PAC-BUS: Meta-Learning Bounds via PAC-Bayes and Uniform Stability

Anonymous Author(s)
Affiliation
Address
email

Abstract

We are motivated by the problem of providing strong generalization guarantees in the context of meta-learning. Existing generalization bounds are either challenging to evaluate or provide vacuous guarantees in even relatively simple settings. We derive a probably approximately correct (PAC) bound for gradient-based meta-learning using two different generalization frameworks in order to deal with the qualitatively different challenges of generalization at the "base" and "meta" levels. We employ bounds for uniformly stable algorithms at the base level and bounds from the PAC-Bayes framework at the meta level. The result is a novel PAC-bound that is tighter when the base learner adapts quickly, which is precisely the goal of meta-learning. We show that our bound provides a tighter guarantee than other bounds on a toy non-convex problem on the unit sphere and a text-based classification example. We also present a practical regularization scheme motivated by the bound in settings where the bound is loose and demonstrate improved performance over baseline techniques.

1 Introduction

A major challenge with current machine learning systems is the need to acquire large amounts of training data in order to learn a new task. Over the past few decades, meta-learning [48, 54] has emerged as a promising avenue for addressing this challenge. Meta-learning relies on the intuition that a new task often bears significant similarity to previous tasks; hence, one can learn to perform a new task very quickly by exploiting data from previously-encountered related tasks. The meta-learning problem formulation thus assumes access to datasets from a variety of tasks during meta-training. The goal of the meta-learner is then to learn inductive biases from these tasks in order to train a base-learner to achieve few-shot generalization on a new task.

Over the past few years, there has been tremendous progress in practical algorithms for meta-learning (see, e.g., [47, 42, 18, 25]). Techniques such as model-agnostic meta-learning (MAML) [18] have demonstrated the ability to perform few-shot learning in a variety of supervised learning and reinforcement learning domains. However, our theoretical understanding of these techniques lags significantly behind successes on the empirical front. In particular, the problem of deriving *generalization bounds* for meta-learning techniques remains an outstanding challenge. Current methods for obtaining generalization guarantees for meta-learning [3, 27, 58] either (i) produce bounds that are extremely challenging to compute or (ii) produce vacuous or near-vacuous bounds in even highly simplified settings (see Section 5 for numerical examples). Indeed, we note that existing work on generalization theory for meta-learning techniques do not explicitly report numerical values for generalization bounds. This is in contrast to the state of generalization theory in the supervised learning setting, where recent techniques demonstrate the ability to obtain non-vacuous generalization guarantees on realistic problems (e.g. ImageNet-scale visual classification problems [17, 59, 41]).

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

The generalization challenge in meta-learning is similar to, but distinct from, the supervised learning case. In particular, any generalization bound for meta-learning must account for *two levels* of generalization. First, one must account for generalization at the base level, i.e., the ability of the base learner to perform well on new data from a given task. This is particularly important in the few-shot learning setting. Second, one must account for generalization at the meta level, i.e., the ability of the meta learner to generalize to new tasks not encountered during meta-training. Moreover, the generalization performance at the two levels is coupled since the meta learner is responsible for learning inductive biases that the base learner can exploit for future tasks.

The key technical insight of this work is to bound the generalization error at the two levels (base and meta) using two *different* generalization theory frameworks that each are particularly well-suited for addressing the specific challenges of generalization. At the base level, we utilize the fact that a learning algorithm that exhibits uniform stability [11] also generalizes well in expectation (see Section 4.1 for a formal statement). Intuitively, uniform stability quantifies the sensitivity of the output of a learning algorithm to changes in the training dataset. As demonstrated by [22], limiting the number of training epochs of a gradient-based learning algorithm leads to uniform stability. In other words, a gradient-based algorithm that *learns quickly* is uniformly stable. Since the goal of meta-learning is *precisely* to train the base learner to learn quickly, we posit that generalization bounds based on uniform stability are particularly well-suited to bounding the generalization error at the base level. At the meta level, we employ a generalization bound based on *Probably Approximately Correct (PAC)-Bayes* theory. Originally developed two decades ago [34, 31], there has been a recent resurgence of interest in PAC-Bayes due to its ability to provide strong generalization guarantees for neural networks [17, 5, 41]. Intuitively, the challenge of generalization at the meta level (i.e., generalizing to new tasks) is similar to the challenge of generalizing to new data in the standard supervised learning setting. In both cases, one must prevent over-fitting to the particular tasks/data one has seen during meta-training/training. Thus, the strong empirical performance of PAC-Bayes theory in supervised learning problems makes it a promising candidate for bounding the generalization error at the meta level.

Contributions. The primary contributions of this work are the following. First, we leverage the insights above in order to develop a novel generalization bound for gradient-based meta-learning using uniform stability and PAC-Bayes theory (Theorem 3). Second, we develop a regularization scheme for MAML [18] that explicitly minimizes the derived bound (Algorithm 1). We refer to the resulting approach as *PAC-BUS* (generalization guarantees for meta-learning via PAC-Bayes and Uniform Stability). Third, we demonstrate our approach on two meta-learning problems: (i) a toy non-convex classification problem on the unit-ball (Section 5.1), and (ii) the *Mini-Wiki* benchmark introduced in [27] (Section 5.2). Even in these relatively small-scale settings, we demonstrate that recently-developed generalization frameworks for meta-learning provide either near-vacuous or loose bounds, while PAC-BUS provides significantly stronger bounds. Fourth, we demonstrate our approach in larger-scale settings where it remains challenging to obtain non-vacuous bounds (for our approach as well as others). Here, we propose a practical regularization scheme which re-weights the terms in the rigorously-derived PAC-BUS upper bound (*PAC-BUS(H)*; Algorithm A.2 in the appendix). Recent work [58] introduces a challenging variant of the *Omniglot* benchmark [28] which highlights and tackles challenges with *memorization* in meta-learning. We show that PAC-BUS(H) is able to prevent memorization on this variant (Section 5.3).

