PACOH: Bayes-Optimal Meta-Learning with PAC-Guarantees

Jonas Rothfuss [1] **Vincent Fortuin** [1] **Martin Josifoski** [2] **Andreas Krause** [1]

Abstract

Meta-learning can successfully acquire useful inductive biases from data. Yet, its generalization properties to unseen learning tasks are poorly understood. Particularly if the number of meta-training tasks is small, this raises concerns about overfitting. We provide a theoretical analysis using the *PAC-Bayesian* framework and derive novel *generalization bounds* for meta-learning. Using these bounds, we develop a class of PAC-optimal meta-learning algorithms with performance guarantees and a principled *meta-level regularization*. Unlike previous PAC-Bayesian meta-learners, our method results in a standard stochastic optimization problem which can be solved efficiently and *scales well*. When instantiating our *PAC-optimal hyper-posterior (PACOH)* with Gaussian processes and Bayesian Neural Networks as base learners, the resulting methods yield state-of-the-art performance, both in terms of predictive accuracy and the quality of uncertainty estimates. Thanks to their principled treatment of uncertainty, our meta-learners can also be successfully employed for *sequential decision problems*.

1. Introduction

Meta-learning aims to extract prior knowledge from data, accelerating the learning process for new learning tasks (Thrun & Pratt, 1998). Most existing meta-learning approaches focus on situations where the number of tasks is large (e.g., Finn et al., 2017; Garnelo et al., 2018). In many practical settings, however, the number of tasks available for meta-training is rather small. In those settings, there is a risk of *overfitting to the meta-training tasks* (meta-overfitting, cf. Qin et al., 2018), thus impairing the performance on yet unseen target tasks. Hence, a key challenge is how to *regularize* the meta-learner to ensure its *generalization to unseen tasks*.

[1]ETH Zurich, Switzerland [2]EPFL, Switzerland. Correspondence to: Jonas Rothfuss <jonas.rothfuss@inf.ethz.ch>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

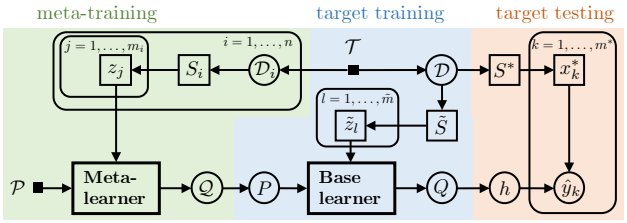

Figure 1. Overview of the described meta-learning framework with hyper-prior $\mathcal{P}$, hyper-posterior $\mathcal{Q}$, target prior P and target posterior Q. The data-generating process in our setup is described by a meta-learning environment $\mathcal{T}$, meta-train task distributions $\mathcal{D}_i$, target task distribution $\mathcal{D}$, data sets S and data points $z = (x, y)$.

The PAC-Bayesian framework provides a rigorous way to reason about the generalization performance of learners (McAllester, 1999). However, initial PAC-Bayesian analyses of meta-learners (Pentina & Lampert, 2014; Amit & Meir, 2018) only consider *bounded* loss functions, which precludes important applications such as regression or probabilistic inference, where losses are typically unbounded. More crucially, they rely on a challenging nested optimization problem, which is computationally much more expensive than standard meta-learning approaches.

To overcome these issues, we derive the first *PAC-Bayesian bound* for meta-learners with *unbounded loss* functions. For Bayesian learners, we further tighten our PAC-Bayesian bounds, relating them directly to the marginal log-likelihood of the Bayesian model and thus *avoiding the reliance on nested optimization*. This allows us to derive the *PAC-optimal hyper-posterior (PACOH)*, which promises strong performance guarantees and a principled meta-level regularization. Importantly, it can be approximated using standard variational methods (Blei et al., 2016), giving rise to a range of *scalable* meta-learning algorithms.

In our experiments, we instantiate our framework with Gaussian Processes (GPs) and Bayesian Neural Networks (BNNs) as base learners. Across several regression and classification environments, our proposed approach achieves *state-of-the-art* predictive *accuracy*, while also improving the *calibration* of the uncertainty estimates. Further, we demonstrate that *PACOH* effectively *alleviates the meta-overfitting problem*, allowing us to successfully extract inductive bias from as little as five tasks while reliably reasoning about the learner's epistemic uncertainty.

Thanks to these properties, *PACOH* can also be employed in a broad range of *sequential decision problems*, which we showcase through a real-world Bayesian optimization task concerning the development of vaccines.

2. Related Work

Meta-learning. A range of methods in meta-learning attempt to learn the "learning program" in form of a recurrent model (Hochreiter et al., 2001; Andrychowicz et al., 2016; Chen et al., 2017), learn an embedding space shared across tasks (Snell et al., 2017; Vinyals et al., 2016) or the initialization of a NN so it can be quickly adapted to new tasks (Finn et al., 2017; Nichol et al., 2018; Rothfuss et al., 2019b). A group of recent methods also use probabilistic modeling to enable uncertainty quantification (Kim et al., 2018; Finn et al., 2018; Garnelo et al., 2018). Though the mentioned approaches are able to learn complex inference patterns, they rely on settings where meta-training tasks are abundant and fall short of performance guarantees. While the risk of meta-overfitting has previously been noted (Qin et al., 2018; Fortuin & Rätsch, 2019; Yin et al., 2020), it still lacks a rigorous formal analysis under realistic assumptions (e.g., unbounded loss functions). Addressing this issue, we study the generalization properties of meta-learners within the PAC-Bayesian framework, and, based on that, contribute a novel meta-learning approach with principled meta-level regularization.

PAC-Bayesian theory. Previous work presents generalization bounds for randomized predictors, assuming a prior to be given exogenously (McAllester, 1999; Catoni, 2007; Germain et al., 2016; Alquier et al., 2016). Further work explores data-dependent priors (Parrado-Hernandez et al., 2012; Dziugaite & Roy, 2016) or extends previous bounds to the scenario where priors are meta-learned (Pentina & Lampert, 2014; Amit & Meir, 2018). However, these meta-generalization bounds are hard to optimize as they leave both the hyper-posterior and posterior unspecified, which leads to difficult nested optimization problems. In contrast, our bounds also hold for unbounded losses and yield a *tractable meta-learning objective* without the reliance on nested optimization.

3. Background: PAC-Bayesian Framework

Preliminaries and notation. A learning task is characterized by an unknown data distribution $\mathcal{D}$ over a domain $\mathcal{Z}$ from which we are given a set of m observations $S = \{z_i\}_{i=1}^m$, $z_i \sim \mathcal{D}$. By $S \sim \mathcal{D}^m$ we denote the i.i.d. sampling of m data points. In supervised learning, we typically consider pairs $z_i = (x_i, y_i)$, where $x_i \in \mathcal{X}$ are observed input features and $y_i \in \mathcal{Y}$ are target labels. Given a sample S, our goal is to find a hypothesis $h \in \mathcal{H}$, typically a function $h : \mathcal{X} \to \mathcal{Y}$ in some hypothesis space $\mathcal{H}$, that enables us to make predictions for new inputs $x^* \sim \mathcal{D}_x$. The quality of the predictions is measured by a *loss function* $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$. Accordingly, we want to minimize the *expected error* under the data distribution, that is, $\mathcal{L}(h, \mathcal{D}) = \mathbb{E}_{z^* \sim \mathcal{D}}\, l(h, z^*)$. Since $\mathcal{D}$ is unknown, we typically use the *empirical error* $\hat{\mathcal{L}}(h, S) = \frac{1}{m} \sum_{i=1}^m l(h, z_i)$ instead.

