ADAPTIVE META-LEARNING VIA DATA-DEPENDENT PAC-BAYES BOUNDS

Lior Friedman
The Viterbi Faculty of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Haifa 3200003, Israel
liorf@campus.technion.ac.il

Ron Meir
The Viterbi Faculty of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Haifa 3200003, Israel
rmeir@ee.technion.ac.il

ABSTRACT

Meta-learning aims to extract common knowledge from similar training tasks in order to facilitate efficient and effective learning on future tasks. Several recent works have extended PAC-Bayes generalization error bounds to the meta-learning setting. By doing so, prior knowledge can be incorporated in the form of a distribution over hypotheses that is expected to lead to low error on new tasks that are similar to those that have been previously observed. In this work, we develop novel bounds for the generalization error on test tasks based on recent data-dependent bounds and provide a novel algorithm for adapting prior knowledge to downstream tasks in a potentially more effective manner. We demonstrate the effectiveness of our algorithm numerically for few-shot image classification tasks with deep neural networks and show a significant reduction in generalization error without any additional adaptation data.

1 INTRODUCTION

Over the last few decades, the field of machine learning has developed rapidly both theoretically and as an engineering practice. Of particular interest is the field of learning and adapting quickly from only a few examples, which requires a balance between prior experience and new information in order to solve new tasks effectively without overfitting. One common approach to tackle this few-shot learning problem is that of meta-learning, where training data is used to create a prior that is conducive to the downstream task. This approach has shown promising empirical results in a variety of domains (see survey (Hospedales et al., 2021)), especially for cases with few test examples.

As an illustrative example of the meta-learning problem, we might consider a visual classification system that must also be capable of identifying new categories of objects with very few examples. In order to achieve this goal, the system must maintain prior knowledge on similar vision problems, and utilize this prior to adapt quickly to the new data.

In order to better understand the generalization capabilities of classification methods and give upper bounds on the gap between the training and test performance, several theoretical frameworks have been devised. Among these frameworks, methods based on PAC-Bayes bounds (McAllester, 1999) are of particular interest, as they result in practical optimization algorithms with potentially non-vacuous generalization guarantees with high probability. As such, it is not surprising that several works have extended the PAC-Bayes framework to the domain of meta-learning, such as Pentina & Lampert (2014), Amit & Meir (2018), Rothfuss et al. (2021), Liu et al. (2021) and Farid & Majumdar (2021).

Several recent works have established non-vacuous generalization bounds for practical deep learning problems, such as the methods suggested by Dziugaite & Roy (2017) and later improved by Pérez-Ortiz et al. (2021). In both cases, the use of a *data-dependent* prior was shown to be a major component in achieving these impressive results. As such, data-dependent PAC-Bayes bounds such as those proposed in Rivasplata et al. (2020) may be of great interest for meta-learning.

In this work, we utilize data-dependent PAC-Bayes techniques to provide an upper bound on the generalization error for new tasks by adapting an existing distribution over priors to better fit the given test task. This approach allows us to use existing methods to meta-learn a distribution over training tasks and provides a potentially tighter guarantee for the new task. We compare our bounds to known bounds for a simple setting, and develop a practical algorithm based on our bounds. We demonstrate the effectiveness of this meta-adaptation approach for classification on vision tasks.

2 BACKGROUND

2.1 PAC-BAYES BOUNDS

The common setting for learning consists of a set of independent examples $S = \{z_i\}_{i=1}^m \subset \mathcal{Z}^m$, drawn from an unknown distribution $z_i \sim \mathcal{D}$. We denote $S \sim \mathcal{D}^m$ the distribution over the samples. For the common setting of classification, each example $z_i = (x_i, y_i)$ is a pair of data and label. Given a space of hypotheses $\mathcal{H}$ and a sample S, we would like to find a hypothesis $h \in \mathcal{H}$ that minimizes the *expected loss* $\mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$, where $\ell : \mathcal{Z} \to [a,b]$ is a bounded loss function (some PAC-Bayes bounds apply for unbounded losses with concentration properties such as sub-gamma and sub-Gaussian distributions, and may be useful for future extensions). Since $\mathcal{D}$ is unknown, we must use the training data S to find a hypothesis that minimizes the expected loss with high probability, based on the sample of size m.

The PAC-Bayes framework, as formulated by McAllester (1999), takes as input the training data S as well as an inductive bias in the form of a prior distribution P over $\mathcal{H}$. These are then used to construct a *posterior distribution* Q over $\mathcal{H}$, and $h \sim Q$ is then sampled. We define the expected and empirical errors

$$\mathcal{L}(h,\mathcal{D}) \triangleq \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)] \;;\; \hat{\mathcal{L}}(h,S) \triangleq \frac{1}{m}\sum_{i=1}^{m}\ell(h,z_i).$$

In the PAC-Bayes setting we sample $h \sim Q$ and estimate the expected performance over the posterior $Q \in \mathcal{M}(\mathcal{H})$ (the set of distributions over hypotheses). This process results in a randomized algorithm that enables the estimation of the expected and empirical errors for the posterior distributions,

$$\mathcal{L}(Q,\mathcal{D}) \triangleq \mathbb{E}_{h\sim Q}[\mathcal{L}(h,\mathcal{D})] \;;\; \hat{\mathcal{L}}(Q,S) \triangleq \mathbb{E}_{h\sim Q}\left[\hat{\mathcal{L}}(h,S)\right] .$$

Following these definitions, one can derive a PAC-Bayes theorem for the single task setting, as formulated by McAllester (1999).

Theorem 2.1. (McAllester's single task bound) Let $P \in \mathcal{M}(\mathcal{H})$ be some prior distribution over $\mathcal{H}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the choice of S, uniformly for all posteriors $Q \in \mathcal{M}(\mathcal{H})$,

$$\mathcal{L}(Q,\mathcal{D}) \leq \hat{\mathcal{L}}(Q,S) + \sqrt{\frac{D_{KL}(Q||P) + \log\frac{m}{\delta}}{2(m-1)}} ,$$

where $D_{KL}(Q||P) \triangleq \mathbb{E}_{h\sim Q}\left[\log\frac{Q(h)}{P(h)}\right]$ is the Kullback-Leibler divergence.

This theorem is commonly interpreted as the expected error being upper bounded by the empirical error plus a complexity term that depends on the probability parameter δ, the sample size m, and the divergence of the posterior from the prior. Since this theorem holds uniformly over all Q, we can derive a practical learning algorithm that chooses Q such that it minimizes the right-hand-side of this bound. Naturally, this bound is affected by the choice of P, the prior over $\mathcal{H}$, as ideally we would like to have a prior that is close to posteriors that achieve low empirical error, thereby motivating the notion of data-dependent priors.

2.2 PAC-BAYES BOUNDS FOR META-LEARNING

The basic meta-learning setting assumes an input comprised of several training tasks from a single task environment. A meta-learning algorithm must extract the necessary common knowledge (in the form of a prior) to efficiently learn new tasks in the same environment. Following the formulation of PAC-Bayes bounds for lifelong learning (Pentina & Lampert, 2014) and meta-learning (Amit & Meir, 2018), we assume a shared sample space $\mathcal{Z}$, hypothesis space $\mathcal{H}$ and loss function $\ell : \mathcal{Z}\times\mathcal{H} \to [a,b]$, and a set of training datasets $\{S_1,...,S_N\}$ of size m each (we assume equal sizes for simplicity of analysis). Each training dataset S_i is assumed to come from an unknown distribution $S_i \sim \mathcal{D}_i^m$, and these distributions are sampled i.i.d. from a shared (and also unknown) task distribution $D_i \sim \tau$.

