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

Anonymous Authors[1]

Abstract

Meta-learning can successfully acquire useful inductive biases from data, especially when a large number of meta-tasks are available. Yet, its generalization properties to unseen tasks are poorly understood. Particularly if the number of meta-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 with unbounded loss functions and Bayesian base learners. Using these bounds, we develop a class of PAC-optimal meta-learning algorithms with performance guarantees and a principled *meta-regularization*. When instantiating our *PAC-optimal hyper-posterior (PACOH)* with Gaussian processes as base learners, the resulting approach consistently outperforms several popular meta-learning methods, both in terms of predictive accuracy and the quality of its uncertainty estimates.

1. Introduction

Meta-learning aims to extract prior knowledge from data, accelerating the learning process in light of new learning tasks (Schmidhuber, 1987; Thrun and 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, c.f., Qin et al., 2018), thus impairing the performance on yet unseen target tasks. Hence, a key question is how to regularize the meta-learner in order to ensure that it generalizes to unseen tasks.

Especially when dealing with small amounts of data, reasoning about the *uncertainty* of a prediction model is crucial (Gal, 2016; Rothfuss et al., 2019a). In this setting, Bayesian methods are a popular choice. Moreover, they lend themselves easily to the meta-learning setting, since they offer a principled way to include prior knowledge into the learner (McNeish, 2016). However, the associated uncertainty estimates are often not well-calibrated due to model misspecification and poorly chosen priors (Kuleshov et al., 2018).

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 and Lampert, 2014; Amit and Meir, 2018) only consider *bounded* loss functions which makes them hardly applicable to regression or probabilistic inference. Moreover, they pose a 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*, which also *does not rely on nested optimization*. For Bayesian learners, we further tighten our PAC-Bayesian bounds, relating them directly to the marginal log-likelihood of the Bayesian model. This allows us to derive the *PAC-optimal hyper-posterior (PACOH)*, which promises strong performance guarantees and a principled meta-level regularization. Most importantly, it can be approximated using standard variational methods (Blei et al., 2016), giving rise to an entire range of meta-learning algorithms.

We instantiate our framework with *Gaussian processes* (GPs) as base learners and empirically evaluate the resulting approach across several synthetic and real-world regression environments. We show that the meta-level regularization imposed by PACOH effectively prevents meta-overfitting. This allows us to obtain substantial improvements over standard GPs, even when only five tasks are available for meta-training. In our empirical study, PACOH consistently outperforms several popular meta-learning approaches, yielding more accurate predictions as well as improved uncertainty estimates. Overall, PACOH not only provides optimal PAC-guarantees, but also constitutes an easily applicable meta-learning framework with strong empirical performance. Especially in settings where data is scarce and good uncertainty estimates are crucial, PACOH offers a promising alternative to hand-designed priors.

In summary, our main contributions are the following:

- We present the first PAC-Bayesian generalization bound for meta-learning with unbounded loss, which does not rely on nested optimization.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

- For Bayesian learners, we provide a tighter meta-learning bound, which relates to the model's marginal log-likelihood.
- Based on this bound, we derive a PAC-optimal hyper-posterior (PACOH) which can be tractably approximated with standard variational techniques.
- We provide strong empirical evidence that our proposed meta-learning method outperforms popular meta-learning approaches on a range of environments.

2. Related work

Meta-learning. One category of meta-learners directly attempts to learn the "learning program" as a recurrent model (Hochreiter et al., 2001; Andrychowicz et al., 2016; Santoro et al., 2016; Chen et al., 2017). Another popular approach is to meta-learn the initialization of a neural network such that it can be adapted to new tasks in a few gradient-steps (Finn et al., 2017; Nichol et al., 2018; Rothfuss et al., 2019b) or to amortize the inference of a stochastic process (Garnelo et al., 2018). Although they are able to learn complex inference patterns, these methods rely on a setting where meta-tasks are abundant and do not provide any performance guarantees. The associated risk of meta-overfitting has previously been noted (Qin et al., 2018; Fortuin and Rätsch, 2019), but still lacks a rigorous formal analysis under realistic assumptions (e.g., with unbounded loss functions). We provide this principled treatment by studying the generalization properties of these algorithms within the PAC-Bayesian framework.

PAC-Bayesian theory. Previous work presents generalization bounds for randomized predictors under various assumptions (McAllester, 1999; Catoni, 2007; Alquier et al., 2016). Such generalization guarantees have been extended to meta-learning (Pentina and Lampert, 2014; Amit and Meir, 2018), but only for bounded loss functions. Moreover, these meta-generalization bounds are hard to minimize as they leave both the hyper-posterior and posterior unspecified, which leads to nested optimization problems. In contrast, our bounds also hold for unbounded losses and give rise to a tractable meta-learning objective. In our derivations, we build on connections between PAC-Bayesian and classical Bayesian inference, as described by Germain et al. (2016).

Hierarchical Bayes and Hyper-parameter Learning. Our meta-learning setup, based on Baxter (2000), shares many aspects with hierarchical modeling and hyper-parameter learning (MacKay, 1991). However, work in this area either focuses on learning priors for a single task (Ong et al., 2005; Wilson et al., 2016; Reeb et al., 2018), makes stronger assumptions on how tasks are statistically related (Yu et al., 2005; Bonilla et al., 2008), or lacks guarantees (e.g., Grant et al., 2018; Yoon et al., 2018).

3. Background: PAC-Bayesian Framework

3.1. Preliminaries and notation

A learning task is characterized by an unknown data distribution $\mathcal{D}$, defined 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 are typically concerned with 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 on unseen input features $x^* \sim \mathcal{D}_x$. The quality of the predictions are 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 are concerned with *randomized predictors*, formally defined as 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 probability measures over the set $\mathcal{H}$. Note that in Bayesian inference, the prior and posterior are assumed to be tightly connected through Bayes' theorem. In contrast, the PAC-Bayesian framework makes fewer assumptions and only requires the prior to be independent of the observed data, while the posterior may depend on it. For a detailed treatment of the PAC-Bayesian methodology, we refer to Guedj (2019). In the following, we overload the notation by also denoting the respective probability densities as Q and P. Moreover, 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 $\hat{\mathcal{L}}(Q, S) = \mathbb{E}_{h \sim Q}\ \hat{\mathcal{L}}(h, S)$.

3.2. PAC-Bayesian bounds

In practice, the generalization error $\mathcal{L}(Q, \mathcal{D})$ is unknown. Thus, one typically resorts to empirical risk minimization (ERM), that is, optimizing $\hat{\mathcal{L}}(Q, S)$ as a proxy for the true objective. However, this may result in overfitting and poor generalization performance. Naturally, the question arises whether we can bound the unknown generalization error based on its empirical estimate. The PAC-Bayesian framework provides such a guarantee with high probability:

Theorem 1. *(Alquier et al., 2016) Given a data distribution $\mathcal{D}$, a hypothesis space $\mathcal{H}$, a loss function $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$, a prior distribution $P \in \mathcal{M}(\mathcal{F})$, a 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 $Q \in \mathcal{M}(\mathcal{H})$:*

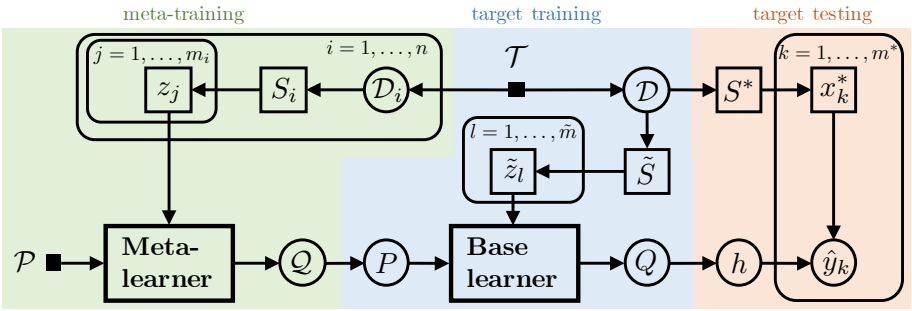


Figure 1: Overview of the described meta-learning framework with environment $\mathcal{T}$, meta-task distributions $\mathcal{D}_i$, target task distribution $\mathcal{D}$, hyper-prior $\mathcal{P}$, hyper-posterior $\mathcal{Q}$, target prior P, target posterior Q, data sets S, and data points $z = (x, y)$.

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

where

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

Note that the prior distribution P must be independent of the data S. Since $\Psi(\beta, m)$ contains $\mathcal{L}(h, \mathcal{D})$ which is unknown in practice, Theorem 1 cannot be directly applied. However, if we make additional assumptions about the loss function l, we can bound $\Psi(\beta, m)$ and thereby obtain useful PAC-Bayesian bounds. In the following, we briefly discuss three such assumptions and the resulting bounds. For detailed explanations and derivations we refer to Appendix A.1.

Bounded loss function. When $l : \mathcal{H} \times \mathcal{Z} \to [a, b]$ is bounded, we can use Hoeffding's lemma to obtain

$$\Psi(\beta, m) \leq \frac{\beta^2(b-a)^2}{8m} \,. \tag{2}$$

Sub-gamma loss. 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 bound $\Psi(\beta, m)$ as follows

$$\Psi(\beta, m) \leq \frac{\beta^2 s^2}{2m(1 - \frac{c\beta}{m})} \,. \tag{3}$$

Sub-gaussian loss. A *sub-gaussian* loss with variance s^2 can be considered the limit case $c \to 0^+$ of the sub-gamma assumption. In this case, we obtain $\Psi(\beta, m) \leq \frac{\beta^2 s^2}{2m}$.