In the PAC-Bayesian framework, we consider *randomized predictors*, i.e., probability measures on the hypothesis space $\mathcal{H}$. This allows us to reason about the predictor's (epistemic) uncertainty, resulting from the fact that only a finite number of data points are available for training. We consider two such probability measures, the *prior* $P \in \mathcal{M}(\mathcal{H})$ and the *posterior* $Q \in \mathcal{M}(\mathcal{H})$. Here, $\mathcal{M}(\mathcal{H})$ denotes the set of all probability measures on $\mathcal{H}$. While in Bayesian inference, the prior and posterior are tightly connected through Bayes' theorem, the PAC-Bayesian framework makes fewer assumptions and only requires the prior to be independent of the observed data (Guedj, 2019). In the following, we assume that the Kullback-Leibler (KL) divergence $D_{KL}\left(Q \| P\right)$ exists. Based on the error definitions above, we can define the so-called *Gibbs error* for a randomized predictor Q as $\mathcal{L}(Q, \mathcal{D}) = \mathbb{E}_{h \sim Q}\, \mathcal{L}(h, \mathcal{D})$ and its empirical counterpart as $\hat{\mathcal{L}}(Q, S) = \mathbb{E}_{h \sim Q}\, \hat{\mathcal{L}}(h, S)$.

PAC-Bayesian bounds. In practice, $\mathcal{L}(Q, \mathcal{D})$ is unknown. Thus, one typically resorts to minimizing $\hat{\mathcal{L}}(Q, S)$ instead. However, this may result in overfitting and poor generalization. Naturally, the question arises whether we can bound the unknown generalization error based on its empirical estimate. The PAC-Bayesian theory provides such a guarantee:

Theorem 1. *(Alquier et al., 2016) Given a data distribution $\mathcal{D}$, hypothesis space $\mathcal{H}$, loss function $l(h, z)$, prior P, confidence level $\delta \in (0, 1]$, and $\beta > 0$, with probability at least $1 - \delta$ over samples $S \sim \mathcal{D}^m$, we have for all $Q \in \mathcal{M}(\mathcal{H})$:*

$$\mathcal{L}(Q, \mathcal{D}) \leq \hat{\mathcal{L}}(Q, S) + \frac{1}{\beta}\left[D_{KL}(Q||P) + \ln\frac{1}{\delta} + \Psi(\beta, m)\right] \tag{1}$$

$$\text{with } \Psi(\beta, m) = \ln \mathbb{E}_P \mathbb{E}_{\mathcal{D}^m} \exp\left[\beta\left(\mathcal{L}(h, \mathcal{D}) - \hat{\mathcal{L}}(h, S)\right)\right]$$

Here, $\Psi(\beta, m)$ is a log moment generating function that quantifies how much the empirical error deviates from the Gibbs error. By making additional assumptions about the loss function l, we can bound $\Psi(\beta, m)$ and thereby obtain tractable bounds. For instance, if $l(h, z)$ is *bounded* in $[a, b]$, we can use Hoeffding's lemma to obtain $\Psi(\beta, m) \leq (\beta^2(b-a)^2)/(8m)$. For unbounded loss functions, one commonly assumes that their moments are bounded. In particular, a loss function l is considered *sub-gamma* with variance factor s^2 and scale parameter c, under a prior π and data distribution $\mathcal{D}$, if it can be described by a sub-gamma random variable $V := \mathcal{L}(h, \mathcal{D}) - l(h, z)$. That is, its moment generating function is upper-bounded by that of a Gamma distribution $\Gamma(s, c)$. For details see Boucheron et al. (2013) and Germain et al. (2016). We can use the sub-gamma

assumption to obtain $\Psi(\beta, m) \leq (\beta^2 s^2)/(2m(1 - \frac{c\beta}{m}))$.

Connections between the PAC-Bayesian framework and Bayesian inference. Typically, we are interested in a posterior distribution Q that promises us the lowest generalization error. In this sense, it seems natural to use the $Q \in \mathcal{M}(\mathcal{H})$ that minimizes the bound in (1). Lemma 1 gives us the closed-form solution to such a minimization problem:

Lemma 1. *(Catoni, 2007) Let $\mathcal{H}$ be a set, $g : \mathcal{H} \to \mathbb{R}$ a (loss) function, and $Q \in \mathcal{M}(\mathcal{H})$ and $P \in \mathcal{M}(\mathcal{H})$ probability densities over $\mathcal{H}$. Then, for any $\beta > 0$ and $h \in \mathcal{H}$,*

$$Q^*(h) := \frac{P(h)e^{-\beta g(h)}}{Z_\beta} = \frac{P(h)e^{-\beta g(h)}}{\mathbb{E}_{h\sim P}\left[e^{-\beta g(h)}\right]} \tag{2}$$

is the minimizing probability density of

$$\underset{Q\in\mathcal{M}(\mathcal{H})}{\arg\min}\ \beta\mathbb{E}_{h\sim Q}\left[g(h)\right] + D_{KL}(Q||P)\ . \tag{3}$$

The respective minimizing distribution is known as *optimal Gibbs posterior* Q^* (Catoni, 2007; Lever et al., 2013). As a direct consequence of Lemma 1, for fixed P, S, m, δ, we can write the minimizer of (1) as

$$Q^*(h) = P(h)e^{-\beta\hat{\mathcal{L}}(h,S)}/Z_\beta(S, P)$$

where $Z_\beta(S, P) = \int_{\mathcal{H}} P(h)e^{-\beta\hat{\mathcal{L}}(h,S)}dh$ is a normalization constant. In a probabilistic setting, we typically use the negative log-likelihood of the data as our loss function $l(\cdot)$, that is, $l(h, z_i) := -\log p(z_i|h)$. In this case, the optimal Gibbs posterior coincides with the *generalized Bayesian posterior* $Q^*(h; P, S) = \frac{P(h)\, p(S\,|\,h)^{\beta/m}}{Z_\beta(S,P)}$ where $Z_\beta(S, P) = \int_{\mathcal{H}} P(h)\left(\prod_{j=1}^m p(z_j|h)\right)^{\beta/m} dh$ is called the *generalized marginal likelihood* of the sample S (Guedj, 2019). For $\beta = m$ we recover the standard Bayesian posterior.

4. PAC-Bayesian Bounds for Meta-Learning

We now present our main theoretical contributions. The corresponding proofs can be found in Appendix A. An overview of our proposed framework is depicted in Figure 1.

Meta-Learning. In the standard learning setup (Sec. 3), we assumed that the learner has prior knowledge in the form of a prior distribution P. When the learner faces a new task, it uses the evidence in the form of a dataset S, to update the prior into a posterior Q. We formalize such a *base learner* $Q(S, P)$ as a mapping $Q : \mathcal{Z}^m \times \mathcal{M}(\mathcal{H}) \to \mathcal{M}(\mathcal{H})$ that takes in a dataset and prior and outputs a posterior.