The goal of a meta-learning algorithm is to construct a prior P such that given samples S_T from a new task $\mathcal{D}_T \sim \tau$, the base learner uses both to construct a posterior $Q(P,S_T)$ over the hypothesis space $\mathcal{H}$. In order to evaluate our constructed prior, we can consider its expected error

$$\mathcal{L}(P,\tau) \triangleq \mathbb{E}_{\mathcal{D}\sim\tau}\left[\mathbb{E}_{S\sim\mathcal{D}^m}\left[\mathbb{E}_{h\sim Q(P,S)}[\mathcal{L}(h,\mathcal{D})]\right]\right] . \tag{1}$$

The meta-learning PAC-Bayes framework can thus be seen as learning a hyper-posterior distribution $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ over priors. Similarly to the single task setting, we assume access to a hyper-prior distribution $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$, as

well as training datasets $\{S_1, ..., S_N\}$. We would like to optimize over $\mathcal{Q}$ in order to minimize the expected *transfer error*

$$\mathcal{L}(\mathcal{Q}, \tau) \triangleq \mathbb{E}_{P\sim\mathcal{Q}}\left[\mathcal{L}(P, \tau)\right].$$

Since the true task distribution τ is unknown, we can use an estimate in the form of the empirical multi-task error

$$\hat{\mathcal{L}}(\mathcal{Q}, S_1, ..., S_N) \triangleq \mathbb{E}_{P\sim\mathcal{H}}\left[\frac{1}{N}\sum_{i=1}^{N}\hat{\mathcal{L}}(Q(P, S_i), S_i)\right].$$

A similar approach to the single task case leads us to PAC-Bayes bounds on the transfer error, such as the following.

Theorem 2.2. (Meta-learning bound (Amit & Meir, 2018)) Let $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ be some hyper-prior distribution, and let $Q : \mathcal{Z}^m \times \mathcal{M}(\mathcal{H}) \to \mathcal{M}(\mathcal{H})$ be a given base learner. Let $\ell : \mathcal{H} \times \mathcal{Z} \to [0, 1]$ be a bounded loss function. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the choice of $\mathcal{D}_1, ..., \mathcal{D}_N \sim \tau, S_i \sim \mathcal{D}_i$, uniformly over all hyper-posteriors $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$,

$$\mathcal{L}(\mathcal{Q}, \tau) \leq \hat{\mathcal{L}}(\mathcal{Q}, S_1, ..., S_N) + \sqrt{\frac{D_{KL}(\mathcal{Q}||\mathcal{P}) + \log\frac{2N}{\delta}}{2(N-1)}} + \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{D_{KL}(\mathcal{Q}||\mathcal{P}) + \mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q_i||P)\right] + \phi}{2(m-1)}}$$

where $Q_i \triangleq Q(P, S_i)$ and $\phi \triangleq \log\frac{2Nm}{\delta}$.

This bound on the transfer error contains two complexity terms: an environment-level term that decreases as $N \to \infty$, and a second task-level term that decreases as $m \to \infty$.

3 Adaptive meta-learning

In this section we go beyond standard meta-learning bounds that use a hyper-posterior formed from previous tasks as a hyper-prior for a new test task. To do so we allow the new hyper-prior to depend both on the learned hyper-posterior and on the new test data, implying that it is now data-dependent (with respect to the test data). We refer to this setup as adaptive, see Figure 1b. In order to deal with the added technical complexity, we make use of recent data-dependent PAC-Bayes bounds from Rivasplata et al. (2020), and provide novel meta PAC-Bayes bounds and algorithms in this augmented setting. The bounds display interesting tradeoffs between the empirical and complexity terms that go beyond standard results.

Unlike existing meta-learning bounds, our bounds depend on the empirical error on the available adaptation data from the new task, and do not depend on the number of training tasks, but rather on the quality of the learned hyper-prior. In particular, equation (7) implies that if the hyper-prior achieves poor empirical performance for the test task, it is wise to ignore it and effectively learn from scratch using the adaptation data. On the other hand, if the hyper-prior provides us with low empirical loss, using complexity terms that encourage remaining near said hyper-prior is beneficial.

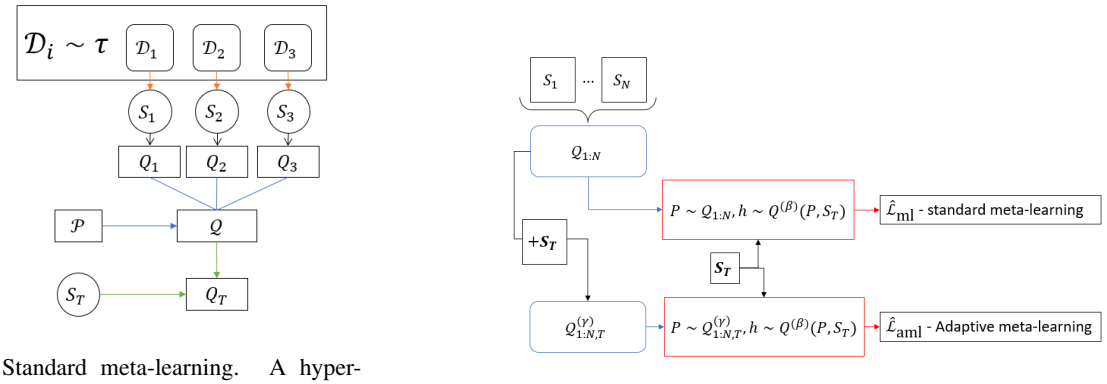
3.1 Meta-adaptation bounds for meta-learning

While these bounds provide us with useful and practical approaches to use the available training data for meta-learning, we have seen that they provide guarantees in expectation for a newly sampled task. In pursuit of tighter bounds for specific downstream tasks, we introduce the idea of meta-adaptation, illustrated in Figure 1b. Using the meta-learned hyper-posterior as a hyper-prior for the downstream task, we derive a high-probability bound on the expected loss for a *specific* downstream task.

As we have seen, the meta-learning framework provides us with an informative hyper-posterior from which a good prior (i.e. one with low expected error) can be sampled. Given a specific downstream task $\mathcal{D}_T \sim \tau$ and a sample $S_T \sim \mathcal{D}_T^m$, we would like to provide a bound on the performance of any hyper-posterior that uses S_T, and the given hyper-prior $\mathcal{Q}_{1:N}$ that we previously meta-learned. In order to do so, we make use of PAC-Bayes bounds for data-dependent priors as the i.i.d. assumption common to PAC-Bayes bounds may not apply. As such, we re-state a known inequality from Rivasplata et al. (2020) and adapt it to the meta-learning setting

Theorem 3.1. (PAC-Bayes for stochastic kernels - adapted from Theorem 2 in Rivasplata et al. (2020)) Let $P \in \mathcal{K}(\mathcal{Z}^m, \mathcal{H})$ be a stochastic kernel (namely, this means P may depend on the sampled data S), let $A : \mathcal{Z}^m \times \mathcal{H} \to \mathbb{R}^k$ be a measurable function for some positive integer k and $F : \mathbb{R}^k \to \mathbb{R}$ be a convex function. Define $f \triangleq F \circ A$, and

$$\xi(P_S, \mathcal{D}, f) = \int_{\mathcal{Z}^m}\int_{\mathcal{H}} e^{f(S,h)} P_S(dh)\mathcal{D}(dS),$$



(a) Standard meta-learning. A hyper-prior $\mathcal{P}$ is adapted using training data $S_1, \ldots, S_N$ to construct a hyper-posterior $\mathcal{Q}$. This hyper-posterior is used to facilitate fast learning on the test task.

(b) Flowchart of meta-adaptation. A hyper-prior $\mathcal{Q}_{1:N}$ is learned from the training data. Standard meta-learning bounds use the test data to adapt the sampled prior before sampling a specific hypothesis. Our approach using data-dependent bounds also applies S_T to the hyper-prior, resulting in a data-dependent hyper-posterior $\mathcal{Q}_{1:N,T}$.

Figure 1: Standard meta-learning and meta-adaptation.

such that the data-dependent P_S is the distribution over $\mathcal{H}$ corresponding to the sampled S. Assuming $\xi(P_S, \mathcal{D}, f)$ is finite (see below), then for any posterior $Q \in \mathcal{K}(\mathcal{Z}^m, \mathcal{H})$, with probability at least $1 - \delta$ over the choice of $S \sim \mathcal{D}$

$$\mathbb{E}_{h \sim Q_S}\left[f(S,h)\right] \leq D_{KL}(Q_S||P_S) + \log\left(\xi(P_S, \mathcal{D}, f)/\delta\right). \tag{2}$$

An important but subtle distinguishing factor between this bound and classical PAC-Bayes bounds is that this bound applies to each individual posterior Q_S, but *does not apply uniformly* to all posteriors. This seemingly minor difference means that the bound of (2) is not applicable for optimization over posteriors. Taking this limitation into account, this Theorem still provides an applicable upper bound on the expected loss for a given trained posterior.