3.3. 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). The following lemma gives us the closed form solution to such a minimization problem over $\mathcal{M}(\mathcal{H})$.

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 $h \in \mathcal{H}$,*

$$Q^*(h) := \frac{P(h)e^{-\beta g(h)}}{Z} = \frac{P(h)e^{-\beta g(h)}}{\mathbb{E}_{h\sim P}\left[e^{-\beta g(h)}\right]} \tag{4}$$

is the minimizing probability density of

$$\underset{Q\in\mathcal{M}(\mathcal{H})}{\arg\min}\ \beta\mathbb{E}_{h\sim Q}\left[g(h)\right] + D_{KL}(Q||P) \,. \tag{5}$$

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, and $\beta = m$ we can write the minimizer of (1) as

$$Q^*(h) = \arg\min_{Q} m\hat{\mathcal{L}}(Q, S) + D_{KL}(Q||P) \tag{6}$$

$$= \frac{1}{Z(S, P)} P(h)e^{-m\hat{\mathcal{L}}(h,S)} \tag{7}$$

where $Z(S, P)$ is a normalization constant. In a probabilistic setting, we would define the loss function $l(\cdot)$ as the negative log-likelihood of the data, that is, $l(h, z_i) := -\log p(z_i|h)$. Thus, the optimal Gibbs posterior coincides with the *Bayesian posterior*

$$Q^*(h; P, S) = \frac{P(h)\, p(S \,|\, h)}{Z(S, P)} \tag{8}$$

where $Z(S, P) = \int_{\mathcal{H}} P(h) \prod_{j=1}^{m} p(z_j|h)\, dh$ is called the *marginal likelihood* of the sample S.

4. PAC-Bayesian bounds for Meta-Learning

We now present our main theoretical contributions. An overview of our proposed framework is depicted in Figure 1.

4.1. Meta-Learning

In the standard supervised learning setup (see Section 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, observed in the form of a dataset S, to update the prior into a posterior distribution 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. Note that the number of samples m may vary between datasets.

While the prior P is arbitrary but fixed with respect to S, it can have a major impact on the posterior chosen by the learning procedure $Q(S, P)$ and the KL-divergence in the PAC-Bayesian bound in (1). Thus, the question arises how to properly choose P. Importantly, for the PAC-Bayes bounds to hold, the choice of P cannot be based on our data set S. However, we may consult data with *similar properties* to improve our choice of P. This is the central idea of *meta-learning* (Vilalta and Drissi, 2002), which aims to learn inductive bias (e.g., in form of a prior) based on a set of statistically related tasks. In the remainder of this section, we follow the setup of Baxter (2000) and Pentina and Lampert (2014).

So far, we have only considered a single learning task with distribution $\mathcal{D}$. To extend this to the meta-learning setting, we consider different tasks $\tau_i = (\mathcal{D}_i, m_i)$. All tasks share the same data domain $\mathcal{Z}$, hypothesis space $\mathcal{H}$ and loss function $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$, but differ in their data distributions $\mathcal{D}_i$ and the number of samples m_i drawn from it. The meta-learner observes n training sets $S_1, ..., S_n$ corresponding to n different tasks. Each dataset $S_i \sim \mathcal{D}_i^{m_i}$ is assumed to be sampled i.i.d. from its respective task distribution $\mathcal{D}_i$. We further assume that each task $\tau_i \sim \mathcal{T}$ is drawn i.i.d. from an *environment* $\mathcal{T}$, which is a probability distribution over data distributions and sample sizes. The goal is to extract knowledge from the observed tasks which can then be used as prior knowledge for learning on new (yet unobserved) target tasks $\tau \sim \mathcal{T}$ (Amit and Meir, 2018). The prior knowledge is represented as a prior distribution $P \in \mathcal{M}(\mathcal{H})$ over learning hypotheses h which the base learner $Q(S, P)$ utilizes for inference on new tasks.

A meta-learner acquires such a prior distribution P in a data-driven way. In order to be able to extend the PAC-Bayesian analysis to the meta-learning setting, we again consider the notion of probability distributions on function spaces. The object of interest, which has previously been $h \in \mathcal{H}$, is now the prior distribution $P \in \mathcal{M}(\mathcal{H})$.

In the meta-learning PAC-Bayes framework, the meta-learner presumes a *hyper-prior* $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$, that is, a distribution over priors P. Observing data sets $S_1, ..., S_n$ from multiple tasks, the meta-learner then updates the hyper-prior to a *hyper-posterior* $\mathcal{Q}$. The performance of this hyper-posterior is measured as 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 $\mathcal{L}(\mathcal{Q}, \mathcal{T})$ is unknown in practice, we can estimate it using the *empirical multi-task error*

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

4.2. PAC-Bayesian Meta-Learning bounds

In the following, we contribute a novel bound on the transfer-error of a meta-learner. The proofs and derivations can be found in Appendix A.3.

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 and $\tilde{m} = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\right)^{-1}$ the harmonic mean of dataset sizes. For any confidence level $\delta \in (0, 1]$ the inequality*

$$\begin{aligned}\mathcal{L}(\mathcal{Q}, \mathcal{T}) \leq \hat{\mathcal{L}}(\mathcal{Q}, S_1, ..., S_n) &+ \left(\frac{1}{n} + \frac{1}{\tilde{m}}\right) D_{KL}(\mathcal{Q}||\mathcal{P}) \\ &+ \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q(S_i, P)||P)\right] \\ &+ C(\delta, n, \tilde{m})\end{aligned} \tag{9}$$

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

$$C(\delta, n, \tilde{m}) = \frac{1}{\tilde{m}} \ln\frac{2n}{\delta} + \frac{1}{n}\ln\frac{2}{\delta} + \frac{(b-a)^2}{4} . \tag{10}$$

If the loss function is sub-gamma with variance factor s^2 and scale parameter c, under the two-level prior $(\mathcal{P}, P)$ and the data distribution $(\mathcal{T}, \mathcal{D})$, the inequality holds for

$$C(\delta, n, \tilde{m}) = \frac{1}{\tilde{m}} \ln\frac{2n}{\delta} + \frac{1}{n}\ln\frac{2}{\delta} + \frac{s^2}{1-c} . \tag{11}$$

Proof. See Appendix A.3. □

The theorem uniformly bounds the true transfer error $\mathcal{L}(\mathcal{Q}, \mathcal{T})$ by the empirical multi-task error $\hat{\mathcal{L}}(\mathcal{Q}, S_1, ..., S_n)$ plus several tractable complexity terms. Similar to the meta-learning PAC-Bayes bounds of Amit and Meir (2018), our proof consists of two main steps. Firstly, 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). Secondly, we bound the transfer-error on the meta-learning which is due to the fact that the meta-learning only receives finitely many tasks τ_i from the task-environment $\mathcal{T}$. Finally, we employ a union bound argument to obtain the result stated in Theorem 2.

While previous PAC-Bayesian bounds for meta-learning (Pentina and Lampert, 2014; Amit and Meir, 2018) assume bounded loss functions, Theorem 2 also provides guarantees

for *unbounded* loss functions. This makes it particularly relevant for regression tasks where unbounded loss functions such as the negative log-likelihood or MSE are often used. The results for unbounded losses require regularity conditions in the form of the sub-gamma assumption, which ensures that the distribution of losses is not heavy-tailed. For instance in a Bayesian linear regression setup, Germain et al. (2016) show that the negative log-likelihood loss is indeed sub-gamma.

Note that, in case of a sub-gamma loss, the bound always maintains a gap $\frac{s^2}{1-c}$ between empirical multi-task and transfer error which does not decay to zero as $n, m \to \infty$. While asymptotically consistent bounds (e.g., Pentina and Lampert (2014)) are theoretically more appealing, in practice they were shown to provide good guarantees only for large samples sizes. In particular, sub-gamma PAC-Bayesian bounds have been shown to be much tighter in simple regression scenarios with limited data ($m \lesssim 10^4$) (Germain et al., 2016). The tradeoff between computational tractability and asymptotic consistency for this particular bound is further discussed in Appendix A.5.

The scale and variance parameters in (11) can be understood as implicit measures of task-environment complexity. If the tasks $\tau \sim \mathcal{T}$ are very similar and simple, a relatively small c and s^2 will suffice to bound the moment generating function of the empirical multi-task error distribution while dissimilar and harder tasks would require $\frac{s^2}{1-c}$ to be large. Note that in order to compute the exact value of the bound, c and s^2 would have to be estimated for every environment, which is challenging for complex environments (Germain et al., 2016). However, the bound can be efficiently minimized for meta-learning purposes without knowledge of these constants. We thus leave the problem of estimating them for future work.