In contrast, in meta-learning we aim to acquire such a prior P in a *data-driven manner*, that is, by consulting a set of n statistically related learning tasks $\{\tau_1, ..., \tau_n\}$. We follow the setup of Baxter (2000) in which all tasks $\tau_i := (\mathcal{D}_i, S_i)$ share the same data domain $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, hypothesis space $\mathcal{H}$ and loss function $l(h, z)$, but may differ in their (unknown) data distributions $\mathcal{D}_i$ and the number of points m_i in the corresponding dataset $S_i \sim \mathcal{D}_i^{m_i}$. To simplify our exposition, we assume w.l.o.g. that $m = m_i\ \forall i$. Furthermore, each task $\tau_i \sim \mathcal{T}$ is drawn i.i.d. from an *environment* $\mathcal{T}$, a probability distribution over data distributions and datasets. The goal is to *extract knowledge from the observed datasets, which can then be used as a prior for learning on new target tasks* $\tau \sim \mathcal{T}$. To extend the PAC-Bayesian analysis to the meta-learning setting, we again consider the notion of probability distributions over hypotheses. While the object of learning has previously been a hypothesis $h \in \mathcal{H}$, it is now the prior distribution $P \in \mathcal{M}(\mathcal{H})$. The meta-learner presumes a *hyper-prior* $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$, that is, a distribution over priors P. Combining the hyper-prior $\mathcal{P}$ with the datasets $S_1, ..., S_n$ from multiple tasks, the meta-learner then outputs a *hyper-posterior* $\mathcal{Q}$ over priors. Accordingly, the hyper-posterior's performance is measured via the expected Gibbs error when sampling priors P from $\mathcal{Q}$ and applying the base learner, the so-called *transfer-error:*

$$\mathcal{L}(\mathcal{Q}, \mathcal{T}) := \mathbb{E}_{P\sim\mathcal{Q}}\left[\mathbb{E}_{(\mathcal{D},m)\sim\mathcal{T}}\left[\mathbb{E}_{S\sim\mathcal{D}^m}\left[\mathcal{L}(Q(S, P), \mathcal{D})\right]\right]\right]$$

While the transfer error is unknown in practice, we can estimate it using the *empirical multi-task error*

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

PAC-Bayesian meta-learning bounds. We now present our first main result: An upper bound on the true transfer error $\mathcal{L}(\mathcal{Q}, \mathcal{T})$, in terms of the empirical multi-task error $\hat{\mathcal{L}}(\mathcal{Q}, S_1, ..., S_n)$ plus several tractable complexity terms.

Theorem 2. *Let $Q : \mathcal{Z}^m \times \mathcal{M}(\mathcal{H}) \to \mathcal{M}(\mathcal{H})$ be a base learner, $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ some fixed hyper-prior $\mathcal{P}$ and $\lambda, \beta > 0$. For any confidence level $\delta \in (0, 1]$ the inequality*

$$\begin{aligned}\mathcal{L}(\mathcal{Q}, \mathcal{T}) \leq \hat{\mathcal{L}}(\mathcal{Q}, S_1, ..., S_n) + \left(\frac{1}{\lambda} + \frac{1}{n\beta}\right) D_{KL}(\mathcal{Q}||\mathcal{P}) \\ + \frac{1}{n}\sum_{i=1}^n \frac{1}{\beta}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q(S_i, P)||P)\right] \\ + C(\delta, \lambda, \beta)\end{aligned} \tag{4}$$

holds uniformly over all hyper-posteriors $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ with probability $1 - \delta$. If the loss function is bounded, that is, $l : \mathcal{H} \times \mathcal{Z} \to [a, b]$, the above inequality holds for $C(\delta, \lambda, \beta) = \left(\frac{\lambda}{8n} + \frac{\beta}{8m}\right)(b - a)^2 + \frac{1}{\sqrt{n}}\ln\frac{1}{\delta}$. If the loss function is sub-gamma with variance factor s_I^2 and scale parameter c_I for the data distributions $\mathcal{D}_i$ and s_{II}^2, c_{II} for the task distribution $\mathcal{T}$, the inequality holds with $C(\delta, \lambda, \beta) = \frac{\lambda s_{II}^2}{2n(1-(c_{II}\lambda)/n)} + \frac{\beta s_I^2}{2m(1-(c_I\beta)/m)} + \frac{1}{\sqrt{n}}\ln\frac{1}{\delta}$.

The proof of Theorem 2 consists of three main steps (see Appx. A.1). First, we bound the base learner's expected

generalization error for each task when given a prior $P \sim \mathcal{Q}$ and m_i data samples (i.e., the error caused by observing only a finite number of samples per task). Second, we bound the generalization gap on the meta-level which is due to the fact that the meta-learning only receives finitely many tasks τ_i from $\mathcal{T}$. In these two steps, we employ the change of measure inequality (Lemma 2) to obtain an upper bound of the generalization error in terms of the empirical error, the KL-divergence and a cumulant-generating function. In the final step, we use either use the bounded loss or sub-gamma assumption, together with Markov's inequality, to bound the cumulant-generating function with high probability, giving rise to $C(\delta, \lambda, \beta)$.

For bounded losses, Theorem 2 provides a structurally similar, but tighter bound than Pentina & Lampert (2014). In particular, by using an improved proof technique, we are able to avoid a union bound argument, allowing us to reduce the negative influence of the confidence parameter δ. Unlike in the bound of Amit & Meir (2018), the contribution of the KL-complexity term $D_{KL}(\mathcal{Q}||\mathcal{P})$ vanishes as $n \to \infty$. As a result, we find that the bound in Theorem 2 is much tighter than (Amit & Meir, 2018) for most practical instantiations. In contrast to Pentina & Lampert (2014) and Amit & Meir (2018), our theorem also provides guarantees for *unbounded* loss functions under moment constraints (see Appendix A.1 for details). This makes Theorem 2 particularly useful for probabilistic models in which the loss function coincides with the inherently unbounded negative log-likelihood.

Common choices for λ and β are either a) $\lambda = \sqrt{n}$, $\beta = \sqrt{m}$ or b) $\lambda = n$, $\beta = m$ (Germain et al., 2016). Choice a) yields consistent bounds, meaning that the gap between the transfer error and the bound vanishes as $n, m \to \infty$. In case of b), the bound always maintains a gap since $C(\delta, n, m)$ does not converge to zero. However, the KL terms decay faster, which can be advantageous for smaller sample sizes. For instance, despite their lack of consistency, sub-gamma bounds with $\beta = m$ have been shown to be much tighter in simple Bayesian linear regression scenarios with limited data ($m \lesssim 10^4$) (Germain et al., 2016).

Previous work (Pentina & Lampert, 2014; Amit & Meir, 2018) proposes meta-learning algorithms that minimize uniform meta-generalization bounds like the one in Theorem 2. However, in practice, the posterior $Q(S, P)$ is often intractable and thus the solution of a numerical optimization problem, like variational inference in case of BNNs. Hence, minimizing the bound in (4) turns into a difficult two-level optimization problem which becomes nearly infeasible to solve for rich hypothesis spaces such as neural networks.

While Theorem 2 holds for any base learner $Q(S, P)$, we would preferably want to use a base learner that gives us *optimal performance guarantees*. As discussed in Sec. 3, the Gibbs posterior not only minimizes PAC-Bayesian error bounds, but also generalizes the Bayesian posterior. Assuming a Gibbs posterior as base learner, the bound in (4) can be restated in terms of the partition function $Z_\beta(S_i, P)$:

Corollary 1. *When choosing a Gibbs posterior* $Q^*(S_i, P) := P(h)\exp(-\beta\hat{\mathcal{L}}(S_i, h))/Z_\beta(S_i, P)$ *as a base learner, under the same assumptions as in Theorem 2, we have*

$$\mathcal{L}(\mathcal{Q}, \mathcal{T}) \leq -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\beta}\mathbb{E}_{P\sim\mathcal{Q}}\left[\ln Z_\beta(S_i, P)\right] + \left(\frac{1}{\lambda} + \frac{1}{n\beta}\right) D_{KL}(\mathcal{Q}||\mathcal{P}) + C(\delta, \lambda, \beta)\ . \tag{5}$$

with probability at least $1 - \delta$.