The term $\xi(P_S, \mathcal{D}, f)$ is known as the *moment-generating function*, and, when it exists, it is an alternative specification of the probability distribution for f. This moment-generating function intuitively quantifies the concentration of the function f in the stochastic kernel $P \in \mathcal{K}(\mathcal{Z}^m, \mathcal{H})$, and will be low if f is well-concentrated. The term $\log \xi(P_S, \mathcal{D}, f)$ is commonly referred to as the *log-moment* and we will also do so throughout the rest of this paper for convenience. One simple setting where this term can be upper bounded is when the prior P_S is data-free and $f(\cdot) \in [a, b]$. By switching the order of expectations (Fubini's theorem) and using Hoeffding's lemma, an upper bound for $f(S,h) = \lambda(\mathcal{L}(h,\mathcal{D}) - \hat{\mathcal{L}}(h,S))$ for any $\lambda \in \mathbb{R}$ would be

$$\mathbb{E}_{h \sim Q_S}\left[\lambda(\mathcal{L}(h,\mathcal{D}) - \hat{\mathcal{L}}(h,S))\right] \leq D_{KL}(Q_S||P_S) + \log\left(\frac{1}{\delta}\right) + \frac{\lambda^2(b-a)^2}{8}.$$

Notably, this bound is similar to that of Catoni (2004). Other choices of f result in other traditional PAC-Bayes bounds such as that of McAllester (1999) via similar methods (see Rivasplata et al. (2020) for further detail). One particularly appealing application of this general bound is for data-dependent priors with bounded log-moment terms, for example by conforming to certain algorithmic stability properties As an example for this notion, Rivasplata et al. (2020) show that for any prior satisfying the differential privacy property $DP(\epsilon)$, the log-moment can be upper bounded by the log-moment of a data-free prior plus a term that depends on the privacy parameter ϵ.

These data-dependent bounds can be applied to the meta-learning setting, giving us the following general result.

Theorem 3.2. Let $\mathcal{Q}_{1:N} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ be a meta-learned hyper-posterior (e.g. the result of meta-training on $\{S_1, ..., S_N\}$), and let $\mathcal{D}_T \sim \tau$ be a *given* test task. Let $Q : \mathcal{Z}^m \times \mathcal{M}(\mathcal{H}) \to \mathcal{M}(\mathcal{H})$ be a given base learner. Let $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$ be a bounded loss function. For any $\delta_T \in (0,1)$, for any hyper-posterior $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$, and for all $\lambda > 0$, with probability at least $1 - \delta_T$ over the draw of $S_T \sim \mathcal{D}_T$:

$$\mathcal{L}(\mathcal{Q}, \mathcal{D}_T) \leq \hat{\mathcal{L}}(\mathcal{Q}, S_T) + \frac{1}{\lambda} D_{KL}(\mathcal{Q}||\mathcal{Q}_{1:N}) + \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right), \tag{3}$$

$$\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T) \triangleq \mathbb{E}_{S \sim \mathcal{D}_T, P \sim \mathcal{Q}_{1:N}, h \sim Q(P,S)}\left[e^{\lambda\left(\mathcal{L}(h,\mathcal{D}_T) - \hat{\mathcal{L}}(h,S)\right)}\right].$$

The full proof of Theorem 3.2 is in Appendix A.2, and follows from Theorem 3.1 using a two-level prior hypothesis $(\mathcal{Q}_{1:N}, Q)$. Concretely, we first sample $P \sim \mathcal{Q}_{1:N}$ and then sample $h \sim Q(P, S_T)$ to arrive at a hypothesis ,namely Q is data-dependent. This two-level prior is compared to a two-level posterior $(\mathcal{Q}, Q)$, for any $\mathcal{Q}$. The proof itself applies even if the loss function l is not bounded, but this assumption is useful for providing bounds on the log-moment term.

This bound differs significantly from more traditional bounds that utilize a data-free two-level prior $(\mathcal{Q}_{1:N}, P)$. Such bounds apply uniformly over hyper-posteriors and are therefore suitable for optimization, but refer to the average loss on training tasks and contain additional complexity terms compared to (3) that result from using a data-free prior. An example of such a bound using Hoeffding's inequality is

$$\mathcal{L}(\mathcal{Q}, \mathcal{D}_T) \leq \hat{\mathcal{L}}(\mathcal{Q}, S_T) + \frac{1}{\lambda} D_{KL}(\mathcal{Q}||\mathcal{Q}_{1:N}) + \frac{1}{\lambda}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q(P, S_T)||P)\right] + \frac{1}{\lambda}\log\frac{1}{\delta_T} + \frac{\lambda}{8m}. \tag{4}$$

One immediate result of (3) is a bound on the expected error of the hyper-prior for the downstream task, achieved by picking $\mathcal{Q} = \mathcal{Q}_{1:N}$, and appears in Corollary A.3.

It is important to note that the log-moment term $\log\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)$ is a key term in both cases. It will be low if the empirical losses are well concentrated around the expected loss for our data-dependent prior. One possible method of achieving this is by choosing a base learner Q with algorithmic stability properties such as the empirical Gibbs posterior, as we will see in the next subsection.

3.2 Meta-adaptation with Gibbs posteriors

One class of hyper-posteriors that is especially appealing for analysis given Theorem 3.2 is the class of Gibbs posteriors.

Definition 3.1. The Gibbs distribution with parameter $\beta > 0$ is defined as

$$Q^{\beta}(P, S)(h) = \frac{P(h)e^{-\beta\hat{\mathcal{L}}(h,S)}}{\mathbb{E}_{h\sim P}\left[e^{-\beta\hat{\mathcal{L}}(h,S)}\right]}. \tag{5}$$

It is well-known (Catoni, 2004) that given a specific value for λ, using Donsker and Varadhan's variational formula (Donsker & Varadhan, 1975) provides the hyper-posterior that minimizes the right-hand side of (3), given by

$$\mathcal{Q}^{\lambda}_{1:N,T} = \frac{\mathcal{Q}_{1:N}e^{-\lambda\hat{\mathcal{L}}(Q,S_T)}}{\mathbb{E}_{P\sim\mathcal{Q}_{1:N}}\left[e^{-\lambda\hat{\mathcal{L}}(Q,S_T)}\right]}.$$

The meaning of this result is that given that we know the relative importance of the empirical loss and the KL-divergence, the optimal hyper-posterior that minimizes (3) is the Gibbs hyper-posterior $\mathcal{Q}^{\lambda}_{1:N,T}$.

This property encourages further exploration of the Gibbs hyper-posterior for meta-adaptation. Using (5), we define

$$\mathcal{Q}^{\gamma}_{1:N,T} \triangleq \frac{\mathcal{Q}_{1:N}e^{-\gamma\hat{\mathcal{L}}(Q^{\beta}(P,S_T),S_T)}}{Z_{\gamma}(S_T, \mathcal{Q}_{1:N})} \quad ; \quad Z_{\gamma}(S_T, \mathcal{Q}_{1:N}) \triangleq \mathbb{E}_{P\sim\mathcal{Q}_{1:N}}\left[e^{-\gamma\hat{\mathcal{L}}(Q^{\beta}(P,S_T),S_T)}\right] \tag{6}$$

as the Gibbs posterior for the meta-learned problem. $\mathcal{Q}^{\gamma}_{1:N,T}$ is a specific case of $\mathcal{Q}^{\lambda}_{1:N,T}$ where the base learner is a Gibbs posterior. We have

$$D_{KL}(\mathcal{Q}^{\gamma}_{1:N,T}||\mathcal{Q}_{1:N}) = -\gamma\hat{\mathcal{L}}(\mathcal{Q}_{1:N,T}, S_T) - \log Z_{\gamma}(S_T, \mathcal{Q}_{1:N}).$$

So plugging these values in (3) gives us

$$\mathcal{L}(\mathcal{Q}^{\gamma}_{1:N,T}, \mathcal{D}_T) \leq \hat{\mathcal{L}}(\mathcal{Q}^{\gamma}_{1:N,T}, S_T) - \frac{\gamma}{\lambda}\hat{\mathcal{L}}(\mathcal{Q}^{\gamma}_{1:N,T}, S_T) - \frac{1}{\lambda}\log Z_{\gamma}(S_T, \mathcal{Q}_{1:N}) + \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right).$$

In order to clarify this expression, we will use more intuitive definitions for our empirical losses.