2 Problem formulation

Samples, tasks, and datasets. Formally, consider the setting where we have an unknown meta distribution $\mathcal{T}$ over tasks (roughly, "tasks" correspond to different, but potentially related, learning problems). A sampled task $t \sim \mathcal{T}$ induces an (unknown) distribution $\mathcal{Z}_t$ over sample space Z. We assume that all sampling is independent and identically distributed (i.i.d.). Note that the sample space Z is shared between tasks, but the distribution $\mathcal{Z}_t$ may be different. We then sample within-task samples $z \sim \mathcal{Z}_t$ and within-task datasets $S = \{z_1, z_2, \ldots, z_m\} \sim \mathcal{Z}_t^m$. We assume that each sample z has a single corresponding label $o(z)$, where function o is an oracle which outputs the correct label of z. At meta-training time, we assume access to l datasets, which we call $\mathbf{S} = \{S_1, S_2, \ldots, S_l\}$. Each dataset S_i in $\mathbf{S}$ is drawn by first selecting a task t_i from $\mathcal{T}$, and then drawing $S_i \sim \mathcal{Z}_{t_i}^m$.

Hypotheses and losses. Let h denote a hypothesis and $L(h, z)$ be the cost incurred by hypothesis h on sample z. The loss is computed by comparing $h(z)$ with the true label $o(z)$. For simplicity, we assume that there is no noise on the labels; we can thus assume that all loss functions have access to

the label oracle function o and thus the loss depends only on hypothesis h and sample z. We note that this assumption is not required for our analysis and is made for the ease of exposition. Overloading the notation, we let $L(h, \mathcal{Z}_t) := \mathbb{E}_{z \sim \mathcal{Z}_t} L(h, z)$ and $L(h, S) := \frac{1}{|S|} \sum_{i=1}^{|S|} L(h, z_i)$.

Meta-learning. As with model-agnostic meta-learning (MAML) [18], we let meta parameters $\theta \in \mathbb{R}^{n_\theta}$ correspond to an initialization of the base learner's hypothesis. Let h_θ be the θ-initialized hypothesis. Generally, the initialization θ is learned from the multiple datasets we have access to at meta-training time. In this work, we will learn a *distribution* Θ over initializations so that we can use bounds from the PAC-Bayes framework. At test time, a new task $t \sim \mathcal{T}$ is sampled and we are provided with a new dataset $S \sim \mathcal{Z}_t^m$. The base learner uses an algorithm A (e.g., gradient descent), the dataset S, and the initialization $\theta \sim \Theta$ in order to fine-tune the hypothesis and perform well on future samples drawn from $\mathcal{Z}_t$. We denote the base learner's updated hypothesis by $h_{A(\theta,S)}$. More formally, our goal is to learn a distribution Θ with the following objective:

$$\min_{\Theta} \mathcal{L}(\Theta, \mathcal{T}) := \min_{\Theta} \mathop{\mathbb{E}}_{t \sim \mathcal{T}} \mathop{\mathbb{E}}_{S \sim \mathcal{Z}_t^m} \mathop{\mathbb{E}}_{\theta \sim \Theta} L(h_{A(\theta,S)}, \mathcal{Z}_t). \tag{1}$$

We are particularly interested in the the few-shot learning case, where the number of samples which the base learner can use to adapt is small. A common technique to improve test performance in the few-shot learning case is to allow for validation data at meta-training time. Thus, in addition to a generalization guarantee on meta-learning without validation data, we will derive a bound when allowing for the use of validation data $S_{\text{va}} \sim \mathcal{Z}_t^n$ during meta-training.

3 Related work

Meta-learning. Meta-learning is a well-studied technique for exploiting similarities between learning tasks [48, 54]. Often used to reduce the need for large amounts of training data, a number of approaches for meta-learning have been explored over decades [8, 10, 12, 24, 56, 47, 42, 25]. Recently, methods based on model agnostic meta-learning (MAML) [18] have demonstrated strong performance across different application domains and benchmarks such as *Omniglot* [28] and *Mini-ImageNet* [57]. These methods operate by optimizing a set of initial parameters that can be quickly fine-tuned via gradient descent on a new task. The approaches mentioned above typically do not provide any generalization guarantees, and none of them compute explicit numerical bounds on generalization performance. Our approach has the structure of gradient-based meta-learning while providing guarantees on generalization.

Generalization bounds for supervised learning. Multiple frameworks have been developed for providing generalization guarantees in the classical supervised learning setting. Early breakthroughs include Vapnik-Chervonenkis (VC) theory [55, 4], Rademacher complexity [50], and the minimum description length principle [9, 43, 29]. More recent frameworks include Probably Approximately Correct (PAC)-Bayes theory [51, 34, 49] and algorithmic stability bounds [11, 15, 22, 44, 1]. PAC-Bayes theory in particular provides some of the tightest known generalization bounds for classical supervised learning approaches such as support vector machines [49, 31, 19]. Since its development, researchers have continued to tighten [31, 33, 45] and generalize the framework [13, 14]. Exciting recent results [17, 35, 36, 7, 5, 41] have demonstrated the promise of PAC-Bayes to provide strong generalization bounds for neural networks on supervised learning problems (see [26] for a recent review of generalization bounds for neural networks). In contrast to the standard supervised learning setting, generalization bounds for meta-learning are less common and remain loose.

Generalization bounds for meta-learning. As described in Section 1, meta-learning bounds must account for two "levels" of generalization (base level and meta level). The approach presented in [32] utilizes algorithmic stability bounds at both levels. However, this requires both meta and base learners to be uniformly stable. This is a strong requirement that is challenging to ensure at the meta level. Another recent method, known as follow-the-meta-regularized-leader (FMRL) [27], provides guarantees for a regularized meta-learning version of the follow-the-leader (FTL) method for online learning [23]. The generalization bounds provided are derived from the application of online-to-batch techniques [2, 16]. A regret bound for meta-learning using an aggregation technique at the meta-level and an algorithm with a uniform generalization bound at the base level is provided in [2]. The techniques mentioned do not present an algorithm which makes use of validation data (in contrast to our approach). Using validation data (i.e., held-out data) is a common technique for improving performance in meta-learning and is particularly important for the few-shot learning case.