While Theorem 2 holds for any base learner $Q(S, P)$, in practice, we would preferably want to use the learner that gives us optimal performance guarantees. As discussed in Section 3.3, the Gibbs posterior not only minimizes PAC-Bayesian error bounds, but also constitutes a generalization of the Bayesian posterior. Under the assumption that we use the Gibbs posterior as base learner, the bound in (9) can be re-stated in terms of the partition function $Z(S_i, P)$:

Corollary 1. *When choosing the Gibbs posterior $Q^*(S_i, P) := P(h)\exp(-m_i\hat{\mathcal{L}}(S_i, h))/Z(S_i, P)$ as a base learner, under the same assumptions as in Theorem 2, with probability $1-\delta$ it holds that*

$$\mathcal{L}(\mathcal{Q}, \mathcal{T}) \leq -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\mathbb{E}_{P\sim\mathcal{Q}}\left[\ln Z(S_i, P)\right] + \left(\frac{1}{n} + \frac{1}{\tilde{m}}\right) D_{KL}(\mathcal{Q}||\mathcal{P}) + C(\delta, n, \tilde{m}) \ . \tag{12}$$

Proof. See Appendix A.4. □

Since this bound assumes a PAC-optimal base learner, it is tighter than the bound in (9), which holds for any (potentially sub-optimal) $Q \in \mathcal{M}(\mathcal{H})$. If we choose the negative log-likelihood as the loss function, then $\ln Z(S_i, P)$ coincides with the marginal log-likelihood, which is tractable for many popular learning models, such as GPs. Accordingly, the bound in (12) consists of the expected marginal log-likelihood under the hyper-posterior $\mathcal{Q}$ as well as the KL-divergence term which serves as a regularization on the meta-level. As the number of training tasks n grows, the relative weighting of the KL term in (12) shrinks. This is consistent with the general notion that regularization should be strong if only little data is available and vanish asymptotically as $n, m \to \infty$.

Theorem 2 and previous meta-level PAC-Bayes bounds (Pentina and Lampert, 2014; Amit and Meir, 2018) explicitly depend on the task specific posteriors $Q_i = Q(S_i, P)$. However, determining the Q_i can itself be a non-trivial optimization problem. Thus, employing such bounds as a meta-learning objective typically results in nested optimization problems (see Amit and Meir, 2018). In contrast, the bound in (12) no longer depends on Q_i, making it much easier to optimize as a meta-learning objective.

5. Meta-Learning the Hyper-Posterior

5.1. Optimizing the PAC-Bayes bound

In our PAC-Bayesian framework, the meta-learner is given a hyper-prior and data from several tasks, based on which it chooses a hyper-posterior $\mathcal{Q}(P)$. So far, we have not made any assumptions on how such a hyper-posterior is chosen. Building on the generalization bounds stated in the previous section, we now discuss how a meta-learner should select $\mathcal{Q}$.

Intuitively, we want a meta-learner that minimizes the transfer-error. Since we cannot directly compute the transfer-error, we may resort to minimizing its upper-bound in (12) with respect to $\mathcal{Q}$. In general, this is a hard problem since it would require a minimization over $\mathcal{M}(\mathcal{M}(\mathcal{H}))$, the space of all probability measures over priors. However, by invoking Lemma 1 on a monotone transformation of the PAC-Bayes bound in (12), we are able to derive the minimizing distribution $\mathcal{Q}^*$ in closed form.

Proposition 1. ***(PAC-Optimal Hyper-Posterior)*** *Given a hyper-prior $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ and datasets $S_1, ..., S_n$, the hyper-posterior $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ that optimizes the PAC-Bayesian meta-learning bound in (12) is given by*

$$\mathcal{Q}^*(P) = \frac{\mathcal{P}(P)\exp\left(\frac{\tilde{m}}{\tilde{m}+n}\sum_{i=1}^{n}\frac{1}{m_i}\ln Z(S_i, P)\right)}{Z^{II}(S_1, ..., S_n, \mathcal{P})} \tag{13}$$

wherein the partition function $Z^{II}(S_1, ..., S_n, \mathcal{P})$ is defined as $Z^{II} = \mathbb{E}_{P\sim\mathcal{P}}\left[\exp\left(\frac{\tilde{m}}{\tilde{m}+n}\sum_{i=1}^{n}\frac{1}{m_i}\ln Z(S_i, P)\right)\right]$.

Proof. See Appendix A.6. □

This gives us a tractable expression for the *PAC-optimal hyper-posterior (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, since it provides the best possible meta-generalization guarantees among all meta-learners in the sense of Theorem 2.

5.2. Approximating the PAC-Optimal Hyper-Posterior

If the (level-I) log-partition function $\ln Z(S_i, P)$ is tractable (e.g., in case of Bayesian linear regression or Gaussian process base learners), we can compute the PACOH $\mathcal{Q}^*$ up to the normalization constant Z^{II}. Such a setup lends itself to classical approximate inference methods (Andrieu et al., 2003; Blei et al., 2016; Liu and Wang, 2016). Thus, Proposition 1 yields an entire class of possible meta-learning methods. We now briefly discuss several tractable approximations of $\mathcal{Q}^*$ which we evaluate empirically in Section 6.

Maximum A Posteriori (MAP). This is the simplest and most crude method, which approximates $\mathcal{Q}^*(P)$ by a Dirac measure $\delta_P(P^*)$ on the prior P^* that maximizes $\mathcal{Q}^*$:

$$P^* = \underset{P \in \mathcal{M}(\mathcal{H})}{\arg\max}\, \mathcal{Q}^*(P) \tag{14}$$

Variational Inference (VI). In case of VI (Blei et al., 2016), we restrict the space of considered hyper-posteriors to a variational family $\mathcal{F} \subset \mathcal{M}(\mathcal{M}(\mathcal{H}))$ and aim to find the posterior in $\mathcal{F}$ that minimizes the KL-divergence to $\mathcal{Q}^*$, that is,

$$\tilde{\mathcal{Q}} = \underset{\mathcal{Q} \in \mathcal{F}}{\arg\min}\, D_{KL}(\mathcal{Q} || \mathcal{Q}^*)\ . \tag{15}$$

It can be shown that the minimizing distribution $\tilde{\mathcal{Q}}$ in (15) is the same as the minimizer of the bound in (12) under the constraint $\mathcal{Q} \in \mathcal{F}$ (see Appendix A.7 for proof). Consequently, we can directly use (12) as an optimization objective.

Stein Variational Gradient Descent (SVGD). SVGD (Liu and Wang, 2016) approximates $\mathcal{Q}^*$ as a set of particles $\hat{\mathcal{Q}} = \{P_1, ..., P_K\}$. Initially, particles $P_k \sim \mathcal{P}$ (which in our case are priors) are sampled randomly. Then, the method iteratively transports the set of particles to match $\mathcal{Q}^*$, by applying a form of functional gradient descent that minimizes $D_{KL}(\hat{\mathcal{Q}} | \mathcal{Q}^*)$ in the reproducing kernel Hilbert space induced by a kernel function $k(\cdot, \cdot)$. In each iteration, the particles are updated by $P_k \leftarrow P_k + \eta_t \phi^*(P_k)$ with

$$\phi^*(P) = \frac{1}{K} \sum_{l=1}^{K} \left[k(P_l, P) \nabla_{P_l} \ln \mathcal{Q}^*(P_l) + \nabla_{P_l} k(P_l, P) \right]$$

where η_t is a (potentially time-dependent) step size.

6. Empirical Study: PAC-Bayesian Meta-Learning of Gaussian Process Priors

In this section, we instantiate *PACOH*, our generic PAC-Bayesian meta-learning framework from Section 5, with a specific learning model, namely Gaussian processes (GPs). For a review of GP regression, see Appendix B and Rasmussen and Williams (2006). In particular, we meta-learn the GP prior $\mathcal{GP}(m_\theta(x), k_\theta(x, x'))$, i.e. the mean and kernel function, parametrized by neural networks. For details, we refer to Appendix C. We empirically evaluate the proposed meta-learning techniques on multiple simulated and real-world regression environments. In particular, we provide quantitative and qualitative evidence that overfitting to meta-tasks *(meta-overfitting) is a problem* that can be *alleviated using PACOH*, and that our method improves upon previous approaches in terms of *predictive accuracy* and *uncertainty calibration*. An implementation of PACOH for GPs is publicly available as *open source code*.[1]

6.1. Meta-Learning environments

In our experiments, we consider two synthetic and three real-world meta-learning environments for regression. Similar to Harrison et al. (2018) and Fortuin and Rätsch (2019) we use *sinusoid* functions that differ in their amplitude, phase-shift, slope and intercept. As a second synthetic environment, we employ the density of 2-dimensional mixtures of *Cauchy* distributions plus random functions sampled from a GP-prior with squared exponential (SE) kernel.

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 case of *PhysioNet*, each task corresponds to one patient and the target is to predict a patient's health trajectory based on a few measurements, in particular the Glasgow Coma Scale (*GCS*) and the hematocrit value (*HCT*).

Since our work focuses on meta-learning with limited data, we restrict the number of tasks and samples drawn from the different environments. Note that in case of PhysioNet, the number of measurements available for training differs across patients. Further details on the environments can be found in Appendix D.1.