Definition 3.2. The adapted meta-loss (AML) is

$$\hat{\mathcal{L}}_{\text{aml}}(\mathcal{Q}^{\gamma}_{1:N,T}, S_T) = \mathbb{E}_{P\sim\mathcal{Q}^{\gamma}_{1:N,T}}\mathbb{E}_{h\sim Q^{\beta}(P,S_T)}\left[\hat{\mathcal{L}}(h, S_T)\right]$$

meaning that S_T is used to adapt both the hyper-prior ($\mathcal{Q}_{1:N}$) and the sampled prior that depends on S_T. $\hat{\mathcal{L}}_{\text{aml}}$ is a function of both γ and β. The standard meta-loss

$$\hat{\mathcal{L}}_{\text{ml}}(\mathcal{Q}_{1:N}, S_T) = \mathbb{E}_{P\sim\mathcal{Q}_{1:N}}\mathbb{E}_{h\sim Q^\beta(P,S_T)}\left[\hat{\mathcal{L}}(h, S_T)\right] ,$$

is the empirical loss of the base learner using the meta-learning hyper-prior $\mathcal{Q}_{1:N}$. $\hat{\mathcal{L}}_{\text{ml}}$ depends on β, but not on γ.

Using these definitions, we have:

$$\mathcal{L}(\mathcal{Q}^\gamma_{1:N,T}, \mathcal{D}_T) \leq (1-\frac{\gamma}{\lambda})\hat{\mathcal{L}}_{\text{aml}}(\mathcal{Q}^\gamma_{1:N,T}, S_T) - \frac{1}{\lambda}\log \mathbb{E}_{P\sim\mathcal{Q}_{1:N}}\left[e^{-\gamma\hat{\mathcal{L}}(Q^\beta(P,S_T),S_T)}\right] + \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right).$$

By simplifying the second term using Jensen's inequality, we arrive at the following theorem.

Corollary 3.3. Let $\mathcal{Q}_{1:N}$ be a hyper-posterior (e.g. the result of meta-training on $\{S_1, ..., S_N\}$), and let $\mathcal{D}_T \sim \tau$ be a given test task. Define $\mathcal{Q}^\gamma_{1:N,T}$ as in (6). For all $\lambda > 0, \gamma > 0$, with probability at least $1-\delta_T$ over the draw of $S_T \sim \mathcal{D}_T$:

$$\mathcal{L}(\mathcal{Q}^\gamma_{1:N,T}, \mathcal{D}_T) \leq (1-\frac{\gamma}{\lambda})\hat{\mathcal{L}}_{\text{aml}}(\mathcal{Q}^\gamma_{1:N,T}, S_T) + \frac{\gamma}{\lambda}\hat{\mathcal{L}}_{\text{ml}}(\mathcal{Q}_{1:N}, S_T) + \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right). \tag{7}$$

In order to better understand the impact of the log-moment term, we remark that Rivasplata et al. (2020) have shown that for $\lambda = \sqrt{m}$ and given that the base learner is the empirical Gibbs posterior Q^β and the loss is bounded,

$$\log\tilde{\xi}(\sqrt{m}, \mathcal{Q}_{1:N}, \mathcal{D}_T) \leq 2 + \log(1+\sqrt{e}) + \frac{2\beta}{\sqrt{m}}.$$

As a consequence of this, choosing $\beta = \sqrt{m}$ would result in the log-moment to be bounded by a constant C. Consequentially, the bound takes the form

$$\mathcal{L}(\mathcal{Q}^\gamma_{1:N,T}, \mathcal{D}_T) \leq (1-\frac{\gamma}{\sqrt{m}})\hat{\mathcal{L}}_{\text{aml}}(\mathcal{Q}^\gamma_{1:N,T}, S_T) + \frac{\gamma}{\sqrt{m}}\hat{\mathcal{L}}_{\text{ml}}(\mathcal{Q}_{1:N}, S_T) + \frac{1}{\sqrt{m}}\log\frac{1}{\delta_T} + \frac{C}{\sqrt{m}}.$$

To simplify, this implies that if $\hat{\mathcal{L}}_{\text{ml}}$ is low (meaning the hyper-prior has low empirical error), we would like to choose a high value for γ in order to balance the first two terms. In other words, if the hyper-prior performs well empirically, there is little harm in adapting it to the new task. If the hyper-prior performs poorly, our bound becomes more complex, as we would like γ to be high enough so that $\hat{\mathcal{L}}_{\text{aml}}$ is low but doing so weakens the remaining terms. Choosing a high value for λ in (7) would allow us to effectively ignore the poorly performing hyper-prior.

3.3 Tighter bounds for the low error setting

Theorem 3.2 is the result of applying the generic data-dependent bound (Theorem 3.1) with the measurement function $f = \lambda(\mathcal{L}(\mathcal{Q}, \mathcal{D}_T) - \hat{\mathcal{L}}(\mathcal{Q}, S_T))$. For cases where $\hat{\mathcal{L}}(\mathcal{Q}, S_T)$ is low, it may be better to use a bound based on the binary KL-divergence $f = \lambda\text{kl}(\hat{\mathcal{L}}(\mathcal{Q}, S_T)||\mathcal{L}(\mathcal{Q}, \mathcal{D}_T))$,

$$\text{kl}(q||p) = q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p}, \quad q, p \in [0,1]$$

Doing so gives the upper bound

$$\text{kl}(\hat{\mathcal{L}}(\mathcal{Q}, S_T)||\mathcal{L}(\mathcal{Q}, \mathcal{D}_T)) \leq \frac{1}{\lambda}D_{KL}(\mathcal{Q}||\mathcal{Q}_{1:N}) + \frac{1}{\lambda}\log\left(\bar{\xi}_{\text{kl}}/\delta_T\right) \tag{8}$$

where $\bar{\xi}_{\text{kl}} \triangleq \bar{\xi}(\mathcal{Q}_{1:N}, \mathcal{D}_T, \lambda, \text{kl})$.

If the base learner Q is the Gibbs posterior ((5)), the moment term of (8) can be meaningfully bounded. To do so, we make use of the notion of differential privacy at scale ϵ, $DP(\epsilon)$ (see Definition A.4 in the appendix).

It has been proven (McSherry & Talwar, 2007; Rivasplata et al., 2020) that the Gibbs posterior $Q(P,S)(h) \propto P(h)e^{-\beta\hat{\mathcal{L}}(h,S)}$ is $DP\left(\frac{2\beta}{m}\right)$. This result can be extended to the meta-learning setting as follows.

Proposition 3.4. Let $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ be a hyper-prior. Let $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$ be a bounded loss function. If the base learner $Q \in \mathcal{M}(\mathcal{H})$ is the Gibbs posterior $Q(P,S)(h) \propto P(h)e^{-\beta\hat{\mathcal{L}}(h,S)}$, then the pair hypothesis $(\mathcal{P}, Q)$ that samples $P \sim \mathcal{P}$ and $h \sim Q(P,S)$ satisfies $DP\left(\frac{2\beta}{m}\right)$.

The proof of Proposition 3.4 is in Appendix A.4. This allows us to extend an existing result for PAC-Bayes with private data-dependent priors. By making use of Theorem 4.2 in Dziugaite & Roy (2018) as well as (8) with $\lambda = m$, we get the following Theorem.

Theorem 3.5. Let $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$ be a bounded loss function. Let $\mathcal{Q}_{1:N}$ be a hyper-posterior (e.g. the result of meta-training on $\{S_1, ..., S_N\}$), and let $\mathcal{D}_T \sim \tau$ be a given test task. If the base learner $Q \in \mathcal{M}(\mathcal{H})$ is the Gibbs posterior $Q(P,S)(h) \propto P(h)e^{-\beta\hat{\mathcal{L}}(h,S)}$, for any $\delta \in (0,1)$, uniformly for all hyper-posteriors $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$, with probability at least $1-\delta$ over the draw of $S_T \sim \mathcal{D}_T$,

$$\mathrm{kl}(\hat{\mathcal{L}}(\mathcal{Q}, S_T)||\mathcal{L}(\mathcal{Q}, \mathcal{D}_T)) \leq \frac{2\beta^2}{m^2} + \frac{\beta}{m}\sqrt{\frac{2\log(4/\delta_T)}{m}} + \frac{1}{m}\left(D_{KL}(\mathcal{Q}||\mathcal{Q}_{1:N}) + \log\frac{4\sqrt{m}}{\delta_T}\right).$$

3.4 Empirical evaluation

Since Theorem 3.2 does not apply uniformly over hyper-posteriors, we cannot derive a practical optimization algorithm just by minimizing the right-hand side of (3). One way to resolve this issue is to add a base-learner complexity term $\frac{1}{\lambda}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q||P)\right]$ to (3), thus arriving at (4) that applies to data-free priors, and derive optimizable meta-adaptation bounds based on it. Another method to address this issue is to use bounds such as Theorem 4.2 of Dziugaite & Roy (2018) that apply uniformly for all posteriors, but only for specific classes of data-dependent priors, in this case differentially-private ones. Another potential solution to this issue is to split the provided data S_T into disjoint adaptation and evaluation sets, S_{adapt} and S_{eval}, during the meta-adaptation phase, and learn the posterior Q based on the adaptation set S_{adapt} alone. Since this is already common practice in meta-learning algorithms (e.g. sampling different data points for the meta-update step Finn et al. (2017)), we chose to do so for our empirical evaluation.