Another method for creating a generalization bound on meta-learning is to use PAC-Bayes bounds at both the base and meta levels [39, 40]. In [3], generalization bounds based on such a framework are provided along with practical optimization techniques. However, the method requires one to maintain distributions over distributions of initializations, which can result in large computation times during training and makes it extremely challenging to numerically compute the bound. Moreover, the approach also does not allow one to incorporate validation data to improve the bound. Recent work has made progress on some of these challenges. In [46], the computational efficiency of training is improved but the challenges associated with numerically computing the generalization bound or incorporating validation data are not addressed. State-of-the-art work tightens the two-level PAC-Bayes guarantee, addresses computation times for training and evaluation of the bound, and allows for validation data [58]. However, all of the two-level PAC-Bayes bounds require a separate PAC-Bayes bound for each task, and thus a potentially loose union bound.

We present a framework which, to our knowledge, is the first to combine algorithmic stability and PAC-Bayes bounds (at the base- and meta- levels respectively) in order to derive a meta-learning algorithm with associated generalization guarantees. As outlined in Section 1, we believe that the algorithmic stability and PAC-Bayes frameworks are particularly well-suited to tackling the specific challenges of generalization at the different levels. We also highlight that *none* of the approaches mentioned above report numerical values for generalization bounds, even for relatively simple problems. Here, we empirically demonstrate that prior approaches tend to provide either near-vacuous or loose bounds even in relatively small-scale settings while our proposed method provides significantly stronger bounds.

4 Generalization bound on meta-learning

We use two different frameworks for the two levels of generalization required in a meta-learning bound. We utilize the PAC-Bayes framework to bound the expected training loss on future tasks, and uniform stability bounds to argue that if we have a low training loss when using a uniformly stable algorithm, then we achieve a low test loss. The following section will introduce these frameworks independently. We then present the overall meta-learning bound and associated algorithm to find a distribution over initialization parameters (i.e., meta parameters) that minimizes the upper bound.

4.1 Preliminaries: two generalization frameworks

4.1.1 Uniform stability

Let $S = \{z_1, z_2, \ldots, z_m\} \in Z^m$ be a set of m samples. Let S^i be identical to dataset S except that the ith sample z_i is replaced by some $z_i' \in Z$. Note that our analysis can be extended to allow for losses bounded by some finite M, but we work with losses bounded within $[0, 1]$ for the sake of simplicity. With these precursors, we define the following notion of *uniform stability* [11] for an algorithm A.

Definition 1 (Uniform Stability [11]) *Algorithm A has β_{US}-differ-by-one (β_{US}-DBO) uniform stability with respect to loss L if for a given task t, $\forall\, z \in Z$, $\forall\, S \in Z^m$, $\forall\, i \in \{1, \ldots, m\}, \forall\, \theta \in \mathbb{R}^{n_\theta}$, the following holds:*

$$|L(h_{A(\theta,S)}, z) - L(h_{A(\theta,S^i)}, z)| \leq \beta_{\text{US}}. \tag{2}$$

The following result establishes a relationship between uniform stability and generalization in expectation.

Theorem 1 (Algorithmic Stability Generalization in Expectation [22]) *Fix a task $t \in \mathcal{T}$. The following inequality holds for hypothesis $h_{A(\theta,S)}$ learned using β_{US}-DBO uniformly stable algorithm A with respect to loss L:*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{Z}_t^m} L(h_{A(\theta,S)}, \mathcal{Z}_t) \leq \mathop{\mathbb{E}}_{S\sim\mathcal{Z}_t^m} L(h_{A(\theta,S)}, S) + \beta_{\text{US}}. \tag{3}$$

Since we work with distributions over initializations, we replace the loss with the expected loss over $\theta \sim \Theta$. We also take the expectation over $t \sim \mathcal{T}$ so that the guarantee is on $\mathcal{L}(\Theta, \mathcal{T})$ from Equation (1):

$$\mathcal{L}(\Theta, \mathcal{T}) = \mathop{\mathbb{E}}_{t\sim\mathcal{T}} \mathop{\mathbb{E}}_{S\sim\mathcal{Z}_t^m} \mathop{\mathbb{E}}_{\theta\sim\Theta} L(h_{A(\theta,S)}, \mathcal{Z}_t) \leq \mathop{\mathbb{E}}_{t\sim\mathcal{T}} \mathop{\mathbb{E}}_{S\sim\mathcal{Z}_t^m} \mathop{\mathbb{E}}_{\theta\sim\Theta} L(h_{A(\theta,S)}, S) + \beta_{\text{US}}. \tag{4}$$

4.1.2 PAC-Bayes theory

For the meta level bound, we make use of the PAC-Bayes generalization bound introduced in [34]. Note that other PAC-Bayes bounds such as the quadratic variant [45] and PAC-Bayes-λ variant [53] may be used and substituted in the following analysis. We first present a general version of the PAC-Bayes bound and then specialize it to our meta-learning setting. Let $f(\theta, s)$ be an arbitrary loss function which only depends on parameters θ and the sample s from an arbitrary distribution $\mathcal{D}$.

Theorem 2 (PAC-Bayes Generalization Bound [34]) *For any data-independent prior distribution Θ_0 over θ, some loss function f where $0 \leq f(\theta, s) \leq 1, \forall\, s, \forall\, \theta$, and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over a sampling of $\{s_1, s_2, \ldots, s_l\} \sim \mathcal{D}^l$, the following holds for all distributions Θ over θ:*

$$\mathop{\mathbb{E}}_{s\sim\mathcal{D}} \mathop{\mathbb{E}}_{\theta\sim\Theta} f(\theta, s) \leq \frac{1}{l} \sum_{i=1}^{l} \mathop{\mathbb{E}}_{\theta\sim\Theta} f(\theta, s_i) + R_{\mathrm{PAC-B}}(\Theta, \Theta_0, \delta, l), \tag{5}$$

where the PAC-Bayes regularizer term is defined as follows

$$R_{\mathrm{PAC-B}}(\Theta, \Theta_0, \delta, l) := \sqrt{\frac{D_{\mathrm{KL}}(\Theta \| \Theta_0) + \ln \frac{2\sqrt{l}}{\delta}}{2l}}, \tag{6}$$

and D_{KL} is the Kullback-Leibler (KL) divergence.

