_{i_t}^{(t)}\right] = \mathop{\mathbb{E}}_{(i_1,\ldots,i_T)\sim\mathbb{Q}}\left[\alpha\sum_{t=2}^{T}\sum_{j=1}^{N_{i_t,t}} f(z_{i_t}, h_{O_{i_t,j}})\tau^{N_{i_t,t}-j}\right]. \quad (46)$$

Since $i_t = i_{O_{i_t,j}}$ for all $j \in N_{i_t,t}$, we have that

$$f(z_{i_t}, h_{O_{i_t,j}}) = f(z_{i_{O_{i_t,j}}}, h_{O_{i_t,j}}) = f(z_{i_{t'}}, h_{t'})$$

for every $t' < t : i_t = i_{t'}$. Thus, the tth computed utility value, $f(z_{i_t}, h_t)$, is referenced whenever the same index is sampled in future iterations. We can therefore reorder the above summations as

$$\sum_{t=2}^{T}\sum_{j=1}^{N_{i_t,t}} f(z_{i_t}, h_{O_{i_t,j}})\tau^{N_{i_t,t}-j} = \sum_{t=1}^{T-1} f(z_{i_t}, h_t)\sum_{j=1}^{N_{i_t,T+1}-N_{i_t,t+1}}\tau^{j-1}. \quad (47)$$

Note that when $N_{i_t,T+1} - N_{i_t,t+1} = 0$ (i.e., when i_t is not sampled again in iterations $t+1, \ldots, T$), the inner summation evaluates to zero. Since the utility function and amplitude are nonnegative, adding a term for i_t that never appears again can only increase the bound. Thus, we can simplify the above expression by extending the inner summation to an infinite series:

$$\sum_{j=1}^{N_{i_t,T+1}-N_{i_t,t+1}}\tau^{j-1} \leq \sum_{j=0}^{\infty}\tau^j \leq \frac{1}{1-\tau}. \quad (48)$$

The last inequality follows from the geometric series identity, since $\tau \in (0,1)$. Combining Equations 46 to 48 yields Equation 10.

References

[1] L. Bégin, P. Germain, F. Laviolette, and J.-F. Roy. PAC-Bayesian bounds based on the Rényi divergence. In *Artificial Intelligence and Statistics*, 2016.

[2] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Neural Information Processing Systems*, 2008.

[3] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[4] O. Catoni. *Pac-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *Institute of Mathematical Statistics Lecture Notes – Monograph Series*. Institute of Mathematical Statistics, 2007.

[5] M. Collins, A. Globerson, T. Koo, X. Carreras, and P. Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *Journal of Machine Learning Research*, 9:1775–1822, 2008.

[6] M. Donsker and S. Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. *Communications on Pure and Applied Mathematics*, 28(1):1–47, 1975.

[7] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[8] A. Elisseeff, T. Evgeniou, and M. Pontil. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6:55–79, 2005.

[9] J. Feng, T. Zahavy, B. Kang, H. Xu, and S. Mannor. Ensemble robustness of deep learning algorithms. *CoRR*, abs/1602.02389, 2016.

[10] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, 1995.

[11] P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian learning of linear classifiers. In *International Conference on Machine Learning*, 2009.

[12] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. *CoRR*, abs/1509.01240, 2015.

[13] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, 2016.

[14] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[15] A. Kontorovich. Concentration in unbounded metric spaces and algorithmic stability. In *International Conference on Machine Learning*, 2014.

[16] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[17] R. Kronmal and A. Peterson. On the alias method for generating random variables from a discrete distribution. *The American Statistician*, 33(4):214–218, 1979.

[18] I. Kuzborskij and C. Lampert. Data-dependent stability of stochastic gradient descent. *CoRR*, abs/1703.01678v2, 2017.

[19] J. Langford and J. Shawe-Taylor. PAC-Bayes and margins. In *Neural Information Processing Systems*, 2002.

[20] J. Lin and L. Rosasco. Optimal learning for multi-pass stochastic gradient methods. In *Neural Information Processing Systems*, 2016.

[21] J. Lin, R. Camoriano, and L. Rosasco. Generalization properties and implicit regularization for multiple passes SGM. In *International Conference on Machine Learning*, 2016.

[22] B. London, B. Huang, and L. Getoor. Stability and generalization in structured prediction. *Journal of Machine Learning Research*, 17(222):1–52, 2016.

[23] D. McAllester. PAC-Bayesian model averaging. In *Computational Learning Theory*, 1999.

[24] C. McDiarmid. On the method of bounded differences. *Surveys in Combinatorics*, 141:148–188, 1989.

[25] L. Rosasco and S. Villa. Learning with incremental iterative regularization. In *Neural Information Processing Systems*, 2015.

[26] M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

[27] S. Shalev-Shwartz. Selfieboost: A boosting algorithm for deep learning. *CoRR*, abs/1411.3436, 2014.

[28] S. Shalev-Shwartz and Y. Wexler. Minimizing the maximal loss: How and why. In *International Conference on Machine Learning*, 2016.

[29] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

[30] Y. Wang, J. Lei, and S. Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of ERM principle. *Journal of Machine Learning Research*, 17(183):1–40, 2016.

[31] P. Zhao and T. Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International Conference on Machine Learning*, 2015.