We use stochastic neural networks (Graves, 2011; Blundell et al., 2015) similarly to the MLAP (Amit & Meir, 2018) algorithm, and consider meta-adaptation with three different hyper-priors. *(i)* The meta-learned hyper-prior $\mathcal{Q}_{1:N}$. *(ii)* A zero-centered stochastic neural network. *(iii)* An adaptive hyper-prior based on the last optimization loop, as described in algorithm 1, and in more detail in Appendix A.1.

Algorithm 1 describes the meta-adaptation algorithm for stochastic neural networks using θ to mark the meta-learner parameters and ϕ to mark the base learner parameters. The standard meta-testing algorithm that is used to adapt a sampled prior to a given task is described in Appendix A.1. We use a slight abuse of notation $\phi \sim \theta$ to denote sampling according to the parametric distribution defined by the stochastic neural network.

Algorithm 1 Meta-adaptation

function META-ADAPT($\theta_{1:N}$, $S_T \sim \mathcal{D}_T$)
  Choose algorithmic parameters $\eta_\alpha, K, \lambda, N_{MC}$
  Initialize $\hat{\theta}_{1:N,T} \leftarrow \theta_{1:N}$
  while Not converged **do** ▷ Or limit number of epochs
    Sample $\hat{\phi}_1, \ldots, \hat{\phi}_K \sim \hat{\theta}_{1:N,T}$
    for each meta-batch k from 1 to K **do**
      Set $S_k \leftarrow S_T$
      Split S_k to $S_{k,\text{adapt}}, S_{k,\text{eval}}$
      for each Monte-Carlo estimation j from 1 to N_{MC} **do**
        Sample $h_j \sim \hat{\phi}_k$,
        Calculate $L_{kj}(h_j, S_{k,\text{eval}}) = \hat{\mathcal{L}}(h_j, S_{k,\text{eval}})$
      $J_k(L_{kj}, \hat{\theta}_{1:N,T}, \lambda, \hat{\phi}_k) = \frac{1}{N_{MC}}\sum_{j=1}^{N_{MC}} L_{kj} + \frac{1}{\lambda}D_{KL}(\hat{\phi}_k||\hat{\theta}_{1:N,T})$ ▷ $S_{k,\text{adapt}}$ can be used to optimize λ w.r.t. J_k
    $J(\hat{\theta}_{1:N,T}, \hat{\phi}_1, \ldots, \hat{\phi}_K, \lambda, J_1, \ldots, J_K) = \frac{1}{K}\sum_{k=1}^{K} J_k + \frac{1}{\lambda}D_{KL}(\hat{\theta}_{1:N,T}||\hat{\theta}_{1:N})$
    Run SGD step w.r.t $\hat{\theta}_{1:N,T}, \hat{\phi}_1, \ldots, \hat{\phi}_K$ on J
  return $\hat{\theta}_{1:N,T}$

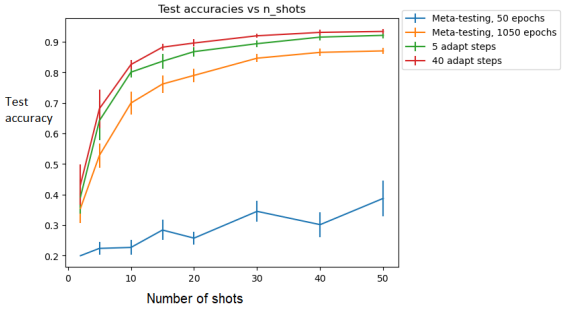
Since we tested our approach on neural networks, the hypothesis space is defined as $\{h(w)|w \in \mathbb{R}^d\}$, where the weights w serve as the parameters of each hypothesis. For image classification tasks, x represents an image, and y

a class label. We use the cross-entropy loss during adaptation, despite the fact that it is not bounded. We note that a clipped variation of this loss does exist, and conforms to theoretical guarantees (Dziugaite & Roy, 2018), and that in practice the cross-entropy loss tends to be low.

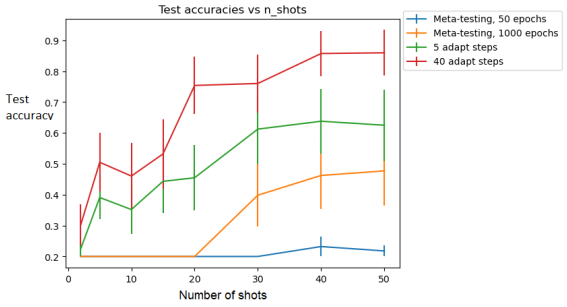
We first conduct experiments on a task environment based on the MNIST dataset (LeCun et al., 1998), where each task was created by performing a random permutation on some of the image pixels. The number of pixels to be shuffled and the total number of classes (referred to as "ways") was chosen in advance. In order to obtain a reasonable hyper-prior $\mathcal{Q}_{1:N}$ on downstream tasks, it makes sense to run standard meta-training methods on the training data. We used the MLAP (Amit & Meir, 2018) algorithm to do so. We ran MLAP-M for 100 meta-training iterations on randomly sampled training tasks, with 100 examples from each class. The resulting network means and variances were then used as the final meta-training hyper-prior $\mathcal{Q}_{1:N}$. We used the same convolutional neural network (CNN) architecture used by Vinyals et al. (2016) for the Omniglot dataset.

We perform tests on two sets of problems, tasks with 100 shuffled pixels, and tasks with permuted labels instead of pixels. For standard meta-testing, we sample a prior $P \sim \mathcal{Q}_{1:N}$ and perform 1000 adaptation steps given the labeled adaptation dataset S_T in order to achieve convergence. For our meta-adaptation method, we initialize the hyper-posterior as the meta-training hyper-prior $\mathcal{Q}_{1:N}$ and perform several steps of meta-adaptation as detailed in Algorithm 1. Following that, we sample a prior $P \sim \mathcal{Q}_{1:N,T}$ and perform 50 SGD steps of standard meta-testing. Since the meta-adaptation process itself includes computational costs, we use far fewer update steps for the sampled prior in order to have a fairer comparison. See appendix A.1 for a full list of hyper-parameters and implementation details.

Figure 2 shows the test accuracy averaged over 10 meta-testing seeds for both tasks. Numerical values for both tasks as well as standard error are reported in Table 1. Significant improvements for the Wilcoxon test ($p < 0.05$) are marked with *. Empirical results for the MNIST tasks show that meta-adaptation is preferable to standard meta-testing for relatively small adaptation set sizes by a notable margin.

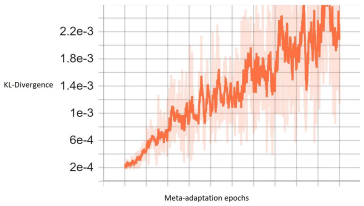


(a) Meta Adaptation - shuffled pixels

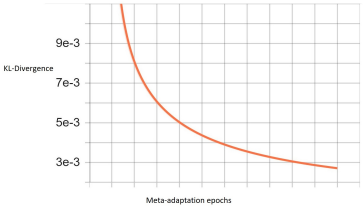


(b) Meta Adaptation - permuted labels

Figure 2: Average test accuracies on both MNIST tasks, for meta-adaptation with varying gradient updates. Error bars represent standard errors from the mean over 10 runs.



(a) $D_{\mathrm{KL}}(\mathcal{Q}_{1:N,T}||\mathcal{Q}_{1:N})$ over training epochs, smoothed.



(b) $D_{\mathrm{KL}}(\mathcal{Q}_{1:N,T}||\mathcal{N}(0, I_d))$ over training epochs.