We aim to use this generalization guarantee to bound the expected training loss on a dataset S. Thus, we would like sample s to be the training data S for some task. This would allow us to combine the PAC-Bayes generalization guarantee with the guarantee in expectation presented in Inequality (4). However, we have previously only sampled $S \sim \mathcal{Z}_t^m$ where distribution $\mathcal{Z}_t$ is induced by a task $t \sim \mathcal{T}$. We therefore define the marginal distribution $\mathcal{D}_S$ over datasets of size m. Note that sampling $S \sim \mathcal{D}_S$ is equivalent to first sampling $t \sim \mathcal{T}$ and then sampling $S \sim \mathcal{Z}_t^m$. Thus, we have training data $\mathbf{S} = \{S_1, S_2, \ldots, S_l\} \sim \mathcal{D}_S^l$. Let $f(\theta, s) = L(h_{A(\theta,S)}, S)$ and Inequality (5) becomes

$$\begin{aligned}\mathop{\mathbb{E}}_{S\sim\mathcal{D}_S} \mathop{\mathbb{E}}_{\theta\sim\Theta} L(h_{A(\theta,S)}, S) &= \mathop{\mathbb{E}}_{t\sim\mathcal{T}} \mathop{\mathbb{E}}_{S\sim\mathcal{Z}_t^m} \mathop{\mathbb{E}}_{\theta\sim\Theta} L(h_{A(\theta,S)}, S) \\ &\leq \frac{1}{l} \sum_{i=1}^{l} \mathop{\mathbb{E}}_{\theta\sim\Theta} L(h_{A(\theta,S_i)}, S_i) + R_{\mathrm{PAC-B}}(\Theta, \Theta_0, \delta, l),\end{aligned} \tag{7}$$

where $A(\theta, S)$ is *any* algorithm which updates θ using only dataset S.

4.2 Meta-learning bound

In order to obtain a generalization guarantee for meta-learning, we combine the analyses for the two levels of generalization above. To do this, we let algorithm A in Inequality (7) be β_{US}-DBO uniformly stable. We then use Inequality (7) to bound the first term in the RHS of Inequality (4). With the following assumption, the result is stated in Theorem 3.

Assumption 1 (Bounded loss.) *The loss function L is bounded: $0 \leq L(h_{A(\theta,S)}, z) \leq 1, \forall\, z \in Z, \forall\, S \sim \mathcal{D}_S, \forall\, \theta \sim \Theta$.*

Theorem 3 (Meta-Learning Generalization Guarantee) *For hypotheses $h_{A(\theta,S)}$ learned with β_{US}-DBO uniformly stable algorithm A, data-independent prior Θ_0 over initializations θ, loss L which satisfies Assumption 1, and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over a sampling of the meta-training dataset $\mathbf{S} \sim \mathcal{D}_S^l$, the following holds for all distributions Θ over θ:*

$$\mathcal{L}(\Theta, \mathcal{T}) \leq \frac{1}{l} \sum_{i=1}^{l} \mathop{\mathbb{E}}_{\theta\sim\Theta} L(h_{A(\theta,S_i)}, S_i) + R_{\mathrm{PAC-B}}(\Theta, \Theta_0, \delta, l) + \beta_{\mathrm{US}}. \tag{8}$$

Theorem 3 is presented for any distributions Θ and Θ_0 over initializations. However, in practice we will use multivariate Gaussian distributions for both. The specialization of Theorem 3 to Gaussian distributions is provided in Appendix A.1.1. Next, we allow for validation data $S_{\mathrm{va}} \sim \mathcal{Z}_t^n$ at meta-training time so that the bound is more suited to the few-shot learning case. We compute the upper bound using the evaluation data $S_{\mathrm{ev}} = \{S, S_{\mathrm{va}}\}$ sampled from the marginal distribution $\mathcal{D}_{S_{\mathrm{ev}}}$ over

Algorithm 1 PAC-BUS: meta-learning via PAC-Bayes and Uniform Stability

Input: Fixed prior distribution $\mathcal{N}_{\mu_0,s_0}$ over initializations
Input: β_{US}-DBO uniformly stable Algorithm A
Input: Meta-training dataset $\mathbf{S}$, learning rate γ
Initialize: $[\mu, \eta] \leftarrow [\mu_0, \log(s_0)]$
Output: Optimized μ^*, s^*
$B(\mu, s, \theta_1, \theta_2 \ldots, \theta_l) := \frac{1}{l}\sum_{i=1}^{l} L(h_{\theta_i}, S_i) + R_{\text{PAC-B}}(\mathcal{N}_{\mu,s}, \mathcal{N}_{\mu_0,s_0}, \delta, l) + \beta_{\text{US}}$
while not converged **do**
  Sample $\xi \sim \mathcal{N}_{0,I}$ and set $\theta \leftarrow \mu + \sqrt{s} \odot \xi$
  for $i = 1$ **to** l **do**
    $\theta_i \leftarrow A(\theta, S_i)$
  end for
  $[\mu, \eta] \leftarrow [\mu, \eta] - \gamma \nabla_{[\mu,\eta]} B(\mu, \exp(\eta), \theta_1, \theta_2 \ldots, \theta_l)$
  $s \leftarrow \exp(\eta)$
end while

datasets of size $m + n$. However, we still only require m samples at meta-test time; see Appendix A.1.2 for the derivation. Note that the training data S is often excluded from the data used to update the meta-learner. However, this is necessary for our approach to obtain a guarantee on few-shot learning performance. The result is a guarantee with high probability over a sampling of $\mathbf{S}_{\text{ev}} \sim \mathcal{D}^l_{S_{\text{ev}}}$:

$$\mathcal{L}(\Theta, \mathcal{T}) \leq \frac{1}{l}\sum_{i=1}^{l} \mathop{\mathbb{E}}_{\theta\sim\Theta} L(h_{A(\theta,S_i)}, S_{\text{ev},i}) + R_{\text{PAC-B}}(\Theta, \Theta_0, \delta, l) + \frac{m\beta_{\text{US}}}{m+n}. \tag{9}$$

4.3 PAC-BUS algorithm

Recall that we aim to find a distribution Θ over initializations that minimizes $\mathcal{L}(\Theta, \mathcal{T})$ as stated in Equation (1). We cannot minimize $\mathcal{L}(\Theta, \mathcal{T})$ directly due to the expectations taken over unknown distributions $\mathcal{T}$ and $\mathcal{Z}_t$ for sampled task t, but we may indirectly minimize it by minimizing the upper bounds in Inequalities (8) or (9).