6.2. PACOH combats meta-overfitting

The majority of previous work on meta-learning (e.g., Finn et al., 2017; Garnelo et al., 2018) assumes that tasks $\tau \sim \mathcal{T}$ are of large or infinite supply during meta-training. As a result, when presented with only a limited number of tasks, most existing meta-learning algorithms suffer from severe overfitting on the meta-training tasks, which adversely impacts their performance on unseen tasks from the same environment (Qin et al., 2018). In contrast to the majority of literature on meta-learning, our proposed class of algorithms based on PACOH offers a principled treatment of overfitting. Particularly, the KL-divergence $D_{KL}(\mathcal{Q} || \mathcal{P})$ between

[1] Link to the code repository will be included upon acceptance

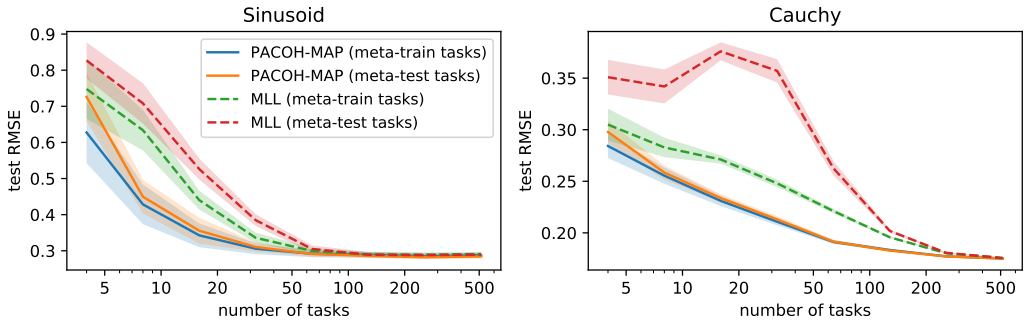


Figure 2: Test RMSE measured on the meta-training tasks and the meta-testing tasks as a function of the number of meta-training tasks for PACOH (ours) and MLL. The performance gap between the meta-train and meta-test tasks clearly demonstrates overfitting in the MLL method, while PACOH performs consistently better and does not overfit.

	Sinusoid	Cauchy	SwissFEL	Physionet-GCS	Physionet-HCT
GP	1.260 ± 0.000	0.287 ± 0.000	0.811 ± 0.000	1.902 ± 0.000	2.904 ± 0.000
PACOH-MAP (ours)	0.301 ± 0.012	0.221 ± 0.003	0.447 ± 0.039	$\mathbf{1.452 \pm 0.014}$	$\mathbf{2.332 \pm 0.010}$
PACOH-VI (ours)	$\mathbf{0.276 \pm 0.005}$	0.226 ± 0.004	0.382 ± 0.022	1.638 ± 0.015	2.391 ± 0.018
PACOH-SVGD (ours)	$\mathbf{0.271 \pm 0.001}$	$\mathbf{0.205 \pm 0.003}$	$\mathbf{0.350 \pm 0.008}$	$\mathbf{1.457 \pm 0.019}$	$\mathbf{2.351 \pm 0.022}$
MLAP	0.421 ± 0.012	0.231 ± 0.616	0.454 ± 0.007	1.893 ± 0.023	2.813 ± 0.232
MLL	0.443 ± 0.034	0.399 ± 0.009	1.175 ± 0.167	1.491 ± 0.025	2.536 ± 0.039
NP	0.466 ± 0.010	0.222 ± 0.002	0.556 ± 0.055	1.846 ± 0.028	2.697 ± 0.074
MAML	0.426 ± 0.029	0.295 ± 0.014	0.958 ± 0.090	1.833 ± 0.132	2.556 ± 0.029

Table 1: Comparison of different meta-learning methods and a standard GP as baseline. Across 5 meta-learning environments, we report test RMSE and its standard deviation. Our proposed method achieves the best performance across all tasks.

hyper-prior and hyper-posterior acts as a regularizer.

To demonstrate the issue of meta-overfitting and the importance of meta-level regularization, we compared the performance of our proposed regularization (PACOH-MAP) to meta-learning based on the marginal log-likelihood (MLL) (Fortuin and Rätsch, 2019), which uses an identical GP model without regularization. Figure 2 shows that the MLL method performs significantly better on the meta-train tasks than on the meta-test tasks in both of our synthetic environments. Our method, in contrast, does not exhibit this discrepancy and performs better than MLL on both kinds of tasks. As expected, this effect is particularly pronounced when the number of meta-training tasks is small (i.e., less than 100). Similar results for other meta-learning methods as well as observations regarding the influence of the regularization strength can be found in the appendix (Sec. D.4). This demonstrates the issue of meta-overfitting, underpins the importance of meta-level regularization, and shows that our proposed framework can be an effective measure to alleviate this problem.

6.3. PACOH improves the predictive performance

We now assess the empirical performance of our proposed PAC-Bayesian algorithms and compare them against popular meta-learning methods across five regression environments.

As described in Section 5, we evaluate three approximations of the PACOH $\mathcal{Q}^*$, namely the MAP estimate (*PACOH-MAP*), VI within the family of fully-factorized Gaussians (*PACOH-VI*), and SVGD with 10 particles (*PACOH-SVGD*). We compare our methods against a standard *GP* with SE kernel (no meta-learning), a GP with neural-network-based mean and kernel function, meta-learned by maximizing the marginal log-likelihood (*MLL*) (Fortuin and Rätsch, 2019), a neural process (*NP*) (Garnelo et al., 2018), model-agnostic meta-learning (*MAML*) (Finn et al., 2017), and the PAC-bound by Amit and Meir (2018) which requires nested optimization (MLAP).

Table 1 reports the results of our study in terms of the root mean squared error (RMSE) on unseen test tasks. In addition, Table S2 in Appendix D.4 shows respective test log-likelihoods. In the majority of environments, all our proposed methods outperform the other meta-learning approaches. Notably, the SVGD approximation consistently performs best on all tasks. This provides further evidence that the introduced meta-learning framework is not only sound, but also yields a class of algorithms that work well in practice. Especially when only few training tasks are available (e.g., in case of the SwissFEL), the performance difference to previous meta-learning methods like NPs and MAML is substantial. As already observed in the meta-overfitting experiments, the performance gap between our method and the baselines decreases with an increasing number of meta-training tasks (see also Fig. S5 in the appendix).

	Sinusoid	Cauchy	SwissFEL	Physionet-GCS	Physionet-HCT
GP	0.171 ± 0.000	0.099 ± 0.000	0.136 ± 0.000	0.339 ± 0.000	0.377 ± 0.000
PACOH-MAP (ours)	0.141 ± 0.007	$\mathbf{0.066 \pm 0.001}$	0.053 ± 0.005	$\mathbf{0.296 \pm 0.002}$	$\mathbf{0.324 \pm 0.001}$
PACOH-VI (ours)	$\mathbf{0.118 \pm 0.003}$	$\mathbf{0.067 \pm 0.002}$	0.051 ± 0.017	0.301 ± 0.001	0.329 ± 0.003
PACOH-SVGD (ours)	$\mathbf{0.115 \pm 0.001}$	$\mathbf{0.065 \pm 0.001}$	$\mathbf{0.042 \pm 0.005}$	$\mathbf{0.297 \pm 0.002}$	$\mathbf{0.325 \pm 0.003}$
MLAP	0.118 ± 0.004	0.084 ± 0.006	0.067 ± 0.007	0.300 ± 0.001	0.349 ± 0.005
MLL	0.232 ± 0.009	0.264 ± 0.001	0.259 ± 0.007	$\mathbf{0.298 \pm 0.003}$	0.342 ± 0.004
NP	0.134 ± 0.012	0.078 ± 0.004	0.252 ± 0.031	0.347 ± 0.007	0.370 ± 0.006
MAML	N/A	N/A	N/A	N/A	N/A

Table 2: Quality of uncertainty estimates induced by different meta-learners. Across 5 meta-learning environments, we report the calibration error and its standard deviation. PACOH consistently yields the best uncertainty calibration.

Note that the MLL method is similar to PACOH-MAP, except that it does not incorporate a hyper-prior as meta-level regularization. The fact that PACOH-MAP improves upon MLL in all environments indicates that MLL suffers from meta-overfitting and demonstrates that incorporating a hyper-prior, as suggested by the PAC-Bayesian theory, constitutes an effective way of regularization. This is consistent with our findings in Section 6.2.

6.4. PACOH improves the uncertainty estimates

Reasoning about the predictive uncertainty of models is crucial in many machine learning applications (Chaloner and Verdinelli, 1995; Shahriari et al., 2016). While Bayesian methods (such as GPs) provide a general framework for quantifying uncertainty, the associated uncertainty estimates are often not well-calibrated due to model misspecification and poorly chosen priors (Kuleshov et al., 2018). We hypothesize that by acquiring the prior in a principled data-driven manner (e.g., with PACOH), we can improve the quality of the predictors' uncertainty estimates, that is, their calibration.

Calibration refers to the consistency between the distributional forecasts and the distribution of the actual observations (Gneiting et al., 2007). For example in regression, calibration means that data points y_j should fall into the 90% confidence interval of our predictor approximately 90% of the time (Kuleshov et al., 2018). The *calibration error* measures the discrepancy between predicted confidence regions and actual frequencies of test data in the respective areas. We follow the definition of Kuleshov et al. (2018) but report the square root of the calibration error for ease of interpretability. For details, we refer to Appendix D.2.1.

Table 2 reports the calibration errors for the different meta-learning methods and environments. Note that the concept of calibration is not applicable to MAML since it only produces point estimates during meta-testing. Compared to the standard GP with a fixed prior, meta-learning a flexible GP prior with PACOH improves the calibration significantly. In contrast, due to its insufficient regularization, MLL produces over-confident predictions, resulting in substantially higher calibration errors than even the standard GP baseline. Once more, the SVGD approximation consistently achieves the best results across all environments. Overall, this demonstrates that PAC-Bayesian meta-learning can be a practical alternative to hand-designed priors, yielding improved predictions as well as better uncertainty estimates.