Figure 3: KL-divergences during the meta-adaptation process for the mini-Imagenet dataset.

Table 2 compares test accuracies for the well-known mini-Imagenet (Vinyals et al., 2016) dataset. Results for MAML (Finn et al., 2017) and VAMPIRE (Nguyen et al., 2020) are from their respective papers, results for Bayesian variational inference (VI) and meta-adaptation are averaged over 5 seeds with standard error reported. Since fewer seeds were used, results have a higher standard error. While our empirical results are not better than existing SoTA meta-learning algorithms, they are comparable and show an improvement beyond standard meta-testing for relatively low adaptation set size $|S_T|$. Better hyper-parameter tuning may be needed in order to achieve improved results. As $|S_T|$

Table 1: Test accuracies for the meta-MNIST dataset.

Method	N_shots	Shuffled-pixels	Permuted-labels
MLAP-M	5	0.53 ± 0.04	0.2 ± 0
Meta-adaptation	5	$0.68^* \pm 0.06$	$0.51^* \pm 0.1$
MLAP-M	20	0.79 ± 0.02	0.2 ± 0.0
Meta-adaptation	20	$0.9^* \pm 0.01$	$0.75^* \pm 0.09$
MLAP-M	50	0.87 ± 0.01	0.48 ± 0.11
Meta-adaptation	50	$0.93^* \pm 0.01$	$0.86^* \pm 0.07$

Table 2: Test accuracies for the mini-Imagenet dataset.

Method	N_shots	Test accuracy %
MAML (Finn et al., 2017)	5	63.15 ± 0.91
VAMPIRE (Nguyen et al., 2020)	5	64.31 ± 0.74
Bayesian-VI	5	52.8 ± 3.6
Meta-adaptation	5	60 ± 2.6
Bayesian-VI	20	61.7 ± 2.2
Meta-adaptation	20	$64.9^* \pm 1.7$

increases, the training loss becomes a better estimate of the expected loss for the task, leading to a better base learner as well as more reliable improvements from meta-adaptation.

Figure 3 shows the typical progression of KL-divergence during the meta-adaptation fitting process. We see that $D_{\mathrm{KL}}(\mathcal{Q}_{1:N,T}||\mathcal{N}(0, I_d))$ decreases (alongside the training loss on the adaptation data), resulting in lower overall variance in network weights as well as a shift in the means. $D_{\mathrm{KL}}(\mathcal{Q}_{1:N,T}||\mathcal{Q}_{1:N})$ tends to increase slowly during meta-adaptation, resulting in a hyper-posterior that diverges from the original meta-learned hyper-prior as the training loss decreases. A different choice of hyper-parameters for λ may cause this term to diverge more slowly, leaving us with a more similar hyper-posterior.

4 Discussion and Future Work

We have derived several PAC-Bayes generalization bounds for meta-testing based on data-dependent bounds, and have demonstrated the efficacy of using the adaptation data in order to create a more appropriate hyper-posterior for a new test task. We have implemented a practical algorithm based on the derived bounds and have demonstrated its efficacy in few-shot image classification. To the best of our knowledge, our approach is the first meta-learning algorithm to consider adapting a learned hyper-prior in a principled approach.

While our experimental results are preliminary, they clearly demonstrate an improvement in terms of generalization error for the MNIST dataset, and results comparable to other well-known meta-learning algorithms (MAML, VAMPIRE) for the Omniglot and mini-Imagenet datasets with similar hypothesis classes. Future work should evaluate whether better parameter tuning can bridge the gap between meta-adaptation and SoTA results, as well as considering results for datasets with more pronounced distirbution shifts between training and testing.

There are two main open issues to be addressed in future work. The first issue is that our approach requires the use of stochastic models, which may lead to high-variance gradients and are therefore non-trivial to optimize. One potential method to address this issue is to make use of PAC-Bayes bounds with different complexity measures such as Wasserstein distance (Ohnishi & Honorio, 2021; Amit et al., 2022). The second issue is to consider our meta-adaptation algorithm in the context of continual learning (Kirkpatrick et al., 2017). Recent work by Haddouche & Guedj (2022) used data-dependent PAC-Bayes bounds in the context of online learning, and it would be of interest to extend these results to the continual setting and derive bounds that provide a clearer tradeoff between low generalization error and avoiding catastrophic forgetting.

ACKNOWLEDGEMENTS

This work was supported by the Israel Science Foundation grant number 1693/22.

REFERENCES

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended PAC-bayes theory. *35th International Conference on Machine Learning, ICML 2018*, 1:310–326, 2018.

Ron Amit, Baruch Epstein, Shay Moran, and Ron Meir. Integral probability metrics pac-bayes bounds. *arXiv preprint arXiv:2207.00614*, 2022.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Olivier Catoni. A PAC-Bayesian approach to adaptive classification. *preprint 840*, 2004.

Monroe D Donsker and SR Srinivasa Varadhan. On a variational formula for the principal eigenvalue for operators with maximum principle. *Proceedings of the National Academy of Sciences*, 72(3):780–783, 1975.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Uncertainty in Artificial Intelligence - Proceedings of the 33rd Conference, UAI 2017*, 2017.

Gintare Karolina Dziugaite and Daniel M. Roy. Data-dependent PAC-Bayes priors via differential privacy. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):8430–8441, 2018. ISSN 10495258.

Alec Farid and Anirudha Majumdar. Generalization bounds for meta-learning via pac-bayes and uniform stability. *Advances in Neural Information Processing Systems*, 34:2173–2186, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *34th International Conference on Machine Learning, ICML 2017*, volume 3, pp. 1856–1868. International Machine Learning Society (IMLS), mar 2017. ISBN 9781510855144. URL http://arxiv.org/abs/1703.03400.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Maxime Haddouche and Benjamin Guedj. Online PAC-Bayes Learning. *arXiv preprint arXiv:2206.00024*, may 2022. doi: 10.48550/arxiv.2206.00024. URL https://arxiv.org/abs/2206.00024v1http://arxiv.org/abs/2206.00024.

Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. ISSN 19393539. doi: 10.1109/TPAMI.2021.3079209.

Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791.

Tianyu Liu, Jie Lu, Zheng Yan, and Guangquan Zhang. Statistical generalization performance guarantee for meta-learning with data dependent prior. *Neurocomputing*, 465:391–405, 2021.

David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170, 1999.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103, 2007. doi: 10.1109/FOCS.2007.66.

Cuong Nguyen, Thanh-Toan Do, and Gustavo Carneiro. Uncertainty in model-agnostic meta-learning using variational inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3090–3100, 2020.

Yuki Ohnishi and Jean Honorio. Novel change of measure inequalities with applications to pac-bayesian bounds and monte carlo estimation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1711–1719. PMLR, 2021.

Anastasia Pentina and Christoph H Lampert. A PAC-bayesian bound for lifelong learning. In *31st International Conference on Machine Learning, ICML 2014*, volume 3, pp. 2656–2664, 2014. ISBN 9781634393973.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and DeepMind Edmonton. Tighter Risk Certificates for Neural Networks. *Journal of Machine Learning Research*, 22:1–40, 2021. URL http://jmlr.org/papers/v22/20-879.html.

Omar Rivasplata, Ilja Kuzborskij, Csaba Szepesvári, and John Shawe-Taylor. PAC-Bayes analysis beyond the usual bounds. In *Advances in Neural Information Processing Systems*, volume 2020-Decem, 2020.

Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. Pacoh: Bayes-optimal meta-learning with pac-guarantees. In *International Conference on Machine Learning*, pp. 9116–9126. PMLR, 2021.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3637–3645, 2016.

A Appendix

A.1 Hyper-parameters and implementation details

Algorithm 2 Standard Meta-testing

function META-TEST(θ, $S_T \sim \mathcal{D}_T$)
  Choose algorithmic parameters η_α, # steps, c
  Sample $\phi \sim \mathcal{N}(\theta)$
  while test step $<$ # steps **do** ▷ Proxy to convergence
    $J(\phi, S_T, \theta) = \hat{\mathcal{L}}(\phi, S_T) + D_{KL}(\phi||\theta) + c \cdot D_{KL}(\phi||(0, \sigma^2 I))$ ▷ $D_{KL}(\phi||(0, \sigma^2 I))$ is a low-norm condition on ϕ
    Take gradient step of size η_α w.r.t. ϕ on ∇J
  return ϕ

In the adaptive hyper-posterior version of the meta-adaptation algorithm, we first initialize the hyper-posterior $\hat{\mathcal{Q}}_{1:N,T}$ to the meta-training hyper-prior $\mathcal{Q}_{1:N}$. In each adaptation step, we optimize (4) w.r.t. $\hat{\mathcal{Q}}_{1:N,T}, \hat{Q}$ and perform gradient updates on the hyper-posterior. We then set the hyper-prior for the next adaptation step as the current hyper-posterior. In the constant hyper-prior version of the meta-adaptation algorithm, the hyper-prior remains the same throughout the meta-adaptation process.