Computing the upper bound requires evaluating an expectation taken over $\theta \sim \Theta$. In general, this is intractable. However, we aim to minimize this upper bound to provide the tightest guarantee possible. Similar to the method in [17], we use an unbiased estimator of $\mathbb{E}_{\theta\sim\Theta} L(h_{A(\theta,S)}, \cdot)$. Let Θ be a multivariate Gaussian distribution over initializations θ with $\mu := \Theta_{\text{mean}}$ and $\text{diag}(s) := \Theta_{\text{covar}}$; thus $\Theta = \mathcal{N}_{\mu,s} := \mathcal{N}(\mu, \text{diag}(s))$ and $\Theta_0 = \mathcal{N}_{\mu_0,s_0}$. We use the following estimator of $\mathbb{E}_{\theta\sim\Theta} L(h_{A(\theta,S)}, \cdot)$:

$$L(h_{A(\theta,S)}, \cdot), \quad \theta \leftarrow \mu + \sqrt{s} \odot \xi, \quad \xi \sim \mathcal{N}_{0,I}. \tag{10}$$

We present the resulting training technique in Algorithm 1. This algorithm can be used to learn a distribution over initializations that minimizes the upper bound presented in Theorem 3 and its specializations. This is presented for the case when A is β_{US}-DBO uniformly stable for some β_{US}. For gradient-based algorithms, the learning rate α often appears directly in the bound for β_{US} [22]. Thus it is potentially beneficial to update α as well. We present Algorithm 1 without learning the learning rate; see Appendix A.3.1 for the version which meta-learns the learning rate.

Determining the gradient of $B(\mu, s, \theta_1, \theta_2 \ldots, \theta_l)$ with respect to μ and η requires computing the Hessian of the loss function if algorithm $A(\theta, S)$ uses a gradient update to compute θ_i. First order approximations often perform similarly to the second-order meta-learning techniques [20, 18, 37], and can be used to speed up the training. Additionally, Algorithm 1 can be modified to use mini-batches of tasks instead of all tasks in the meta update to improve training times.

In practice, we are interested in algorithms such as the stochastic gradient method (SGM) (as described in [22]) and gradient descent (GD) for the base learner. We can obtain bounds on the uniform stability constant β_{US} when using gradient methods with the results from [22]. See Appendix A.2 for details on the β_{US} bounds we use in this work. With a bound on β_{US}, we can calculate all the terms in $B(\mu, s, \theta_1, \theta_2 \ldots, \theta_l)$ and use Algorithm 1 to minimize the meta-learning upper bound. When evaluating the upper bound, we use the sample convergence bound [30, 17] to upper bound the expectation taken over $\theta \sim \Theta$. See Appendix A.4 for details.

Table 1: We present the generalization bounds (for $\delta = 0.01$) provided by each method if applicable, and use the sample convergence bound [30] for MR-MAML, and PAC-BUS, but not MLAP-M[1]. Note that for these methods, we specifically minimize their respective meta-learning bounds. We also report the meta-test CEL_s (test loss) for all methods. We present the mean and standard deviation after 5 trials. We highlight that our approach provides the strongest generalization guarantee.

Classification on Ball	MAML [18]	MLAP-M [3]	MR-MAML [58]	PAC-BUS (ours)
Bound	None	1.0538 ± 0.0012[1]	0.3422 ± 0.0006	$\mathbf{0.2213 \pm 0.0012}$
Test Loss	0.1701 ± 0.0070	0.1645 ± 0.0045	$\mathbf{0.1584 \pm 0.0012}$	0.1657 ± 0.0014

5 Examples

We demonstrate our approach on three examples below. In the first two examples, our primary goal is to demonstrate the tightness of our generalization bounds compared to other meta-learning bounds. We also present empirical test performance on held-out data; however, we emphasize that the focus of our work is to obtain improved generalization guarantees (and not necessarily to improve empirical test performance). In the third example, we present an algorithm that is motivated by our theoretical framework and demonstrate its ability to improve empirical performance on a challenging task.

5.1 Example: classification on the unit ball

We evaluate the tightness of the generalization bound in Equation (9) on a toy two-class classification problem where the sample space Z is the unit ball $B^2(0,1)$ in two dimensions with radius 1 and centered at the origin. Data points for each task are sampled from $\mathcal{Z}_t$, where a task corresponds to a particular concept which labels the data as $(+)$ if within $B^2(c_t, r_t)$ and $(-)$ otherwise. Center c_t is sampled uniformly from the $y \geq 0$ semi-ball $B^2_{y\geq 0}(0, 0.4)$ of radius 0.4. The radius r_t is then sampled uniformly from $[0.1, 1 - \|c_t\|]$. Notably, this problem is challenging for losses which are convex in the parameters of the network (since the decision boundary between classes is nonlinear). Thus, generalization bounds which rely on convex losses (such as [27]) will not be able to provide guarantees on networks that perform well. We choose the softmax-activated cross entropy loss, CEL_s, as the loss function. Before running Algorithm 1, we address a few technical challenges that arise from Assumption 1 as well as computing c_L and c_S. We address these in Appendix A.3.

We then apply Algorithm 1 using the few-shot learning bound in Equation (9). We present the guarantee on the meta-test loss associated with each training method in Table 1. In addition, we present the average meta-test loss after training with 10 samples. We compare our bounds and empirical performance with the meta-learning by adjusting priors (MLAP) technique [3] and the meta-regularized MAML (MR-MAML) technique [58]. All methods are given held-out data to learn a prior before minimizing their respective upper bounds. Additionally, networks are constrained such that $\|N(z)\|_F \leq r$ since all bounds require the loss to be within $[0, 1]$. We compare the aforementioned methods' meta-test loss to MAML with weights constrained in the same manner (note that MAML does not provide a guarantee). Upper bounds which use the PAC-Bayes framework are computed with many evaluations from the posterior distribution. This allows us to apply the sample convergence bound [30] (as in Equation (A.13) for our bound) unless otherwise noted.

We find that PAC-BUS provides a significantly stronger guarantee compared with the other methods. Note that the guarantee provided by MLAP-M [3] is vacuous because the meta-test loss is bounded between 0 and 1, while the guarantee is above 1.