7. Conclusion

We presented new PAC-Bayesian generalization bounds for meta-learning, including the first ones for unbounded loss functions. By utilizing the specific structure of the base learner, we were able to obtain meta-level generalization bounds that do not depend on the individual task posteriors Q_i and are tighter than the general bounds. Additionally, we derived the PAC-optimal hyper-posterior that promises the best performance guarantees. Overall, our contributions transform PAC-Bayesian meta-learning from a previously nested optimization problem (c.f., Amit and Meir (2018)) into standard approximate inference on $\mathcal{Q}^*$. When employing PACOH on GPs, we achieved strong empirical results. Particularly in settings where meta-training data is scarce and good uncertainty estimates are crucial, PACOH offers a practical alternative to hand-designed priors.

References

Pierre Alquier, James Ridgway, Nicolas Chopin, and Yee Whye Teh. On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research*, 17(236):1–41, 2016.

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended pac-bayes theory. In *ICML*, 2018.

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003. ISSN 08856125. doi: 10.1023/A:1020281327116.

Marcin Andrychowicz, Misha Denil, Sergio Gómez Colmenarejo, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning

to learn by gradient descent by gradient descent. Technical report, 2016. URL https://arxiv.org/pdf/1606.04474.pdf.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Sanjoy Dasgupta and David McAllester, editors, *ICML*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. jan 2016. doi: 10.1080/01621459.2017.1285773.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

Stephane Boucheron, Gabor Lugosi, and Pascal Massart. *Concentration inequalities : a nonasymptotic theory of independence*. Oxford University Press, 2013. ISBN 9780199535255.

Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statist. Sci.*, 10(3):273–304, 08 1995. doi: 10.1214/ss/1177009939. URL https://doi.org/10.1214/ss/1177009939.

Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando De Freitas. Learning to Learn without Gradient Descent by Gradient Descent. In *ICML*, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Vincent Fortuin and Gunnar Rätsch. Deep mean functions for meta-learning in gaussian processes. *arXiv preprint arXiv:1901.08098*, 2019.

Vincent Fortuin, Gunnar Rätsch, and Stephan Mandt. Multivariate time series imputation with variational autoencoders. *arXiv preprint arXiv:1907.04155*, 2019.

Yarin Gal. Uncertainty in deep learning. 2016.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, pages 1884–1892, 2016.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007. doi: 10.1111/j.1467-9868.2007.00587.x.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Benjamin Guedj. A primer on pac-bayesian learning. *arXiv preprint arXiv:1901.05353*, 2019.

Stephen José Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. In *NIPS*, page 177–185, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc. ISBN 1558600159.

James Harrison, Apoorva Sharma, and Marco Pavone. Meta-learning priors for efficient online bayesian regression. *arXiv preprint arXiv:1807.08912*, 2018.

Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning To Learn Using Gradient Descent. In *ICANN*, pages 87–94, 2001.

Johannes Kirschner, Mojmir Mutný, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. In *Proc. International Conference for Machine Learning (ICML)*, June 2019a.

Johannes Kirschner, Manuel Nonnenmacher, Mojmir Mutný, Nicole Hiller, Andreas Adelmann, Rasmus Ischebeck, and Andreas Krause. Bayesian optimization for fast and safe parameter tuning of swissfel. In *Proc. International Free-Electron Laser Conference (FEL2019)*, June 2019b.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *ICML*, 2018.

Guy Lever, François Laviolette, and John Shawe-Taylor. 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.

Qiang Liu and Dilin Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. aug 2016. URL http://arxiv.org/abs/1608.04471.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

David J.C. MacKay. Bayesian interpolation. *Neural computation*, 4:415–447, 1991.

David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

Daniel McNeish. On using bayesian methods to address small sample problems. *Structural Equation Modeling: A Multidisciplinary Journal*, 23(5):750–773, 2016.

Christopher J Milne, Thomas Schietinger, Masamitsu Aiba, Arturo Alarcon, Jürgen Alex, Alexander Anghel, Vladimir Arsov, Carl Beard, Paul Beaud, Simona Bettoni, et al. Swissfel: the swiss x-ray free electron laser. *Applied Sciences*, 7(7):720, 2017.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018.

Cheng Soon Ong, Alexander J. Smola, and Robert C. Williamson. Learning the Kernel with Hyperkernels. *JMLR*, 3:1043–1071, 2005.

Anastasia Pentina and Christoph Lampert. A pac-bayesian bound for lifelong learning. In *International Conference on Machine Learning*, pages 991–999, 2014.

Yunxiao Qin, Weiguo Zhang, Chenxu Zhao, Zezheng Wang, Hailin Shi, Guojun Qi, Jingping Shi, and Zhen Lei. Rethink and redesign meta learning. *arXiv preprint arXiv:1812.04955*, 2018.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's Razor. In *NIPS*, volume 13, pages 294—-300, 2001.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes in machine learning*. 2006.

David Reeb, Andreas Doerr, Sebastian Gerwinn, and Barbara Rakitsch. Learning gaussian processes by minimizing pac-bayesian generalization bounds. In *Advances in Neural Information Processing Systems*, pages 3337–3347, 2018.

Jonas Rothfuss, Fabio Ferreira, Simon Boehm, Simon Walther, Maxim Ulrich, Tamim Asfour, and Andreas Krause. Noise regularization for conditional density estimation. *arXiv preprint arXiv:1907.08982*, 2019a.

Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. In *ICLR*, 2019b.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, Timothy Lillicrap, and Google Deepmind. Meta-Learning with Memory-Augmented Neural Networks. In *ICML*, 2016.

Juergen Schmidhuber. *Evolutionary principles in self-referential learning. On learning how to learn: The meta-meta-... hook*. PhD thesis, Technische Universitaet Munchen, 1987.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, Jan 2016. ISSN 1558-2256.

Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pages 245–248. IEEE, 2012.

Sebastian Thrun and Lorien Pratt, editors. *Learning to Learn*. Kluwer Academic Publishers, USA, 1998. ISBN 0792380479.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342, 2018.

Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*, pages 1012–1019, 2005.

Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006.

Appendix

A. Proofs and Derivations

A.1. Bounding $\Psi(\beta, k)$ in Theorem 1

Below we state a generic version of Theorem 1 that uses notation which will make it easier for us to incorporate its results in later proofs.

Theorem A1 (Alquier et al. (2016)). *Given a data distribution $\mathcal{D} \in \mathcal{M}(\mathcal{Z})$, a hypothesis space $\mathcal{F}$, a loss function $l : \mathcal{F} \times \mathcal{Z} \to \mathbb{R}$, a prior distribution $\pi \in \mathcal{M}(\mathcal{F})$, a real number $\delta \in (0, 1]$, and a real number $\beta > 0$, with probability at least $1 - \delta$ over samples $S \sim \mathcal{D}^k$, we have $\forall \rho \in \mathcal{M}(\mathcal{F})$:*

$$\mathcal{L}(\rho, \mathcal{D}) \leq \hat{\mathcal{L}}(\rho, S) + \frac{1}{\beta}\left[D_{KL}(\rho||\pi) + \ln\frac{1}{\delta} + \Psi(\beta, k)\right] ,$$

where

$$\Psi(\beta, k) = \ln \mathbb{E}_{h\sim\pi}\mathbb{E}_{S\in\mathcal{D}^k} \exp\left[\beta\left(\mathcal{L}(h, \mathcal{D}) - \hat{\mathcal{L}}(h, S)\right)\right] . \tag{16}$$

Bounded loss function When the loss function is bounded, that is, $l : \mathcal{F} \times \mathcal{Z} \to [a, b]$, we can use Hoeffding's lemma to bound $\Psi(\beta, n)$. In particular, we define the random variable $l_j = l(h, z_j) \in [a, b]$ with expected value $\mathbb{E}[l_j] = \mathcal{L}(h, \mathcal{D})$ and write

$$\begin{aligned}\Psi(\beta, k) &= \ln \mathbb{E} \exp\left(\frac{\beta}{k}\sum_{j=1}^{k}(\mathbb{E}[l_j] - l_j)\right) = \sum_{j=1}^{k} \ln \mathbb{E} \exp\left(\frac{\beta}{k}(\mathbb{E}[l_j] - l_j)\right) \\ &\leq \sum_{j=1}^{k} \ln \mathbb{E} \exp\left(\frac{\beta^2(b-a)^2}{8k^2}\right) = \frac{\beta^2(b-a)^2}{8k} .\end{aligned} \tag{17}$$

Sub-gamma loss 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 $\Gamma(s, c)$:

$$\ln \mathbb{E}_{h\sim\pi}\mathbb{E}_{z\sim\mathcal{D}}\left[e^{\lambda V}\right] \leq \frac{\lambda^2 s^2}{2(1 - c\lambda)} \quad \forall \lambda \in (0, 1/c) .$$

For details see Boucheron et al. (2013) and Germain et al. (2016). We can use the sub-gamma assumption to bound $\Psi(\beta, k)$ as follows

$$\Psi(\beta, k) = \sum_{j=1}^{k} \ln \mathbb{E} \exp\left(\frac{\beta}{k} l_j\right) \leq \frac{\beta^2 s^2}{2k(1 - \frac{c\beta}{k})} . \tag{18}$$

Sub-gaussian loss A *sub-gaussian* loss function with variance s^2 can be considered as a limit case of the previously discussed sub-gamma assumption when $c \to 0^+$. As direct consequence, $\Psi(\beta, n)$ can be bounded by

$$\Psi(\beta, k) \leq \frac{\beta^2 s^2}{2k} . \tag{19}$$

A.2. Proof of Lemma 1

In this section, we provide a proof for a generic version of Lemma 1 that uses general symbols for the different entities of interest. This will make it easier to invoke the lemma in later proofs.