The network architecture used for classification on the permuted MNIST dataset is identical to that used for the Omniglot dataset by Vinyals et al. (2016): 4 layers of 2D 3×3 convolution layers followed by batch norm and max pooling each. These are followed by a linear classification layer with 5 classes (commonly referred to as number of ways). The training loss used was standard cross-entropy loss. We used the Adam optimizer (Kingma & Ba, 2015) for meta-training and meta-testing. Code to reproduce the experiments is available at https://github.com/lioritan/meta-adapt-pb.

Table 3: Hyper-parameter choices, MNIST

Notation	Description	Value/s
#Training epochs	Number of training epochs for hyper-prior	100
Train Sample size	# of training examples per class per epoch	100
#Test epochs	Number of base learner gradient updates	$50, 1000$
η_α	Learning rate for the meta-learner	0.01
η_β	Learning rate for the base learner	0.1
#Adaptation steps	Number of training epochs for hyper-posterior	$5, 40$
MC-iterations	Number of Monte-Carlo samples to average	3
Test permutations	Number of pixels permuted on each test image	100
#ways	Number of classes in classification	5
#shots	Number of samples per class in S_T	$2, 5, 10, 15, 20, 30, 40, 50$
seed	Random seed, chosen arbitrarily	$42, 1337, 7, 13, 999, 752, 56789, 145790, 11, 306050$

A.2 Proof of the generic meta-adaptation bound

We restate Theorem 3.1 for stochastic kernels.

Theorem A.1. (PAC-Bayes for stochastic kernels - adapted from Theorem 2 in Rivasplata et al. (2020)) Let $P \in \mathcal{K}(\mathcal{Z}^m, \mathcal{H})$ be a stochastic kernel (namely, the prior may depend on the sample data S), let $A : \mathcal{Z}^m \times \mathcal{H} \to \mathbb{R}^k$ be a measurable function for some positive integer k and $F : \mathbb{R}^k \to \mathbb{R}$ be a convex function. Define $f \triangleq F \circ A$, and

$$\xi(P_S, \mathcal{D}, f) = \int_{\mathcal{Z}^m} \int_{\mathcal{H}} e^{f(S,h)} P_S(dh) \mathcal{D}(dS),$$

Table 4: Hyper-parameter choices, mini-Imagenet

Notation	Description	Value/s
#Training epochs	Number of training epochs for hyper-prior	20000
Train Sample size	# of training examples per class per epoch	20
#Test epochs	Number of base learner gradient updates	5
η_α	Learning rate for the meta-learner	0.001
η_β	Learning rate for the base learner	0.01
#Adaptation steps	Number of training epochs for hyper-posterior	$1000, 2000$
MC-iterations	Number of Monte-Carlo samples to average	10
#ways	Number of classes in classification	5
#shots	Number of samples per class in S_T	$5, 20$
seed	Random seed, chosen arbitrarily	$11, 13, 999, 56789, 306050$

such that P_S is the distribution over $\mathcal{H}$ corresponding to the sampled S. Assuming $\xi(P_S, \mathcal{D}, f)$ is finite, then for any $\delta \in (0,1)$ and any posterior $Q \in \mathcal{K}(\mathcal{Z}^m, \mathcal{H})$, with probability at least $1-\delta$ over the choice of $S \sim \mathcal{D}$,

$$\mathbb{E}_{h\sim Q_S}\left[f(S,h)\right] \leq D_{KL}(Q_S||P_S) + \log\left(\xi(P_S, \mathcal{D}, f)/\delta\right). \tag{9}$$

Next, we restate and prove Theorem 3.2.

Theorem A.2. Let $\mathcal{Q}_{1:N} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ be a meta-learned hyper-posterior (e.g. the result of meta-training on $\{S_1, ..., S_N\}$), and let $\mathcal{D}_T \sim \tau$ be a *given* test task. Let $Q : \mathcal{Z}^m \times \mathcal{M}(\mathcal{H}) \to \mathcal{M}(\mathcal{H})$ be a given base learner. Let $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$ be a bounded loss function. For any $\delta_T \in (0,1)$, any hyper-posterior $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$, and for all $\lambda > 0$, with probability at least $1-\delta_T$ over the draw of $S_T \sim \mathcal{D}_T$

$$\begin{aligned}\mathcal{L}(\mathcal{Q}, \mathcal{D}_T) \leq \hat{\mathcal{L}}(\mathcal{Q}, S_T) + \frac{1}{\lambda} D_{KL}(\mathcal{Q}||\mathcal{Q}_{1:N}) \\ + \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right),\end{aligned} \tag{10}$$

where

$$\begin{aligned}&\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T) \triangleq \\ &\mathbb{E}_{S\sim\mathcal{D}_T, P\sim\mathcal{Q}_{1:N}, h\sim Q(P,S)}\left[e^{\lambda\left(\mathcal{L}(h,\mathcal{D}_T) - \hat{\mathcal{L}}(h,S)\right)}\right].\end{aligned} \tag{11}$$

Proof. First, we consider the setting of Theorem A.1 and extend it to meta-learning. We define a stochastic kernel for the 2-level hypothesis case as a pair $(\mathcal{P}, P) \in \mathcal{K}(\mathcal{Z}^m, \mathcal{M}(\mathcal{H}) \times \mathcal{H})$ such that for a given sample $S \in \mathcal{Z}^m$, $(\mathcal{P}, P')(S)$ is the distribution over $\mathcal{H}$ corresponding to sampling from the hyper-prior $P \sim \mathcal{P}(S)$ corresponding to S and then sampling from $h \sim P'(S, P)$. For clarity, we denote these as distributions as $\mathcal{P}_S$ and $P'_{S,P}$.

Let $A : \mathcal{Z}^m \times \mathcal{H} \to \mathbb{R}^k$ be a measurable function for some positive integer k, and $F : \mathbb{R}^k \to \mathbb{R}$ a convex function. For $f \triangleq F \circ A$ let

$$\begin{aligned}&\xi((\mathcal{P}_S, P'_S), \mathcal{D}, f) = \\ &\int_{\mathcal{Z}^m}\int_{\mathcal{M}(\mathcal{H})\times\mathcal{H}} e^{f(S,h)}\left(\mathcal{P}_S \times P'_{S,P}\right)(dP, dh)\mathcal{D}(dS).\end{aligned}$$

Using (9) with $f(S,(P,h)) = \lambda(\mathcal{L}(h,\mathcal{D}) - \hat{\mathcal{L}}(h,S))$, a 2-level posterior $(\mathcal{Q}_S, Q_S)$ and a 2-level prior $(\mathcal{Q}_{1:N}, Q_S)$, we have that for any $\delta_T \in (0,1)$, the following inequality holds uniformly for all posteriors $Q \in \mathcal{K}(\mathcal{Z}^m, \mathcal{H})$ with probability at least $1-\delta_T$ over the choice of $S_T \sim \mathcal{D}_T$,

$$\begin{aligned}&\mathbb{E}_{h\sim(\mathcal{Q}_{S_T}, Q_{S_T})}\left[\lambda(\mathcal{L}(h,\mathcal{D}_T) - \hat{\mathcal{L}}(h,S_T))\right] \leq \\ &D_{KL}((\mathcal{Q}_{S_T}, Q_{S_T})||(\mathcal{Q}_{1:N}, Q_{S_T})) \\ &+ \log\left(\xi\left((\mathcal{Q}_{1:N}, Q_S), \mathcal{D}_T, \lambda(\mathcal{L}(h,\mathcal{D}) - \hat{\mathcal{L}}(h,S))\right)/\delta_T\right)\end{aligned} \tag{12}$$

It is important to note that the only assumption we make here is that the hyper-prior $\mathcal{Q}_{1:N}$ is data-free with respect to the new data $S_T \sim \mathcal{D}_T$. Since we consider meta-learned distributions over priors, this is a reasonable assumption, as it is satisfied if we have not seen S_T during meta-training.