5.2 Example: Mini-Wiki

Next, we present results on the *Mini-Wiki* benchmark introduced in [27]. This is derived from the Wiki3029 dataset presented in [6]. The dataset is comprised of 4-class, m-shot learning tasks with sample space $Z = \{z \in \mathbb{R}^d \mid \|z\|_2 = 1\}$. Sentences from various Wikipedia articles are passed through the $d = 50$-dimension continuous-bag-of-words GloVe embedding [38] to generate samples. For this learning task, we use a k-class version of CEL_s and logistic regression. Since this example is

[1] Due to high computation times associated with estimating the MLAP upper bound, this value is not computed with the sample convergence bound as the other upper bounds are. Thus, the value presented does not carry a guarantee, but would be similar if computed with the sample convergence bound. The value is shown to give a qualitative sense of the guarantee.

Table 2: We compare the generalization bounds (for $\delta = 0.01$) provided by each method where applicable and use the sample convergence bound for MR-MAML and PAC-BUS. Since we specifically minimize these methods' upper bounds, we can fairly compare the relative tightness of each bound. We also report the meta-test loss (CEL_s) for each method for exposition. We report the mean and standard deviation after 5 trials. We highlight that our approach provides the strongest guarantee.

4-Way *Mini-Wiki*	1-shot	3-shot	5-shot
FLI-Batch Bound [27]	0.6638 ± 0.0011	0.6366 ± 0.0006	0.6343 ± 0.0014
MR-MAML Bound [58]	0.7400 ± 0.0003	0.7312 ± 0.0003	0.7283 ± 0.0005
PAC-BUS Bound (ours)	$\mathbf{0.4999 \pm 0.0003}$	$\mathbf{0.5058 \pm 0.0002}$	$\mathbf{0.5101 \pm 0.0002}$
MAML [18]	$\mathbf{0.3916 \pm 0.0009}$	$\mathbf{0.3868 \pm 0.0005}$	$\mathbf{0.3883 \pm 0.0005}$
FLI-Batch [27]	0.4091 ± 0.0008	0.4078 ± 0.0005	0.4097 ± 0.0012
MR-MAML [58]	0.3922 ± 0.0009	0.3869 ± 0.0003	0.3884 ± 0.0005
PAC-BUS (ours)	0.3922 ± 0.0009	0.3878 ± 0.0003	0.3895 ± 0.0005

convex, we can use GD and bound β_{US} with Theorem A.1 in the appendix [22]. We keep the loss bounded by constraining the network $\|N(z)\|_F \leq r$ and scale the loss as in the previous example. The tightness of the bounds on c_L and c_S affected the upper bound in Inequality (9) more than in the previous example, so we bound them as tightly as possible. See Appendix A.7 for the calculations.

We apply Algorithm 1 using the bound which allows for validation data, Inequality (9), to learn on 4-way *Mini-Wiki* $m = \{1, 3, 5\}$-shot. The results are presented in Table 2. We compare our results with the FMRL variant which provides a guarantee [27], follow-the-last-iterate (FLI)-Batch, and with MR-MAML [58]. FLI-Batch does not require bounded losses explicitly, but requires that the parameters of the network lie within a ball of radius r. For the logistic regression used in the example, this is equivalent to $\|N(z)\|_F \leq r$. Thus, we scale the loss and use the same r for each method to provide a fair comparison. We also show the results of training with MAML constrained in the same way for reference. Each method is given the same amount of held-out data for training a prior.

As in the previous example, PAC-BUS provides a significantly tighter guarantee than the other methods (Table 2). We see similar empirical meta-test loss for MAML [18], MR-MAML [58], and PAC-BUS with slightly higher loss for FLI-Batch [27]. In addition, we computed the meta-test accuracy as the percentage of correctly classified sentences. See Table A.1 in Section A.8.2 for these results along with other experimental details.

5.3 Example: memorizable Omniglot

We have demonstrated the ability of our approach to provide strong generalization guarantees for meta-learning in the settings above. We now consider a more complex setting where we are unable to obtain strong guarantees. In this example, we employ a learning heuristic based on the PAC-BUS upper bound, *PAC-BUS(H)*; see Appendix A.3.2 for the details and the Algorithm. We relax Assumption 1 and no longer constrain the network as in previous sections. Instead, we maintain and update estimates of the Lipschitz and smoothness constants of the network, using [52], and incorporate them into the uniform stability regularizer term, β_{US}. We then scale each regularizer term (i.e., $R_{\text{PAC-B}}(\Theta, \Theta_0, \delta, l)$ and β_{US}) by hyper-parameters λ_1 and λ_2 respectively. Analogous to the technique described in [58], we aim to incorporate the form of the theoretically-derived regularizer into the loss, without requiring it to be as restrictive during learning. The result is a regularizer that punishes large deviation from the prior Θ_0 and too much adaptation at the base-learning level.

We test our method on *Omniglot* [28] for 20-way, $m = \{1, 5\}$-shot classification in the non-mutually exclusive (NME) case [58]. In [58], the problem of memorization in meta-learning is explored and demonstrated with non-mutually exclusive learning problems. *NME Omniglot* corresponds to randomization of class labels for a task at test time only. This worsens the performance of any network that memorized class labels; see [58] for more details.[2] We compare our method to an analogous heuristic presented in [58], which also has a $D_{\text{KL}}(\Theta \| \Theta_0)$ term in the loss. Thus, this heuristic (referred to as MR-MAML(W) [58]) regularizes the change in weights of the network. Additionally, we compare to the heuristic described in [27] (FLI-Online) which performs better in practice than the FLI-Batch method. We do not provide data for training a prior in this case, as it may cause memorization. We use standard MAML as a reference. See Table 3 for the results.

[2] We use a slightly different task setup as the one in [58]; see Appendix A.8.3 for the details of our setup.

Table 3: We present the meta-test accuracy as a percentage on non-mutually-exclusive *Omniglot* [58]. In contrast to the previous examples, here we aim to achieve the best empirical performance for each method. In particular, this task compares each methods' ability to prevent memorization. We report the mean and standard deviation after 5 trials.