Lemma A1. ***(Catoni, 2007)*** *Let A be a set, $g : A \to \mathbb{R}$ a function, and $\rho \in \mathcal{M}(A)$ and $\pi \in \mathcal{M}(A)$ probability densities over A. Then for any $\beta > 0$ and $\forall a \in A$,*

$$\rho^*(a) := \frac{\pi(a)e^{-\beta g(a)}}{Z} = \frac{\pi(a)e^{-\beta g(a)}}{\mathbb{E}_{a\sim\pi}\left[e^{-\beta g(a)}\right]} \tag{20}$$

is the minimizing probability density

$$\underset{\rho \in \mathcal{M}(A)}{\arg\min}\ \beta \mathbb{E}_{a\sim\rho}\left[g(a)\right] + D_{KL}(\rho||\pi)\ . \tag{21}$$

For ease of exposition, we only provide a proof for the case when $A = \{a_1, ..., a_K\}$ is a finite set. An extension to the general case is straightforward and can be found in Zhang (2006) and Catoni (2007). When A is countable, we can express probability densities as categorical distributions, that is, $\rho(a_k) = \rho_k$ and $\pi(a_k) = \pi_k\ \forall k$, where $\sum_{k=1}^K \rho_k = \sum_{k=1}^K \pi_k = 1$, $\pi_k, \rho_k > 0$.

This allows us to write the task of finding ρ^* as a constrained optimization problem:

$$\underset{\rho \in [0,1]^K}{\arg\min}\, J(\rho) = \underset{\rho \in [0,1]^K}{\arg\min} \sum_{k=1}^K \rho_k \left(\beta g(a_k) + \ln \frac{\rho_k}{\pi_k}\right) \quad \text{s.t.} \quad \sum_{k=1}^K \rho_k = 1\ . \tag{22}$$

The respective Lagrangian is

$$\mathcal{L}(\rho, \lambda) = \sum_{k=1}^K \rho_k \left(\beta g(a_k) + \ln \frac{\rho_k}{\pi_k}\right) - \lambda \left(\sum_{k=1}^K \rho_k - 1\right)\ , \tag{23}$$

with the respective partial derivatives:

$$\frac{\partial \mathcal{L}}{\partial \rho_k} = \beta g(a_k) + \ln \frac{\rho_k}{\pi_k} + 1 - \lambda = 0 \qquad k = 1, ..., K \tag{24}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{k=1}^K \rho_k - 1 = 0\ . \tag{25}$$

From (24) we get that

$$\rho_k = \pi_k e^{\lambda - \beta g(a_k) - 1}\ , \tag{26}$$

which we insert in (25) to identify the Lagrange multiplier as

$$\lambda = 1 - \ln \sum_{k=1}^K \pi_k e^{-\beta g(a_k)}\ . \tag{27}$$

Finally, using (27) in (26) we obtain

$$\rho_k^* = \frac{\pi_k e^{-\beta g(a_k)}}{\sum_{k=1}^K \pi_k e^{-\beta g(a_k)}}\ , \tag{28}$$

which concludes the proof. Note that ρ_k^* fulfills the constraint $\rho \in [0,1]^K$ since $\pi_k \geq 0$ and $\sum_{k=1}^K \pi_k e^{-\beta g(a_k)} > \pi_k e^{-\beta g(a_k)}$. Hence, the Hessian $\nabla_\rho^2 J(\rho) = \text{diag}(\rho_1^{-1}, ..., \rho_K^{-1})$ is positive semi-definite $\forall \rho \in [0, \infty)^K$ and ρ^* is the global minimizer of $J(\rho)$ within the K-dimensional probability simplex. □

A.3. Proof of Theorem 2

Step 1 (Task specific generalization) First, we bound the generalization error of each of the observed tasks $\tau_i = (\mathcal{D}_i, m_i)$, when using a learning algorithm $Q : \mathcal{M} \times \mathcal{Z}^{m_i} \to \mathcal{M}$, which outputs a posterior distribution $Q = Q(S_i, P)$ over hypotheses θ, given a prior distribution P and a data set $S_i \sim \mathcal{D}_i^{m_i}$ of size m_i.

In particular, we use Theorem A1 with the following instantiations. The samples are $Z^k = S_i$ with $k := m_i$ and distribution $\mathcal{D} := \mathcal{D}_i$. Further, we define $f := (P, h)$ as a tuple of a prior distribution P and hypothesis h. This can be understood as a two-level hypothesis, wherein P constitutes a hypothesis of the meta-learning problem and h a hypothesis for solving the supervised task at hand. In a similar manner, we define two-level priors and posteriors, that is, $\pi = (\mathcal{P}, P)$ and $\rho = (\mathcal{Q}, Q(S_i, P))$, denoting the distribution of first sampling P from $\mathcal{Q}$ and then θ from $Q(S_i, P)$.

Using the above definitions, the KL-divergence term can be re-written in the following way:

$$\begin{aligned} D_{KL}(\rho||\pi) &= \mathbb{E}_{f\sim\rho}\left[\ln\frac{\rho(h)}{\pi(h)}\right] = \mathbb{E}_{P\sim\mathcal{Q}}\left[\mathbb{E}_{h\sim Q(P,S_i)}\left[\ln\frac{\mathcal{Q}(P)Q(S_i,P)(h)}{\mathcal{P}(P)P(h)}\right]\right] \\ &= \mathbb{E}_{P\sim\mathcal{Q}}\left[\ln\frac{\mathcal{Q}(P)}{\mathcal{P}(P)}\right] + \mathbb{E}_{P\sim\mathcal{Q}}\left[\mathbb{E}_{h\sim Q(P,S_i)}\left[\ln\frac{Q(S_i,P)(h)}{P(h)}\right]\right] \\ &= D_{KL}(\mathcal{Q}||\mathcal{P}) + \mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q(P,S_i)||P)\right] \ . \end{aligned}$$

Finally, we can bound the task specific generalization error based on Theorem A1 with $\beta := m_i$, obtaining

$$\mathcal{L}(\mathcal{Q},\mathcal{D}_i) \leq \mathbb{E}_{P\sim\mathcal{Q}}\mathbb{E}_{h\sim Q}\left[\hat{\mathcal{L}}(h,S_i)\right] + \frac{1}{m_i}\left(D_{KL}(\mathcal{Q}||\mathcal{P}) + \mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q||P)\right] + \ln\frac{1}{\delta_i} + \Psi(m_i,m_i)\right) \ , \tag{29}$$

which holds over all choices of $(\mathcal{Q},Q) \in \mathcal{M}(\mathcal{M}(\mathcal{H})) \times \mathcal{M}(\mathcal{H})$ with probability at least $1-\delta_i$.

Step 2 (Task environment generalization) In the next step, we bound the generalization on the task-environment level. Let $\tau_i = (\mathcal{D}_i, m_i) \sim \mathcal{T}$ be tasks drawn i.i.d. from the task-environment distribution $\mathcal{T}$. We set $f := P$, $\pi := \mathcal{P}$ and $\rho := \mathcal{Q}$ and define the meta-level empirical loss function as $l(P,S) = \hat{\mathcal{L}}(P,M) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(P,D_i)$. Let $\mathcal{P}$ denote the hyper-prior and $\mathcal{Q}$ the hyper-posterior. As a result from Theorem A1 with $\beta := n$, we obtain

$$\mathcal{L}(\mathcal{Q},\mathcal{T}) \leq \hat{\mathcal{L}}(\mathcal{Q},M) + \frac{1}{n}\left(D_{KL}(\mathcal{Q}||\mathcal{P}) + \ln\frac{1}{\delta_0} + \Psi(n,n)\right) \tag{30}$$

over all $\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ with probability at least $1-\delta_0$

Step 3 (Union bound) Finally, we have to combine the results from Step 1 and 2. For that, we bound the probability of the intersection of the events in (29) and (30) with a union bound argument. In particular, for any $\delta > 0$ we set $\delta_i := \frac{\delta}{2n}$ and $\delta_0 = \frac{\delta}{2}$. Further, we define $\tilde{m} = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\right)^{-1}$ as the harmonic mean of the sample sizes m_i.

$$\begin{aligned} \mathcal{L}(\mathcal{Q},\mathcal{T}) &\leq \hat{\mathcal{L}}(\mathcal{Q},M) + \frac{1}{n}\left(D_{KL}(\mathcal{Q}||\mathcal{P}) + \ln\frac{2}{\delta} + \Psi(n,n)\right) && (31) \\ &\leq \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\mathcal{Q},\tau_i) + \frac{1}{n}\left(D_{KL}(\mathcal{Q}||\mathcal{P}) + \ln\frac{2}{\delta} + \Psi(n,n)\right) && (32) \\ &\leq \frac{1}{n}\sum_{i=1}^{n}\hat{\mathcal{L}}(\mathcal{Q},S_i) + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\left(D_{KL}(\mathcal{Q}||\mathcal{P}) + \mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q_i||P)\right] + \ln\frac{2n}{\delta} + \Psi(m_i,m_i)\right) && (33) \\ &\quad + \frac{1}{n}\left(D_{KL}(\mathcal{Q}||\mathcal{P}) + \ln\frac{2}{\delta} + \Psi(n,n)\right) && (34) \\ &= \hat{\mathcal{L}}(\mathcal{Q},S_1,...,S_n) + \left(\frac{1}{n}+\frac{1}{\tilde{m}}\right)D_{KL}(\mathcal{Q}||\mathcal{P}) + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q_i||P)\right] && (35) \\ &\quad + \frac{1}{\tilde{m}}\ln\frac{2n}{\delta} + \frac{1}{n}\ln\frac{2}{\delta} + \frac{1}{n}\Psi(n,n) + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\Psi(m_i,m_i) && (36) \\ &= \hat{\mathcal{L}}(\mathcal{Q},S_1,...,S_n) + \left(\frac{1}{n}+\frac{1}{\tilde{m}}\right)D_{KL}(\mathcal{Q}||\mathcal{P}) + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q_i||P)\right] + C(\delta,n,\tilde{m}) \ . && (37) \end{aligned}$$