Since this bound assumes a *PAC-optimal base learner*, it is tighter than the bound in (4), which holds for any (potentially sub-optimal) $Q \in \mathcal{M}(\mathcal{H})$. More importantly, (5) avoids the explicit dependence on $Q(S, P)$, *turning the previously mentioned two-level optimization problem into a standard stochastic optimization problem*. Moreover, if we choose the negative log-likelihood as the loss function and $\lambda = n, \beta_i = m_i$, then $\ln Z_\beta(S_i, P)$ *coincides with the marginal log-likelihood*, which is tractable for various popular learning models, such as Gaussian processes.

The bound in Corollary 1 consists of the expected generalized marginal log-likelihood under the hyper-posterior $\mathcal{Q}$ as well as the KL-divergence term, which serves as a *regularizer on the meta-level*. As the number of training tasks n grows, the relative weighting of the KL term shrinks. This is consistent with the general notion that regularization should be strong if only little data is available and vanish asymptotically as $n, m \to \infty$.

Meta-Learning the hyper-posterior. A natural way to obtain a PAC-Bayesian meta-learning algorithm could be to minimize the bound in Corollary 1 w.r.t. $\mathcal{Q}$. However, we can even derive the closed-form solution of such PAC-Bayesian meta-learning problem, that is, the minimizing hyper-posterior $\mathcal{Q}^*$. For that, we exploit once more the insight that the minimizer of (5) can be written as a Gibbs distribution (cf. Lemma 1), giving us the following result:

Proposition 1. ***(PAC-Optimal Hyper-Posterior)*** *Given a hyper-prior* $\mathcal{P}$ *and datasets* $S_1, ..., S_n$*, the hyper-posterior minimizing the meta-learning bound in (5) is given by*

$$\mathcal{Q}^*(P) = \frac{\mathcal{P}(P)\exp\left(\frac{\lambda}{n\beta+\lambda}\sum_{i=1}^{n}\ln Z_\beta(S_i, P)\right)}{Z^{II}(S_1, ..., S_n, \mathcal{P})} \tag{6}$$

with $Z^{II} = \mathbb{E}_{P\sim\mathcal{P}}\left[\exp\left(\frac{\lambda}{n\beta+\lambda}\sum_{i=1}^{n}\ln Z_\beta(S_i, P)\right)\right]$.

This gives us a tractable expression for the *PACOH* $\mathcal{Q}^*(P)$ up to the (level-II) partition function Z^{II}, which is constant with respect to P. We refer to $\mathcal{Q}^*$ as PAC-optimal, as it provides the best possible meta-generalization guarantees among all meta-learners in the sense of Theorem 2.

Algorithm 1 PACOH with SVGD approximation of $\mathcal{Q}^*$

Input: hyper-prior $\mathcal{P}$, datasets $S_1, ..., S_n$
Input: SVGD kernel function $k(\cdot, \cdot)$, step size η
$\boldsymbol{\phi} := [\phi_1, ..., \phi_K]$, with $\phi_k \sim \mathcal{P}$ // init prior particles
while not converged **do**
  for $k = 1, ..., K$ **do**
    for $i = 1, ..., n$ **do**
      $\ln Z_{i,k} \leftarrow \texttt{MLL\_Estimator}(S_i, \phi_k, \beta)$
    $\nabla_{\phi_k} \ln \tilde{\mathcal{Q}}^* \leftarrow \nabla_{\phi_k} \ln \mathcal{P} + \frac{\lambda}{\lambda + n\beta} \sum_{i=1}^{n} \nabla_{\phi_k} \ln Z_{i,k}$
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \eta\, \mathbf{K}\, \nabla_{\boldsymbol{\phi}} \ln \tilde{\mathcal{Q}}^* + \nabla_{\boldsymbol{\phi}} \mathbf{K}$ // SVGD update
Output: set of priors $\{P_{\phi_1}, ..., P_{\phi_K}\}$

5. Meta-Learning using the PACOH

After having introduced the closed-form solution of the PAC-Bayesian meta-learning problem in Sec. 4, we now discuss how to translate the *PACOH* into a practical meta-learning algorithm when employing GPs and BNNs as base learners.

5.1. Approximating the PACOH

Given the hyper-prior and (level-I) log-partition function $\ln Z(S_i, P)$, we can compute the PACOH $\mathcal{Q}^*$ up to the normalization constant Z^{II}. Such a setup lends itself to approximate inference methods (Blei et al., 2016). In particular, we employ *Stein Variational Gradient Descent (SVGD)* (Liu & Wang, 2016) which approximates $\mathcal{Q}^*$ as a set of particles $\hat{\mathcal{Q}} = \{P_{\phi_1}, ..., P_{\phi_K}\}$[1]. Here, P_ϕ denotes a prior with parameters ϕ. Alg. 1 summarizes the resulting generic meta-learning procedure. Initially, we sample K particles $\phi_k \sim \mathcal{P}$ from the hyper-posterior. For notational brevity, we stack the particles into a $K \times \dim(\phi)$ matrix $\boldsymbol{\phi} := [\phi_1, ..., \phi_K]^\top$. Then, in each iteration, we estimate the score of $\mathcal{Q}^*$,

$$\nabla_{\phi_k} \ln \mathcal{Q}^*(\phi_k) = \nabla_{\phi_k} \ln \mathcal{P}(\phi_k) + \frac{\lambda}{n\beta + \lambda} \sum_{i=1}^{n} \nabla_{\phi_k} \ln Z_{i,k}$$

wherein $\ln Z_{i,k} = \ln Z_\beta(S_i, P_{\phi_k})$, and update the particle matrix using the SVGD update rule:

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \eta\, \mathbf{K}\, \nabla_{\boldsymbol{\phi}} \ln \tilde{\mathcal{Q}}^* + \nabla_{\boldsymbol{\phi}} \mathbf{K}\,; \tag{7}$$

where $\nabla_{\boldsymbol{\phi}} \ln \tilde{\mathcal{Q}}^* := [\nabla_{\phi_1} \ln \mathcal{Q}^*(\phi_1), ..., \nabla_{\phi_K} \ln \mathcal{Q}^*(\phi_K)]^\top$ denotes the matrix of stacked score gradients, $\mathbf{K} := [k(\phi_k, \phi_{k'})]_{k,k'}$ the kernel matrix induced by a kernel function $k(\cdot, \cdot)$ and η the step size for the SVGD updates.

To this point, how to parametrize the prior P_ϕ and how to estimate the generalized marginal log-likelihood $\ln Z_{i,k} = \ln Z_\beta(S_i, P_{\phi_k})$ (`MLL_Estimator`) in Alg. 1 have remained unspecified. In the following two subsections, we discuss these components in more detail for GPs and BNNs.

[1]Note that any other approximate inference can be employed instead. We chose SVGD as we found it to work best in practice.

5.2. Meta-Learning Gaussian Process Priors

Setup. In GP regression, each data point corresponds to a tuple $z_{i,j} = (x_{i,j}, y_{i,j})$. For the i-th dataset, we write $S_i = (\mathbf{X}_i, \mathbf{y}_i)$, where $\mathbf{X}_i = (x_{i,1}, ..., x_{i,m_i})^\top$ and $\mathbf{y}_i = (y_{i,1}, ..., y_{i,m_i})^\top$. GPs are a Bayesian method in which the prior $P_\phi(h) = \mathcal{GP}(h | m_\phi(x), k_\phi(x, x'))$ is specified by a kernel $k_\phi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a mean function $m_\phi : \mathcal{X} \to \mathbb{R}$. Similar to Wilson et al. (2016) and Fortuin & Rätsch (2019), we instantiate m_ϕ and k_ϕ as neural networks, and aim to meta-learn the parameter vector ϕ. Moreover, we use $\lambda = n$, $\beta = m$, the negative log-likelihood as loss function and a Gaussian hyper-prior $\mathcal{P} = \mathcal{N}(0, \sigma_{\mathcal{P}}^2 I)$ over the GP prior parameters ϕ.