20-WAY *Omniglot*	NME 1-SHOT	NME 5-SHOT
MAML [18]	23.4 ± 2.2	75.1 ± 4.8
FLI-ONLINE [27]	22.4 ± 0.5	39.1 ± 0.5
MR-MAML(W) [58]	84.2 ± 2.2	94.3 ± 0.3
PAC-BUS(H) (OURS)	$\mathbf{87.9 \pm 0.5}$	$\mathbf{95.0 \pm 0.9}$

We see that MAML [18] and FLI-Online [27] do not prevent memorization on *NME Omniglot* [58]. This is especially apparent in the 1-shot learning case, where their performance suffers significantly due to this memorization. Both MR-MAML(W) [58] and PAC-BUS(H) prevent memorization, with PAC-BUS(H) outperforming MR-MAML(W). Note that PAC-BUS(H) outperforms MR-MAML(W) by a wider margin in the 1-shot case as compared with the 5-shot case. We believe this is due to the effectiveness of the uniform stability regularizer at the base level. MR-MAML(W) suffers more in the 1-shot case because over-adaptation is more likely with fewer within-task examples.

6 Conclusion and discussion

We presented a novel generalization bound for gradient-based meta-learning, PAC-BUS. We use different generalization frameworks for tackling the distinct challenges of generalization at the two levels of meta-learning. In particular, we employ uniform stability bounds and PAC-Bayes bounds at the base- and meta-learning levels respectively. On a toy non-convex problem and the *Mini-Wiki* meta-learning task [27], we provide significantly tighter generalization guarantees as compared to state-of-the-art meta-learning bounds while maintaining comparable empirical performance. To our knowledge, this work presents the first numerically-evaluated generalization guarantees associated with a proposed meta-learning bound. On memorizable *Omniglot* [28, 58], we show that a heuristic based on the PAC-BUS bound prevents memorization of class labels in contrast to MAML [18], and better performance than meta-regularized MAML [58]. We believe our framework is well suited to the few-shot learning problems for which we present empirical results, but our framework is potentially applicable to a broad range of different settings (e.g., reinforcement learning).

We note a few challenges with our method as motivation for future work. Our bound is vacuous on larger scale learning problems such as *Omniglot*. This is partially caused by a larger KL-divergence term in the PAC-Bayes bound when using deep convolutional networks (due to the increased dimensionality of the weight vector). In addition, we do not have a theoretical analysis on the convergence properties of the algorithms presented, so we must experimentally determine the number of samples required for tight bounds. In the results of Section 5.1 and 5.2, despite an improved bound over other methods, our method does not necessarily improve empirical test performance. We emphasize that our focus in this work was on deriving stronger generalization guarantees rather than improving empirical performance. However, obtaining approaches that provide both stronger guarantees and empirical performance is an important direction for future work.

Future work can also explore ways in which to incorporate tighter or broader PAC-Bayes bounds into the framework provided here. One interesting avenue is to extend PAC-BUS by using an unbounded PAC-Bayes bound for the meta-generalization step (e.g. as presented in [21]). Another promising direction is to incorporate regularization on the weights of the network directly (e.g., L_2 regularization or gradient clipping) to create networks with smaller Lipschitz and smoothness constants.

Broader impact. The approach we present in this work aims to strengthen performance guarantees for gradient-based meta-learning. We believe that strong generalization guarantees in meta-learning, especially in the few-shot learning case, could lead to broader application of machine learning in real-world applications. One such example is for medical diagnosis, where abundant training data for certain diseases may be difficult to obtain. Another example on which poor performance is not an option is any safety critical robotic system, such as ones which involve human interaction.

Meta-learning methods typically require a lot of data and training time, and ours is not an exception. In our case, it took multiple weeks of computation time on Amazon Web Services (AWS) instances to train and compute all networks and results we present in this paper. This creates challenges with accessibility and energy usage.

References

[1] Karim Abou-Moustafa and Csaba Szepesvári. An exponential efron-stein inequality for l_q stable learning rules. *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, 2019.

[2] Pierre Alquier, The Tien Mai, and Massimiliano Pontil. Regret Bounds for Lifelong Learning. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[3] Ron Amit and Ron Meir. Meta-Learning by Adjusting Priors Based on Extended PAC-Bayes Theory. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[4] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[5] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger Generalization Bounds for Deep Nets via a Compression Approach. *arXiv preprint arXiv:1802.05296*, 2018.

[6] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[7] Peter L. Bartlett, Dylan J. Foster, and Matus J Telgarsky. Spectrally-Normalized Margin Bounds for Neural Networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[8] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the Optimization of a Synaptic Learning Rule. *Proceedings of the Conference on Optimality in Artificial and Biological Neural Networks*, pages 6–8, 1992.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's Razor. *Information Processing Letters*, 24(6):377–380, 1987.

[10] Léon Bottou and Vladimir Vapnik. Local Learning Algorithms. *Neural Computation*, 4: 888–900, 1992.

[11] Olivier Bousquet and André Elisseeff. Stability and Generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[12] Rich Caruana. Multitask Learning. *Machine Learning*, 28:41–75, 1997.

[13] Olivier Catoni. *Statistical Learning Theory and Stochastic Optimization*. École d'Été de Probabilités de Saint-Flour 2001. Springer, 2004.

[14] Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *Lecture notes - Monograph Series*. Institute of Mathematical Statistics, 2007.

[15] Alain Celisse and Benjamin Guedj. Stability revisited: new generalisation bounds for the Leave-one-Out. *arXiv preprint arXiv:1608.06412*, 2016.

[16] Giulia Denevi, Dimitris Stamos, Carlo Ciliberto, and Massimiliano Pontil. Online-within-online meta-learning. *Proceedings of the 32st Conference on Advances in Neural Information Processing Systems*, 2019.

[17] Gintare Karolina Dziugaite and Daniel M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017.

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[19] Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian Learning of Linear Classifiers. In *Proceedings of the 26th International Conference on Machine Learning*, pages 353–360. ACM, 2009.

[20] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[21] Maxime Haddouche, Benjamin Guedj, Omar Rivasplata, and John Shawe-Taylor. PAC-Bayes Unleashed: Generalisation Bounds With Unbounded Losses. *arXiv preprint arXiv:2006.07279*, 2020.

[22] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train Faster, Generalize Better: Stability of Stochastic Gradient Descent. *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[23] Elad Hazan. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

[24] Tom Heskes. Solving a Huge Number of Similar Tasks: a Combination of Multi-Task Learning and a Hierarchical Bayesian Approach. *Proceedings of the 15th International Conference on Machine Learning*, 1998.