In that, we defined $C(\delta,n,\tilde{m})$ as

$$C(\delta,n,\tilde{m}) := \frac{1}{\tilde{m}}\ln\frac{2n}{\delta} + \frac{1}{n}\ln\frac{2}{\delta} + \frac{1}{n}\Psi(n,n) + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\Psi(m_i,m_i) \ . \tag{38}$$

Bounded loss function If we assume that the loss function is bounded, that is $l : \mathcal{H} \times \mathcal{Z} \to [a, b]$, we can use (2) to bound $C(\delta, n, \tilde{m})$ as follows:

$$C(\delta, n, \tilde{m}) \leq \frac{1}{\tilde{m}} \ln \frac{2n}{\delta} + \frac{1}{n} \ln \frac{2}{\delta} + \frac{1}{n} \frac{n^2 (b-a)^2}{8n} + \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \frac{m_i^2 (b-a)^2}{8 m_i} \tag{39}$$

$$= \frac{1}{\tilde{m}} \ln \frac{2n}{\delta} + \frac{1}{n} \ln \frac{2}{\delta} + \frac{(b-a)^2}{4} \ . \tag{40}$$

Sub-gamma loss function If we assume that the loss function is sub-gamma with variance factor s^2 and scale parameter c, under the two-level prior $(\mathcal{P}, P)$ and the data distribution $(\mathcal{T}, \mathcal{D})$, we can use (3) to bound $C(\delta, n, \tilde{m})$ as follows:

$$C(\delta, n, \tilde{m}) \leq \frac{1}{\tilde{m}} \ln \frac{2n}{\delta} + \frac{1}{n} \ln \frac{2}{\delta} + \frac{1}{n} \frac{n^2 s^2}{2n(1-c)} + \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \frac{m_i^2 s^2}{2 m_i (1-c)} \tag{41}$$

$$= \frac{1}{\tilde{m}} \ln \frac{2n}{\delta} + \frac{1}{n} \ln \frac{2}{\delta} + \frac{s^2}{1-c} \ . \tag{42}$$

□

A.4. Proof of Corollary 1: PAC-Bayes Meta-Learning Bound with Marginal Likelihood

When we choose the posterior Q as the optimal Gibbs posterior $Q_i^* := Q^*(S_i, P)$, it follows that

$$\hat{\mathcal{L}}(\mathcal{Q}, S_1, ..., S_n) + \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \mathbb{E}_{P \sim \mathcal{Q}} \left[D_{KL}(Q_i^* || P) \right] \tag{43}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{E}_{P \sim \mathcal{Q}} \mathbb{E}_{h \sim Q_i^*} \left[\hat{\mathcal{L}}(h, S_i) \right] + \frac{1}{m_i} \left(\mathbb{E}_{P \sim \mathcal{Q}} \left[D_{KL}(Q_i^* || P) \right] \right) \right) \tag{44}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \left(\mathbb{E}_{P \sim \mathcal{Q}} \mathbb{E}_{h \sim Q_i^*} \left[m_i \hat{\mathcal{L}}(h, S_i) + \ln \frac{Q_i^*(h)}{P(h)} \right] \right) \tag{45}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \left(\mathbb{E}_{P \sim \mathcal{Q}} \mathbb{E}_{h \sim Q_i^*} \left[\sum_{j=1}^{m_i} l(h, z_i) + \ln \frac{P(h) e^{-\sum_{j=1}^{m_i} l(h, z_i)}}{P(h) Z(S_i, P)} \right] \right) \tag{46}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \left(-\mathbb{E}_{P \sim \mathcal{Q}} \left[\ln Z(S_i, P) \right] \right) \ . \tag{47}$$

This allows us to write the inequality in (9) as

$$\mathcal{L}(\mathcal{Q}, \mathcal{T}) \leq -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \mathbb{E}_{P \sim \mathcal{Q}} \left[\ln Z(S_i, P) \right] + \left(\frac{1}{n} + \frac{1}{\tilde{m}} \right) D_{KL}(\mathcal{Q} || \mathcal{P}) + C(\delta, n, \tilde{m}) \ . \tag{48}$$

According to Lemma A1, the Gibbs posterior $Q^*(S_i, P)$ is the minimizer of (45), in particular

$$Q^*(S_i, P) = \frac{P(h) e^{-m_i \hat{\mathcal{L}}(h, S_i)}}{Z(S_i, P)} = \underset{Q \in \mathcal{M}(\mathcal{H})}{\arg\min} \, \mathbb{E}_{h \sim Q} \left[\hat{\mathcal{L}}(h, S_i) \right] + \frac{1}{m_i} D_{KL}(Q || P) \quad \forall P \in \mathcal{M}(\mathcal{H}), \forall i = 1, ..., n \ . \tag{49}$$

Hence, we can write

$$\mathcal{L}(\mathcal{Q}, \mathcal{T}) \leq -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \mathbb{E}_{P \sim \mathcal{Q}} \left[\ln Z(S_i, P) \right] + \left(\frac{1}{n} + \frac{1}{\tilde{m}} \right) D_{KL}(\mathcal{Q} || \mathcal{P}) + C(\delta, n, \tilde{m}) \tag{50}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{P \sim \mathcal{Q}} \left[\min_{Q \in \mathcal{M}(\mathcal{H})} \hat{\mathcal{L}}(Q, S_i) + \frac{1}{m_i} D_{KL}(Q || P) \right] + \left(\frac{1}{n} + \frac{1}{\tilde{m}} \right) D_{KL}(\mathcal{Q} || \mathcal{P}) + C(\delta, n, \tilde{m}) \tag{51}$$

$$\leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{P\sim\mathcal{Q}}\left[\hat{\mathcal{L}}(Q,S_i)+\frac{1}{m_i}D_{KL}(Q||P)\right]+\left(\frac{1}{n}+\frac{1}{\tilde{m}}\right)D_{KL}(\mathcal{Q}||\mathcal{P})+C(\delta,n,\tilde{m}) \tag{52}$$

$$=\hat{\mathcal{L}}(\mathcal{Q},S_1,...,S_n)+\left(\frac{1}{n}+\frac{1}{\tilde{m}}\right)D_{KL}(\mathcal{Q}||\mathcal{P})+\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q_i||P)\right]+C(\delta,n,\tilde{m})\ , \tag{53}$$

which proves that the bound for Gibbs-optimal base learners in (48) and (12) is tighter than the bound in Theorem 2 which holds uniformly for all $Q \in \mathcal{M}(\mathcal{H})$. □

A.5. Alternative PAC-Bayesian bounds for meta-learning Choice of β in Theorem 2

In this section, we discuss alternative bounds to Theorem 2 and Corollary 1 that arise due to small modifications of the proofs. In fact, if we choose $\beta = \sqrt{m_i}$ in Step 1 and $\beta = \sqrt{n}$ in Step 2 of the proof, we obtain the following results:

Theorem 3. *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 and $\sqrt{\tilde{m}} := \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{m_i}}$. For any confidence level $\delta \in (0,1]$ the inequality*

$$\begin{aligned}\mathcal{L}(\mathcal{Q},\mathcal{T}) \leq\ & \hat{\mathcal{L}}(\mathcal{Q},S_1,...,S_n)+\left(\frac{1}{\sqrt{n}}+\frac{1}{\sqrt{\tilde{m}}}\right)D_{KL}(\mathcal{Q}||\mathcal{P})\\ &+\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{m_i}}\mathbb{E}_{P\sim\mathcal{Q}}\left[D_{KL}(Q(S_i,P)||P)\right]\\ &+\tilde{C}(\delta,\sqrt{n},\sqrt{\tilde{m}})\end{aligned} \tag{54}$$

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

$$\tilde{C}(\delta,\sqrt{n},\sqrt{\tilde{m}})=\frac{1}{\sqrt{\tilde{m}}}\ln\frac{2n}{\delta}+\frac{1}{\sqrt{n}}\ln\frac{2}{\delta}+\left(\frac{1}{\sqrt{n}}+\frac{1}{\sqrt{\tilde{m}}}\right)\frac{(b-a)^2}{2}\ . \tag{55}$$

If the loss function is sub-gamma with variance factor s^2 and scale parameter c, under the two-level prior $(\mathcal{P},P)$ and the data distribution $(\mathcal{T},\mathcal{D})$, the inequality holds for

$$\tilde{C}(\delta,\sqrt{n},\sqrt{\tilde{m}})=\frac{1}{\sqrt{\tilde{m}}}\ln\frac{2n}{\delta}+\frac{1}{\sqrt{n}}\ln\frac{2}{\delta}+\frac{s^2}{2(\sqrt{n}-c)}+\frac{1}{n}\sum_{i=1}^{n}\frac{s^2}{2(\sqrt{m_i}-c)}\ . \tag{56}$$

Proof. Analogous to the proof in A.3 with $\beta = \sqrt{m_i}$ in Step 1 and $\beta = \sqrt{n}$ Step 2. □