Algorithm. In our setup, $\ln Z_m(S_i, P_\phi) = \ln p(\mathbf{y}_i | \mathbf{X}_i, \phi)$ is the marginal log-likelihood of the GP which is available in closed form. In particular, the `MLL_Estimator` is given by (48) in Appx. B. Thus, the score $\nabla_\phi \ln \mathcal{Q}^*(\phi)$ is tractable, allowing us to perform SVGD efficiently. Alg. 2 in Appx. D summarizes the proposed meta-learning method which we refer to as *PACOH-GP*.

5.3. Meta-Learning Bayesian Neural Network Priors

Setup. Let $h_\theta : \mathcal{X} \to \mathcal{Y}$ be a function parametrized by a neural network (NN) with weights $\theta \in \Theta$. Using the NN mapping, we define a conditional distribution $p(y|x, \theta)$. For regression, we may set $p(y|x, \theta) = \mathcal{N}(y | h_\theta(x), \sigma^2)$, where σ^2 is the observation noise variance. We treat $\ln \sigma$ as a learnable parameter similar to the neural network weights θ so that a hypothesis coincides with a tuple $h = (\theta, \ln \sigma)$. For classification, we choose $p(y|x, \theta) = \text{Categorical}(\text{softmax}(h_\theta(x)))$. Our loss function is the negative log-likelihood $l(\theta, z) = -\ln p(y|x, \theta)$.

Next, we define a family of priors $\{P_\phi : \phi \in \Phi\}$ over the NN parameters θ. For computational convenience, we employ diagonal Gaussian priors, that is, $P_{\phi_l} = \mathcal{N}(\mu_{P_k}, \text{diag}(\sigma_{P_k}^2))$ with $\phi := (\mu_{P_k}, \ln \sigma_{P_k})$. Note that we represent σ_{P_k} in the log-space to avoid additional positivity constraints. In fact, any parametric distribution that supports re-parametrized sampling and has a tractable log-density (e.g., normalizing flows (c.f., Rezende & Mohamed, 2015)) could be used. Moreover, we use a zero-centered, spherical Gaussian hyper-prior $\mathcal{P} := \mathcal{N}(0, \sigma_{\mathcal{P}}^2 I)$ over the prior parameters ϕ.

Approximating the marginal log-likelihood. Unlike for GPs, the (generalized) marginal log-likelihood (MLL)

$$\ln Z_\beta(S_i, P_\phi) = \ln \mathbb{E}_{\theta \sim P_\phi} \left[e^{-\beta_i \hat{\mathcal{L}}(\theta, S_i)} \right] \tag{8}$$

is intractable for BNNs. Estimating and optimizing $\ln Z_\beta(S_i, P_\phi)$ is not only challenging due to the high-dimensional expectation over Θ but also due to numerical instabilities inherent in computing $e^{-\beta_i \hat{\mathcal{L}}(\theta, S_i)}$ when

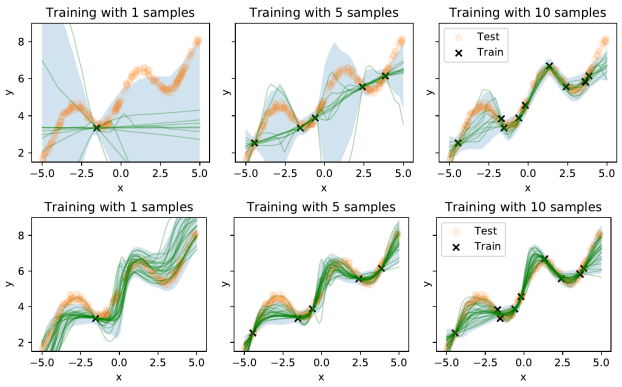

Figure 2. BNN posterior predictions with (top) standard Gaussian prior vs. (bottom) meta-learned prior. Meta-learning with *PACOH-NN* was conducted on the *Sinusoids* environment.

$\beta_i = m$ is large. Aiming to overcome these issues, we compute numerically stable Monte Carlo estimates of $\nabla_\phi \ln Z_\beta(S_i, P_{\phi_k})$ by combining the LogSumExp (LSE) with the re-parametrization trick (Kingma & Welling, 2014). In particular, the `MLL_Estimator` draws L samples $\theta_l := f(\phi_k, \epsilon_l) = \mu_{P_k} + \sigma_{P_k} \odot \epsilon_l,\ \epsilon_l \sim N(0, I)$ and computes the generalized MLL estimate as follows:

$$\ln \tilde{Z}_{\beta_i}(S_i, P_\phi) := \text{LSE}_{l=1}^{L} \left(-\beta_i \hat{\mathcal{L}}(\theta_l, S_i) \right) - \ln L \quad (9)$$

Note that $\ln \tilde{Z}_\beta(S_i, P_\phi)$ is a consistent but not an unbiased estimator of $\ln Z_\beta(S_i, P_\phi)$ (see Appx. D for details).

Algorithm. Alg. 3 in Appx. D summarizes the proposed meta-learning method which we henceforth refer to as *PACOH-NN*. To estimate the score $\nabla_{\phi_{k'}} \ln \mathcal{Q}^*(\phi_{k'})$, we can even use mini-batching on the task level. This mini-batched version, outlined in Alg. 4, maintains K particles to approximate the hyper-posterior, and in each forward step samples L NN-parameters (of dimensionaly $|\Theta|$) per prior particle, that are deployed on a mini-batch of n_{bs} tasks, to estimate the score of $\mathcal{Q}^*$. As a result, the total space complexity is on the order of $\mathcal{O}(|\Theta|K+L)$ and the computational complexity of the algorithm for a single iteration is $\mathcal{O}(K^2 + KLn_{bs})$.

A key advantage of *PACOH-NN* over previous methods for meta-learning BNN priors (e.g., Pentina & Lampert, 2014; Amit & Meir, 2018) is that it *turns the previously nested optimization problem into a much simpler standard stochastic optimization problem*. This makes meta-learning not only much more stable but also more scalable. In particular, we do not need to explicitly compute the task posteriors Q_i and can employ mini-batching over tasks. Thus, the computational and space complexities *do not* depend on the number of tasks n. In comparison, *MLAP* (Amit & Meir, 2018) has a memory footprint of $\mathcal{O}(|\Theta|n)$ making meta-learning prohibitive even for moderately many (e.g., 50) tasks.

6. Experiments

We now empirically evaluate the two methods *PACOH-GP*[2] and *PACOH-NN*[3] that were introduced in Section 5. Comparing them to existing meta-learning approaches on various regression and classification environments, we demonstrate that our *PACOH*-based methods (i) outperform previous meta-learning algorithms in *predictive accuracy*, (ii) improve the calibration of *uncertainty estimates*, (iii) are much more *scalable* than previous PAC-Bayesian meta-learners, and (iv) effectively combat *meta-overfitting*. Finally, we showcase how meta-learned *PACOH-NN* priors can be harnessed in a real-world *sequential decision making* task concerning peptide-based vaccine development.