First, let us make sure that the moment terms are equivalent. We note here that since the hyper-prior $\mathcal{Q}_{1:N}$ does not depend on S_T, the expectation can be easily decomposed.

$$\begin{aligned}
&\xi\left((\mathcal{Q}_{1:N}, Q_S), \mathcal{D}_T, f = \lambda(\mathcal{L}(h,\mathcal{D}) - \hat{\mathcal{L}}(h,S))\right) \\
&= \int_{\mathcal{Z}^m}\int_{\mathcal{H}} e^{f(S,h)}\mathcal{Q}_{1:N}(dP)Q(P,S)(dh)\mathcal{D}_T(dS), \\
&= \mathbb{E}_{S\sim\mathcal{D}_T, P\sim\mathcal{Q}_{1:N}, h\sim Q(P,S)}\left[e^{\lambda\left(\mathcal{L}(h,\mathcal{D}_T) - \hat{\mathcal{L}}(h,S)\right)}\right], \\
&\triangleq \tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T).
\end{aligned}$$

Substituting this in (12) and moving terms, we get (with probability at least $1-\delta_T$ over the draw of $S_T \sim \mathcal{D}_T$),

$$\begin{aligned}
&\mathbb{E}_{h\sim(\mathcal{Q}_{S_T}, Q_{S_T})}\left[\mathcal{L}(h,\mathcal{D}_T)\right] \leq \mathbb{E}_{h\sim(\mathcal{Q}_{S_T}, Q_{S_T})}\left[\hat{\mathcal{L}}(h,S_T)\right] \\
&+ \frac{1}{\lambda} D_{KL}((\mathcal{Q}_{S_T}, Q_{S_T})||(\mathcal{Q}_{1:N}, Q_{S_T})) \\
&+ \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right)
\end{aligned} \tag{13}$$

By definition, this is equivalent to writing

$$\begin{aligned}
\mathcal{L}(\mathcal{Q}_{S_T}, \mathcal{D}_T) \leq\ &\hat{\mathcal{L}}(\mathcal{Q}_{S_T}, S_T) \\
&+ \frac{1}{\lambda} D_{KL}((\mathcal{Q}_{S_T}, Q_{S_T})||(\mathcal{Q}_{1:N}, Q_{S_T})) \\
&+ \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right).
\end{aligned} \tag{14}$$

For the KL-divergence expression, we apply a standard KL decomposition

$$\begin{aligned}
&D_{KL}((\mathcal{Q}_{S_T}, Q_{S_T})||(\mathcal{Q}_{1:N}, Q_{S_T})) \\
&= \mathbb{E}_{(P,h)\sim(\mathcal{Q}_{S_T}, Q_{S_T})}\left[\log\frac{\mathcal{Q}_{S_T}(P)Q(P,S_T)(h)}{\mathcal{Q}_{1:N}(P)Q(P,S_T)(h)}\right] \\
&= \mathbb{E}_{P\sim\mathcal{Q}_{S_T}}\mathbb{E}_{h\sim Q(P,S_T)}\left[\log\frac{\mathcal{Q}_{S_T}(P)}{\mathcal{Q}_{1:N}(P)}\right] \\
&= \mathbb{E}_{P\sim\mathcal{Q}_{S_T}}\left[\log\frac{\mathcal{Q}_{S_T}(P)}{\mathcal{Q}_{1:N}(P)}\right] \\
&= D_{KL}(\mathcal{Q}_{S_T}||\mathcal{Q}_{1:N}).
\end{aligned}$$

Finally, combining this result with (14) gives us the final inequality:

For any $\delta_T \in (0,1)$, the following holds uniformly for all hyper-posteriors $\mathcal{Q}_{S_T} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ with probability at least $1-\delta_T$ over the draw of $S_T \sim \mathcal{D}_T$:

$$\begin{aligned}
\mathcal{L}(\mathcal{Q}_{S_T}, \mathcal{D}_T)) \leq\ &\hat{\mathcal{L}}(\mathcal{Q}_{S_T}, S_T) + \frac{1}{\lambda} D_{KL}(\mathcal{Q}_{S_T}||\mathcal{Q}_{1:N}) \\
&+ \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right).
\end{aligned}$$

□

A.3 TEST TASK BOUND FOR A META-LEARNED HYPER-PRIOR

Corollary A.3. Under the same conditions as Theorem 3.2, for any $\delta_T \in (0,1)$, and for all $\lambda > 0$, with probability at least $1-\delta_T$ over the draw of $S_T \sim \mathcal{D}_T$

$$\begin{aligned}\mathcal{L}(\mathcal{Q}_{1:N}, \mathcal{D}_T) \leq \hat{\mathcal{L}}(\mathcal{Q}_{1:N}, S_T) \\ + \frac{1}{\lambda}\log\left(\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)/\delta_T\right),\end{aligned} \tag{15}$$

where $\tilde{\xi}(\lambda, \mathcal{Q}_{1:N}, \mathcal{D}_T)$ is as defined in Theorem 3.2.

Proof. Since Theorem 3.2 applies for any hyper-posterior, it specifically applies for $\mathcal{Q} = \mathcal{Q}_{1:N}$. Making this substitution in (3) arrives at (15). □

A.4 HYPER-PRIORS WITH GIBBS BASE LEARNERS ARE DIFFERENTIALLY-PRIVATE

Definition A.1. (Differential privacy) (Dwork et al., 2006) Let $S, S' \in \mathcal{Z}^m$ be datasets that differ by a single element. A randomized algorithm $\mathcal{A}$ is called ϵ-differentially private, denoted $DP(\epsilon)$, if for any $I \subset \text{image}(\mathcal{A})$

$$\Pr(\mathcal{A}(S) \in I) \leq e^{\epsilon}\Pr(\mathcal{A}(S') \in I).$$

An equivalent definition for stochastic kernels that is easier to understand for our setting is the following.

Definition A.2. (Differential privacy) Let $S, S' \in \mathcal{Z}^m$ be datasets that differ by a single element. Let $Q \in \mathcal{K}(\mathcal{Z}^m, \mathcal{M}(\mathcal{H}))$ be a stochastic kernel. Q is $DP(\epsilon)$ if

$$\frac{Q(S,A)}{Q(S',A)} \leq e^{\epsilon}, \quad \forall A \in \mathcal{M}(\mathcal{H}).$$

Proposition A.4. Let $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ be a hyper-prior. Let $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$ be a bounded loss function. If the base learner $Q \in \mathcal{M}(\mathcal{H})$ is the Gibbs posterior $Q(P,S)(h) \propto P(h)e^{-\beta\hat{\mathcal{L}}(h,S)}$, then the pair hypothesis $(\mathcal{P}, Q)$ satisfies $DP\left(\frac{2\beta}{m}\right)$.

Proof. From Theorem 6 in McSherry & Talwar (2007), the Gibbs posterior $Q(P,S)(h)$ satisfies $DP\left(2\beta\Delta L\right)$, where ΔL is the the largest possible difference $\sup_{h\in\mathcal{H}}[\hat{\mathcal{L}}(h,S) - \hat{\mathcal{L}}(h,S')]$ for S, S' that differ by one example. Since the loss is bounded in $[0,1]$ and S, S' are of size m, we have $\Delta L \leq \frac{1}{m}$, and so the base learner $Q(P,S)(h)$ satisfies $DP\left(\frac{2\beta}{m}\right)$.

It remains to prove that for all $(A_1, A_2) \in (\mathcal{M}(\mathcal{M}(\mathcal{H})), \mathcal{M}(\mathcal{H}))$,

$$\frac{\mathcal{P}(S,A_1)Q(S,A_2)}{\mathcal{P}(S',A_1)Q(S',A_2)} \leq e^{\frac{2\beta}{m}}.$$

From the DP property,

$$\frac{\mathcal{P}(S,A_1)Q(S,A_2)}{\mathcal{P}(S',A_1)Q(S',A_2)} \leq \frac{\mathcal{P}(S,A_1)}{\mathcal{P}(S',A_1)}e^{\frac{2\beta}{m}}.$$

Since $\mathcal{P}$ is a hyper-prior, we assume it is data-free with respect to S, and so

$$\mathcal{P}(S,A_1) = \mathcal{P}(S',A_1) = \mathcal{P}(A_1).$$

□