[25] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-Learning in Neural Networks: A Survey. *arXiv preprint arXiv:2004.05439*, 2020.

[26] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic Generalization Measures and Where to Find Them. *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[27] Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Provable Guarantees for Gradient-Based Meta-Learning. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[28] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One Shot Learning of Simple Visual Concepts. *Cognitive Science*, 33, 2011.

[29] John Langford. Tutorial on Practical Prediction Theory for Classification. *Journal of Machine Learning Research*, 6(Mar):273–306, 2005.

[30] John Langford and Rich Caruana. (Not) Bounding the True Error. *Proceedings of the 14th Conference on Advances in Neural Information Processing Systems*, 2002.

[31] John Langford and John Shawe-Taylor. PAC-Bayes & margins. *Proceedings of the 16th Conference on Advances in Neural Information Processing Systems*, 2003.

[32] Andreas Maurer. Algorithmic Stability and Meta-Learning. *Journal of Machine Learning Research*, 6:967–994, 2005.

[33] David McAllester. A PAC-Bayesian Tutorial with A Dropout Bound. *arXiv preprint arXiv:1307.2118*, 2013.

[34] David A McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

[35] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. *preprint arXiv:1707.09564*, 2017.

[36] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring Generalization in Deep Learning. In *Advances in Neural Information Processing Systems*, pages 5949–5958, 2017.

[37] Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[38] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[39] Anastasia Pentina and Christoph H. Lampert. A PAC-Bayesian Bound for Lifelong Learning. *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[40] Anastasia Pentina and Christoph H. Lampert. Lifelong Learning with Non-i.i.d. Tasks. *Proceedings of the 28th Conference on Advances in Neural Information Processing Systems*, 2015.

[41] María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter Risk Certificates for Neural Networks. *arXiv preprint arXiv:2007.12911*, 2020.

[42] Sachin Ravi and Hugo Larochelle. Optimization as a Model for Few-Shot Learning. *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[43] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.

[44] Omar Rivasplata, Emilio Parrado-Hernandez, John Shawe-Taylor, Shiliang Sun, and Csaba Szepesvari. Pac-bayes bounds for stable algorithms with instance-dependent priors. *Proceedings of the 31st Conference on Advances in Neural Information Processing Systems*, 2018.

[45] Omar Rivasplata, Vikram M. Tankasali, and Csaba Szepesvari. PAC-Bayes with Backprop. *arXiv preprint arXiv:1908.07380*, 2019.

[46] Jonas Rothfuss, Vincent Fortuin, and Andreas Krause. PACOH: Bayes-Optimal Meta-Learning with PAC-Guarantees. *arXiv preprint arXiv:2002.05551*, 2020.

[47] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-Learning with Memory-Augmented Neural Networks. *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[48] Jurgen Schmidhuber. Evolutionary Principles in Self-Referential Learning. On Learning how to Learn: The Meta-Meta-Meta...-Hook. Diploma thesis, Technische Universitat Munchen, Germany, 1987.

[49] Matthias Seeger. PAC-Bayesian Generalisation Error Bounds for Gaussian Process Classification. *Journal of Machine Learning Research*, 3(Oct):233–269, 2002.

[50] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[51] John Shawe-Taylor and Robert C. Williamson. A PAC Analysis of a Bayesian Estimator. *Proceedings of the 10th Conference on Computational Learning Theory*, 1997.

[52] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying Some Distributional Robustness with Principled Adversarial Training. *arXiv preprint arXiv:1710.10571*, 2020.

[53] Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A Strongly Quasiconvex PAC-Bayesian Bound. *Machine Learning Research*, 76:1–26, 2017.

[54] Sebastian Thrun and Lorien Pratt. *Learning to Learn*. Springer Science & Business Media, 1998.

[55] Vladimir N. Vapnik and A. Ya Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Dokl. Akad. Nauk*, 181(4), 1968.

[56] Ricardo Vilalta and Youssef Drissi. A Perspective View and Survey of Meta-Learning. *Artificial Intelligence Review*, 18:77–95, 2002.

[57] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. *Proceedings of the 30th Conference on Advances in Neural Information Processing Systems*, 2016.

[58] Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-Learning without Memorization. *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[59] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan Adams, and Peter Orbanz. Nonvacuous Generalization Bounds at the ImageNet Scale: A PAC-Bayesian Compression Approach. *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We provide a meta-learning bound (Theorem 3) and a regularization scheme to minimize the bound (Algorithm 1), show that it provides tighter guarantees as compared with other meta-learning bounds (see Tables 1 and 2), and demonstrate an effective heuristic (Algorithm A.2) based on the bound on a larger scale task (see Table 3).
   (b) Did you describe the limitations of your work? [Yes] A discussion of the limitations of this work is in the second paragraph of Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the last paragraph of Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Problem formulation assumptions are described in Section 2 and assumptions of the theoretical results are presented in the statement of Theorem 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] A proof of Theorem 3 (the main theoretical result of this work) is presented as part of Section 4.2. Derivations of the specialized bounds are presented in Appendix A.1.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code, data (or instructions for downloading data), and instructions for reproducing all results in the paper are included in the supplemental code.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Information about how tasks are generated, the specific data splits, range of hyperparameters searched, and other related details are described in Appendix A.8.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All results (Tables 1, 2, 3, and A.1) report the standard deviation and mean after 5 trials.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details of computing resources and computation times for the results presented in the paper are provided in Appendix A.8.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] The datasets used in this paper (*Mini-wiki* [27] and *Omniglot* [28]) are cited throughout their use in the paper. The code on which some implementations are based is cited in the supplemental code. The python libraries used are cited in Appendix A.8 and the included code.
   (b) Did you mention the license of the assets? [Yes] Licenses of all datasets and software assets are mentioned in Appendix A.8
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The supplemental code is an asset for running the algorithms presented in this paper. This will be publicly available on github (with an MIT license) after publication.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We briefly describe the collection process for the datasets used in this paper in Appendix A.8.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We discuss the potential for offensive content in the *Mini-Wiki* [27] in Appendix A.8.2, and the potential for personally identifiable information in *Omniglot* [28] in Appendix A.8.3.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]