6.1. Experiment Setup

Regression environments. In our experiments, we consider two synthetic and four real-world meta-learning environments for *regression*. As a synthetic environment we employ *Sinusoids* of varying amplitude, phase and slope as well as a 2-dimensional mixture of *Cauchy* distributions plus random GP functions. As real-world environments, we use datasets corresponding to different calibration sessions of the Swiss Free Electron Laser (*SwissFEL*) (Milne et al., 2017; Kirschner et al., 2019b), as well as data from the *PhysioNet* 2012 challenge, which consists of time series of electronic health measurements from intensive care patients (Silva et al., 2012), in particular the Glasgow Coma Scale (*GCS*) and the hematocrit value (*HCT*). Here, the different tasks correspond to different patients. Moreover, we employ the Intel Berkeley Research Lab temperature sensor dataset (*Berkeley-Sensor*) (Madden, 2004) where the tasks require auto-regressive prediction of temperature measurements corresponding to sensors installed in different locations of the building. Further details can be found in Appendix E.1.

Classification environments. We conduct experiments with the multi-task *classification* environment *Omniglot* (Lake et al., 2015), consisting of handwritten letters across 50 alphabets. Unlike previous work (e.g., Finn et al., 2017) we do not perform data augmentation and do not recombine letters of different alphabets, thus preserving the data's original structure and mitigating the need for prior knowledge. In particular, one task corresponds to 5-way 5-shot classification of letters within an alphabet. This leaves us with much fewer tasks (30 meta-train, 20 meta-test tasks), making the environment more challenging and interesting for uncertainty quantification. This also allows us to include *MLAP* in the experiment which hardly scales to more than 50 tasks.

[2]The source code for *PACOH-GP* is available at `tinyurl.com/pacoh-gp-code`.

[3]An implementation of *PACOH-NN* can be found at `tinyurl.com/pacoh-nn-code`.

	Cauchy	SwissFel	Physionet-GCS	Physionet-HCT	Berkeley-Sensor
Vanilla GP	0.275 ± 0.000	0.876 ± 0.000	2.240 ± 0.000	2.768 ± 0.000	0.276 ± 0.000
Vanilla BNN (Liu & Wang, 2016)	0.327 ± 0.008	0.529 ± 0.022	2.664 ± 0.274	3.938 ± 0.869	0.109 ± 0.004
MLL-GP (Fortuin & Rätsch, 2019)	0.216 ± 0.003	0.974 ± 0.093	1.654 ± 0.094	2.634 ± 0.144	0.058 ± 0.002
MLAP (Amit & Meir, 2018)	0.219 ± 0.004	0.486 ± 0.026	2.009 ± 0.248	2.470 ± 0.039	0.050 ± 0.005
MAML (Finn et al., 2017)	0.219 ± 0.004	0.730 ± 0.057	1.895 ± 0.141	2.413 ± 0.113	$\mathbf{0.045 \pm 0.003}$
BMAML (Kim et al., 2018)	0.225 ± 0.004	0.577 ± 0.044	1.894 ± 0.062	2.500 ± 0.002	0.073 ± 0.014
NP (Garnelo et al., 2018)	0.224 ± 0.008	0.471 ± 0.053	2.056 ± 0.209	2.594 ± 0.107	0.079 ± 0.007
PACOH-GP (ours)	0.209 ± 0.008	0.376 ± 0.024	$\mathbf{1.498 \pm 0.081}$	$\mathbf{2.361 \pm 0.047}$	0.065 ± 0.005
PACOH-NN (ours)	$\mathbf{0.195 \pm 0.001}$	$\mathbf{0.372 \pm 0.002}$	$\mathbf{1.561 \pm 0.061}$	$\mathbf{2.405 \pm 0.017}$	$\mathbf{0.043 \pm 0.001}$

Table 1. Comparison of standard and meta-learning algorithms in terms of test RMSE in 5 meta-learning environments for regression. Reported are mean and standard deviation across 5 seeds. Our proposed method *PACOH* achieves the best performance across all tasks.

	Cauchy	SwissFel	Physionet-GCS	Physionet-HCT	Berkeley-Sensor
Vanilla GP	0.087 ± 0.000	0.135 ± 0.000	0.268 ± 0.000	$\mathbf{0.277 \pm 0.000}$	0.119 ± 0.000
Vanilla BNN (Liu & Wang, 2016)	0.055 ± 0.006	0.085 ± 0.008	0.277 ± 0.013	0.307 ± 0.009	0.179 ± 0.002
MLL-GP (Fortuin & Rätsch, 2019)	0.059 ± 0.003	0.096 ± 0.009	0.277 ± 0.009	0.305 ± 0.014	0.153 ± 0.007
MLAP (Amit & Meir, 2018)	0.086 ± 0.015	0.090 ± 0.021	0.343 ± 0.017	0.344 ± 0.016	0.108 ± 0.024
BMAML (Kim et al., 2018)	0.061 ± 0.007	0.115 ± 0.036	0.279 ± 0.010	0.423 ± 0.106	0.161 ± 0.013
NP (Garnelo et al., 2018)	0.057 ± 0.009	0.131 ± 0.056	0.299 ± 0.012	0.319 ± 0.004	0.210 ± 0.007
PACOH-GP (ours)	0.056 ± 0.004	0.038 ± 0.006	$\mathbf{0.262 \pm 0.004}$	0.296 ± 0.003	0.098 ± 0.005
PACOH-NN (ours)	$\mathbf{0.046 \pm 0.001}$	$\mathbf{0.027 \pm 0.003}$	0.267 ± 0.005	0.302 ± 0.003	$\mathbf{0.067 \pm 0.005}$

Table 2. Comparison of standard and meta-learning methods in terms of the test calibration error in 5 regression environments. We report the mean and standard deviation across 5 random seeds. PACOH yields the best uncertainty calibration in the majority of environments.

Baselines. We use a *Vanilla GP* with squared exponential kernel and a *Vanilla BNN* with a zero-centered, spherical Gaussian prior and SVGD posterior inference (Liu & Wang, 2016) as baselines. Moreover, we compare our proposed approach against various popular meta-learning algorithms, including model-agnostic meta-learning (*MAML*) (Finn et al., 2017), Bayesian MAML (*BMAML*) (Kim et al., 2018) and the PAC-Bayesian approach by Amit & Meir (2018) (*MLAP*). For regression experiments, we also report results for neural processes (*NPs*) (Garnelo et al., 2018) and a GP with neural-network-based mean and kernel function, meta-learned by maximizing the marginal log-likelihood (*MLL-GP*) (Fortuin & Rätsch, 2019). Among all, *MLAP* is the most similar to our approach as it is neural-network-based and minimizes PAC-Bayesian bounds on the transfer error. Though, unlike *PACOH-NN*, it relies on nested optimization of the task posteriors Q_i and the hyper-posterior $\mathcal{Q}$. *MLL-GP* is similar to *PACOH-GP* insofar that it also maximizes the sum of marginal log-likelihoods $\ln Z_m(S_i, P_\phi)$ across tasks. However, unlike *PACOH-GP* it lacks any form of meta-level regularization.

6.2. Experiment Results

Qualitative example. Figure 2 illustrates *BNN* predictions on a sinusoidal regression task with a standard Gaussian prior as well as a *PACOH-NN* prior meta-learned with 20 tasks from the *Sinusoids* environment. We can see that the standard Gaussian prior provides poor inductive bias, not only leading to bad mean predictions away from the test points but also to poor 95% confidence intervals (blue shaded areas). In contrast, the meta-learned *PACOH-NN* prior encodes useful inductive bias towards sinusoidal function shapes, leading to better predictions and uncertainty estimates, even with minimal training data.

	Accuracy	Calibration error
Vanilla BNN (Liu & Wang, 2016)	0.795 ± 0.006	0.135 ± 0.009
MLAP (Amit & Meir, 2018)	0.700 ± 0.0135	0.108 ± 0.010
MAML (Finn et al., 2017)	0.693 ± 0.013	0.109 ± 0.011
BMAML (Kim et al., 2018)	0.764 ± 0.025	0.191 ± 0.018
PACOH-NN (ours)	$\mathbf{0.885 \pm 0.090}$	$\mathbf{0.091 \pm 0.010}$

Table 3. Comparison of meta-learning algorithms in terms of test accuracy and calibration error on the *Omniglot* environment. Among the methods, PACOH-NN makes the most accurate and best-calibrated class predictions.

PACOH improves the predictive accuracy. Using the meta-learning environments and baseline methods that we introduced in Sec. 6.1, we perform a comprehensive benchmark study. Table 1 reports the results on the regression environments in terms of the root mean squared error (RMSE) on unseen test tasks. Among the approaches, *PACOH-NN* and *PACOH-GP consistently perform best or are among the best methods*. Similarly, *PACOH-NN* achieves the *highest accuracy* in the Omniglot classification environment (cf. Table 3). Overall, this demonstrates that the introduced meta-learning framework is not only sound, but also yields state-of-the-art empirical performance in practice.

PACOH improves the predictive uncertainty. We hypothesize that by acquiring the prior in a principled data-driven manner (e.g., with *PACOH*), we can improve the quality of the GP's and BNN's uncertainty estimates. To investigate the effect of meta-learned priors on the uncertainty estimates of the base learners, we compute the probabilistic predictors' calibration errors, reported in Table 2 and 3.

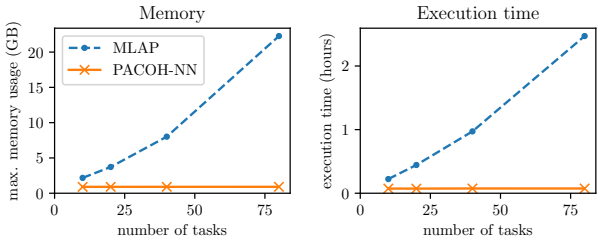

Figure 3. Comparison of *PACOH-NN* and *MLAP* in memory footprint and compute time, as the number of meta-training task grows. *PACOH-NN* scales much better in the number of tasks than *MLAP*.

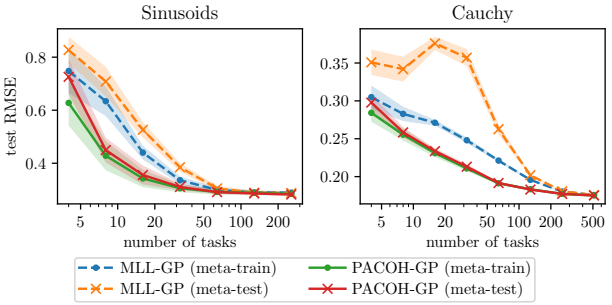

Figure 4. Test RMSE on meta-training and meta-testing tasks as a function of the number of meta-training tasks for PACOH-GP and MLL-GP. The performance gap between the train and test tasks demonstrates overfitting in the MLL method, while PACOH performs consistently better and barely overfits.

The *calibration error* measures the discrepancy between the predicted confidence regions and the actual frequencies of test data in the respective areas (Kuleshov et al., 2018). Note that, since *MAML* only produces point predictions, the concept of calibration does not apply to it. We observe that meta-learning priors with *PACOH-NN* consistently improves the Vanilla BNN's uncertainty estimates. Similarly, *PACOH-GP* yields a lower calibration error than the Vanilla GP in the majority of the envionments. For meta-learning environments where the task similarity is high, like *Swiss-FEL* and *Berkeley-Sensor*, the improvement is substantial.

PACOH is scalable. Unlike *MLAP* (Amit & Meir, 2018), *PACOH-NN* does not need to maintain posteriors Q_i for the meta-training tasks and can use mini-batching on the task level. As a result, it is *computationally much faster and more scalable* than previous PAC-Bayesian meta-learners. This is reflected in its computation and memory complexity, discussed in Section 5. Figure 3 showcases this computational advantage during meta-training with *PACOH-NN* and *MLAP* on the *Sinusoids* environment with varying number of tasks, reporting the maximum memory requirements, as well as the training time. While *MLAP's* memory consumption and compute time grow linearly and become prohibitively large even for less than 100 tasks, *PACOH-NN* maintains a constant memory and compute load as the number of tasks grow.

PACOH combats meta-overfitting. As Qin et al. (2018) and Yin et al. (2020) point out, many popular meta-learners (e.g., Finn et al., 2017; Garnelo et al., 2018) require a large number of meta-training tasks to generalize well. When presented with only a limited number of tasks, such algorithms suffer from severe meta-overfitting, adversely impacting their performance on unseen tasks from $\mathcal{T}$. This can even lead to *negative transfer*, such that meta-learning actually hurts the performance when compared to standard learning. In our experiments, we also observe such failure cases: For instance, in the classification environment (cf. Table 3), *MAML* fails to improve upon the Vanilla BNN. Similarly, in the regression environments (cf. Table 3) we find that *NPs*, *BMAML* and *MLL-GP* often yield worse-calibrated predictive distributions than the Vanilla BNN and GP respectively. In contrast, thanks to its theoretically principled construction, *PACOH-NN* is able to achieve positive transfer even when the tasks are diverse and small in number. In particular, the hyper-prior acts as meta-level regularizer by penalizing complex priors that are unlikely to convey useful inductive bias for unseen learning tasks.

To investigate the importance of meta-level regularization through the hyper-prior in more detail, we compare the performance of our proposed method *PACOH-GP* to *MLL-GP* (Fortuin & Rätsch, 2019) which also maximizes the sum of GP marginal log-likelihoods across tasks but has no hyper-prior nor meta-level regularization. Fig. 4 shows that MLL-GP performs significantly better on the meta-training tasks than on the meta-test tasks in both of our synthetic regression environments. This gap between meta-train performance and meta-test performance signifies overfitting on the meta-level. In contrast, our method hardly exhibits this gap and consistently outperforms MLL-GP. As expected, this effect is particularly pronounced when the number of meta-training tasks is small (i.e., less than 100) and vanishes as n becomes large. Similar results for other meta-learning methods can be found in Appendix E.5. Once more, this demonstrates the importance of meta-level regularization, and shows that our proposed framework effectively addresses the problem of meta-overfitting.

6.3. Meta-Learning for Sequential Decision Making

Finally, we showcase how a relevant real-world application such as *vaccine design* can benefit from our proposed method. The goal is to discover peptides which bind to major histocompatibility complex class-I molecules (MHC-I). Following the Bayesian optimization (BO) setup of Krause & Ong (2011), each task corresponds to searching for maximally binding peptides, a vital step in the design of peptide-based vaccines. The tasks differ in their targeted MHC-I allele, corresponding to different genetic variants of the MHC-I protein. We use data from Widmer et al. (2010), which contains the standardized binding affinities

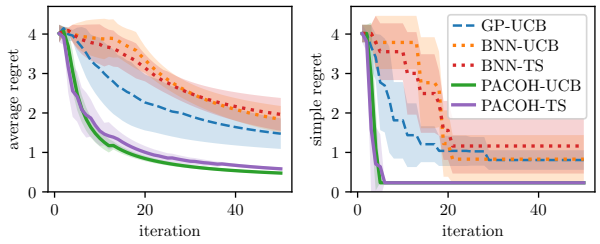

Figure 5. MHC-I peptide design task: Regret for different priors and bandit algorithms. A meta-learned *PACOH-NN* prior substantially improves the regret, compared to a standard BNN/GP prior.

(IC_{50} values) of different peptide candidates (encoded as 45-dimensional feature vectors) to the MHC-I alleles.

We use 5 alleles (tasks) to meta-learn a BNN prior with *PACOH-NN* and leave the most genetically dissimilar allele (A-6901) for our bandit task. In each iteration, the experimenter (i.e., the Bayesian optimization algorithm) chooses to test one peptide among the pool of more than 800 candidates and receives its binding affinity as a reward feedback. In particular, we employ UCB (Auer, 2002) and Thompson-Sampling (TS) (Thompson, 1933) as bandit algorithms, comparing the BNN-based bandits with meta-learned prior (*PACOH-UCB/TS*) against a zero-centered Gaussian BNN prior (*BNN-UCB/TS*) and a Gaussian process (*GP-UCB*) (Srinivas et al., 2009).

Fig. 5 reports the respective average regret and simple regret over 50 iterations. Unlike the bandit algorithms with standard BNN/GP prior, *PACOH-UCB/TS* reaches near optimal regret within less than 10 iterations and after 50 iterations still maintains a significant performance advantage. This highlights the importance of *transfer (learning)* for solving real-world problems and demonstrates the effectiveness of *PACOH-NN* to this end. While the majority of meta-learning methods rely on a large number of meta-training tasks (Qin et al., 2018), *PACOH-NN* allows us to achieve promising positive transfer, even in complex real-world scenarios with only a handful (in this case 5) of tasks.

7. Conclusion

We presented PACOH, a novel, theoretically principled, and scalable PAC-Bayesian meta-learning approach. PACOH outperforms existing methods in terms of predictive performance and uncertainty calibration, while providing PAC-Bayesian guarantees without relying on costly nested optimization. It can be used with different base learners (e.g., GPs or BNNs) and achieves positive transfer in regression and classification with as little as five meta-tasks. BNNs meta-learnt with PACOH can be effectively used for sequential decision making, as demonstrated by our vaccine design application. We believe our approach provides an important step towards learning useful inductive bias from data in a flexible, scalable, and principled manner.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program grant agreement No 815943 and was supported by Oracle Cloud Services. Vincent Fortuin is funded by a PhD fellowship from the Swiss Data Science Center and by the grant 2017-110 of the Strategic Focus Area "Personalized Health and Related Technologies (PHRT)" of the ETH Domain. We thank David Lindner, Gideon Dresdner, and Claire Vernade for their valuable feedback.

References

Alquier, P., Ridgway, J., Chopin, N., and Teh, Y. W. On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research*, 2016.

Amit, R. and Meir, R. Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *International Conference on Machine Learning*, 2018.

Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, 2016.

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Baxter, J. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 2000.

Bergstra, J., Yamins, D., and Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International Conference on Machine Learning*, 2013.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational Inference: A Review for Statisticians. *arXiv preprint arXiv:1601.00670*, 2016.

Boucheron, S., Lugosi, G., and Massart, P. Concentration inequalities : a nonasymptotic theory of independence. chapter 2.4, pp. 27–30. Oxford University Press, 2013.

Catoni, O. PAC-Bayesian supervised classification: the thermodynamics of statistical learning. *arXiv*, 2007.

Chen, Y., Hoffman, M. W., Colmenarejo, S. G., Denil, M., Lillicrap, T. P., Botvinick, M., and De Freitas, N. Learning to Learn without Gradient Descent by Gradient Descent. In *International Conference on Machine Learning*, 2017.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Uncertainty in Artificial Intelligence*, 2016.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, 2018.

Fortuin, V. and Rätsch, G. Deep Mean Functions for Meta-Learning in Gaussian Processes. *arXiv preprint: arXiv:1901.08098*, 2019.

Fortuin, V., Rätsch, G., and Mandt, S. Multivariate time series imputation with variational autoencoders. *arXiv preprint arXiv:1907.04155*, 2019.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S., and Teh, Y. W. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Germain, P., Bach, F., Lacoste, A., and Lacoste-Julien, S. Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, 2016.

Guedj, B. A primer on PAC-Bayesian learning. In *2nd Congress of the French Mathematical Society*, 2019.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.

Hochreiter, S., Younger, A. S., and Conwell, P. R. Learning To Learn Using Gradient Descent. In *International Conference on Artificial Neural Networks*, 2001.

Kim, T., Yoon, J., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kirschner, J., Mutný, M., Hiller, N., Ischebeck, R., and Krause, A. Adaptive and Safe Bayesian Optimization in High Dimensions via One-Dimensional Subspaces. In *International Conference on Machine Learning*, 2019a.

Kirschner, J., Nonnenmacher, M., Mutný, M., Hiller, N., Adelmann, A., Ischebeck, R., and Krause, A. Bayesian Optimization for Fast and Safe Parameter Tuning of SwissFEL. In *International Free-Electron Laser Conference (FEL2019)*, 2019b.

Krause, A. and Ong, C. S. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, 2011.

Kuleshov, V., Fenner, N., and Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2018.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 2015.

Lever, G., Laviolette, F., and Shawe-Taylor, J. Tighter PAC-Bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 473:4–28, feb 2013. ISSN 0304-3975. doi: 10.1016/J.TCS.2012.10.013.

Liu, Q. and Wang, D. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *Advances in Neural Information Processing Systems*, 2016.

Madden, S. Intel lab data. `http://db.csail.mit.edu/labdata/labdata.html`, 2004. Accessed: Sep 8, 2020.

McAllester, D. A. Some PAC-Bayesian theorems. *Machine Learning*, 1999.

Milne, C. J., Schietinger, T., Aiba, M., Alarcon, A., Alex, J., Anghel, A., Arsov, V., Beard, C., Beaud, P., Bettoni, S., et al. Swissfel: the swiss x-ray free electron laser. *Applied Sciences*, 2017.

Nichol, A., Achiam, J., and Schulman, J. On First-Order Meta-Learning Algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Parrado-Hernandez, E., Ambroladze, A., Shawe-Taylor, J., and Sun, S. PAC-Bayes Bounds with Data Dependent Priors. *Journal of Machine Learning Research*, 2012.

Pentina, A. and Lampert, C. A PAC-Bayesian bound for lifelong learning. In *International Conference on Machine Learning*, 2014.

Qin, Y., Zhang, W., Zhao, C., Wang, Z., Shi, H., Qi, G., Shi, J., and Lei, Z. Rethink and redesign meta learning. *arXiv preprint arXiv:1812.04955*, 2018.

Rasmussen, C. E. and Ghahramani, Z. Occam's Razor. In *NIPS*, volume 13, pp. 294—-300, 2001.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes in machine learning*. 2006.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *International Conference on Machine Learning*, 2015.

Rothfuss, J., Ferreira, F., Boehm, S., Walther, S., Ulrich, M., Asfour, T., and Krause, A. Noise Regularization for Conditional Density Estimation. *arXiv preprint arXiv:1907.08982*, 2019a.

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., and Abbeel, P. ProMP: Proximal Meta-Policy Search. In *International Conference on Learning Representations*, 2019b.

Silva, I., Moody, G., Scott, D. J., Celi, L. A., and Mark, R. G. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *Computing in Cardiology*, 2012.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2009.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

Thrun, S. and Pratt, L. (eds.). *Learning to Learn*. Kluwer Academic Publishers, 1998.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

Widmer, C., Toussaint, N. C., Altun, Y., and Rätsch, G. Inferring latent task structure for multitask learning by multiple kernel learning. *BMC bioinformatics*, 2010.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *Artificial Intelligence and Statistics*, pp. 370–378, 2016.

Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. Meta-learning without memorization. In *International Conference on Learning Representations*, 2020.