Corollary 2. *When choosing the Gibbs posterior $Q^*(S_i,P) := P(h)\exp(-\sqrt{m_i}\hat{\mathcal{L}}(S_i,h))/\tilde{Z}(S_i,P)$ with $\tilde{Z}(S_i,P) = \int_{\mathcal{H}} P(h)\exp(-\sqrt{m_i}\hat{\mathcal{L}}(S_i,h))dh$ as a base learner, under the same assumptions as in Theorem 3, with probability $1-\delta$ it holds that*

$$\begin{aligned}\mathcal{L}(\mathcal{Q},\mathcal{T}) \leq\ & -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\mathbb{E}_{P\sim\mathcal{Q}}\left[\ln\tilde{Z}(S_i,P)\right]\\ &+\left(\frac{1}{\sqrt{n}}+\frac{1}{\sqrt{\tilde{m}}}\right)D_{KL}(\mathcal{Q}||\mathcal{P})+C(\delta,n,\sqrt{\tilde{m}})\ .\end{aligned} \tag{57}$$

Proof. Analogous to the proof in Appendix A.4. □

In contrast to Theorem 2, the constants in $\tilde{C}(\delta,\sqrt{n},\sqrt{\tilde{m}})$ now vanish to 0 as n and m_i grow, i.e. $\lim_{n,m_i\to\infty}\tilde{C}(\delta,\sqrt{n},\sqrt{\tilde{m}}) = 0$. However, this desirable property comes with two drawbacks. First, the KL-terms in (54) vanish more slowly with n and m_i than the similar terms in Theorem 2. Second, and more importantly, $\ln\tilde{Z}(S_i,P)$ in Corollary (2) no longer coincides with the Bayesian marginal log-likelihood when $l(h,z) = -p(z|h)$. In contrast, $\ln Z(S_i,P)$ in Corollary 1 is tractable for various Bayesian base learners such as GPs, thus, playing an instrumental role towards converting the bounds into scalable algorithms (c.f. Section 5).

A.6. Proof of Proposition 1: PAC-Optimal Hyper-Posterior

In this section, we derive the hyper-posterior distribution $\mathcal{Q}^* \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ which, given a hyper-prior $\mathcal{P} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))$ and datasets $S_1, ..., S_n$, minimizes the PAC-Bayesian meta-learning bound in (12).

An objective function corresponding to (12) reads as

$$J(\mathcal{Q}) = -\mathbb{E}_{\mathcal{Q}} \left[\frac{\tilde{m}}{\tilde{m}+n} \sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P) \right] + D_{KL}(\mathcal{Q}||\mathcal{P}) \ . \tag{58}$$

To obtain $J(\mathcal{Q})$, we omit all additive terms from (12) that do not depend on $\mathcal{Q}$ and multiply by the scaling factor $\frac{\tilde{m}n}{\tilde{m}+n}$. Since the described transformations are monotone, the minimizing distribution of $J(\mathcal{Q})$, that is,

$$\mathcal{Q}^* = \underset{\mathcal{Q} \in \mathcal{M}(\mathcal{M}(\mathcal{H}))}{\arg\min} \ J(\mathcal{Q}) \ , \tag{59}$$

is also the minimizer of (12). More importantly, $J(\mathcal{Q})$ is structurally similar to the generic minimization problem in (21). Hence, we can invoke Lemma A1 with $A = \mathcal{M}(\mathcal{H})$, $g(a) = -\sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P)$, $\beta = \frac{\tilde{m}}{\tilde{m}+n}$, to show that the optimal hyper-posterior is

$$\mathcal{Q}^*(P) = \frac{\mathcal{P}(P) \exp\left(\frac{\tilde{m}}{\tilde{m}+n} \sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P) \right)}{Z^{\text{II}}(S_1, ..., S_n, \mathcal{P})} \ , \tag{60}$$

wherein

$$Z^{\text{II}}(S_1, ..., S_n, \mathcal{P}) = \mathbb{E}_{P \sim \mathcal{P}} \left[\exp\left(\frac{\tilde{m}}{\tilde{m}+n} \sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P) \right) \right] \ .$$

□

Technically, this concludes the proof of Proposition 1. However, we want to remark the following interesting result:

If we choose $\mathcal{Q} = \mathcal{Q}^*$, the PAC-Bayes bound in (12) can be expressed in terms of the meta-level partition function Z^{II}, that is,

$$\mathcal{L}(\mathcal{Q}, \mathcal{T}) \leq -\frac{\tilde{m}+n}{\tilde{m}n} \ln Z^{\text{II}}(S_1, ..., S_n, \mathcal{P}) + C(\delta, n, \tilde{m}) \ . \tag{61}$$

We omit a detailed derivation of (61) since it is similar to the one in Appendix A.4.

A.7. Proof of the Equivalence of Variational Inference and Minimization of the PAC-Bayes Meta-Learning Bound

We can write the optimal variational distribution $\tilde{\mathcal{Q}}$ with respect to $\mathcal{Q}^*$ as

$$\begin{aligned}
\tilde{\mathcal{Q}} &= \underset{\mathcal{Q} \in \mathcal{F}}{\arg\min} \ D_{KL}(\mathcal{Q}||\mathcal{Q}^*) && (62)\\
&= \underset{\mathcal{Q} \in \mathcal{F}}{\arg\min} \ \mathbb{E}_{P \sim \mathcal{Q}} \left[\ln \mathcal{Q}(P) - \ln \mathcal{Q}^*(P) \right] && (63)\\
&= \underset{\mathcal{Q} \in \mathcal{F}}{\arg\min} \ \mathbb{E}_{P \sim \mathcal{Q}} \left[\ln \mathcal{Q}(P) - \ln \mathcal{P}(P) - \left(\frac{\tilde{m}}{\tilde{m}+n} \sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P) \right) + \ln Z^{\text{II}}(S_1, ..., S_n, \mathcal{P}) \right] && (64)\\
&= \underset{\mathcal{Q} \in \mathcal{F}}{\arg\min} \ KL(\mathcal{Q}||\mathcal{P}) - \frac{\tilde{m}}{\tilde{m}+n} \sum_{i=1}^{n} \frac{1}{m_i} \mathbb{E}_{P \sim \mathcal{Q}} \left[\ln Z(S_i, P) \right] && (65)\\
&= \underset{\mathcal{Q} \in \mathcal{F}}{\arg\min} \ -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \mathbb{E}_{P \sim \mathcal{Q}} \left[\ln Z(S_i, P) \right] + \left(\frac{1}{n} + \frac{1}{\tilde{m}} \right) D_{KL}(\mathcal{Q}||\mathcal{P}) \ . && (66)
\end{aligned}$$

Here, we multiplied (65) with $\left(\frac{1}{n} + \frac{1}{\tilde{m}}\right)$ to obtain (66). Now it is straightforward to see that (66) is the same as the meta-learning PAC-Bayes bound in (12) up to the constant $C(\delta, n, \tilde{m})$. Hence, we can conclude that variational inference with respect to $\mathcal{Q}^*$ is equivalent to minimizing (12) over the same variational family $\mathcal{F}$.

B. Gaussian process regression

In GP regression, each data point corresponds to a feature-target tuple $z_{i,j} = (x_{i,j}, y_{i,j}) \in \mathbb{R}^d \times \mathbb{R}$. 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 $P_\theta(h) = \mathcal{GP}\left(h | m_\theta(x), k_\theta(x, x')\right)$ is specified by a positive definite kernel $k_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a mean function $m_\theta : \mathcal{X} \to \mathbb{R}$.

The empirical loss under the GP posterior Q^* coincides with the negative log-likelihood of regression targets $\mathbf{y}_i$, that is, $\hat{\mathcal{L}}(Q^*, S_i) = -\frac{1}{m_i} \ln p(\mathbf{y}_i | \mathbf{X}_i)$. Under a Gaussian likelihood $p(\mathbf{y}|\mathbf{h}) = \mathcal{N}(\mathbf{y}; h(\mathbf{x}), \sigma^2 I)$, the marginal log-likelihood $\ln Z(S_i, P_\theta) = \ln p(\mathbf{y}_i | \mathbf{X}_i, \theta)$ can be computed in closed form as

$$\begin{aligned} \ln p(\mathbf{y}|\mathbf{X}, \theta) = &- \frac{1}{2} \left(\mathbf{y} - m_{\mathbf{X},\theta})\right)^\top \tilde{K}_{\mathbf{X},\theta}^{-1} \left(\mathbf{y} - m_{\mathbf{X},\theta}\right) \\ &- \frac{1}{2} \ln |\tilde{K}_{\mathbf{X},\theta}| - \frac{m_i}{2} \ln 2\pi \ , \end{aligned} \tag{67}$$

where $\tilde{K}_{\mathbf{X},\theta} = K_{\mathbf{X},\theta} + \sigma^2 I$, with the kernel matrix $K_{\mathbf{X},\theta} = (k_\theta(x_l, x_k))_{l,k=1}^{m_i}$, observation noise variance σ^2, and mean vector $m_{\mathbf{X},\theta} = (m_\theta(x_1), ..., m_\theta(x_{m_i}))^\top$.

Previous work on Bayesian model selection in the context of GPs argues that the log-determinant $\frac{1}{2} \ln |\tilde{K}_{\mathbf{X},\theta}|$ in the marginal log-likelihood (67) acts as a complexity penalty (Rasmussen and Ghahramani, 2001; Rasmussen and Williams, 2006). However, we suspect that this complexity regularization is only effective if the class of considered priors is restrictive, for instance if we only optimize a small number of parameters such as the length- and output scale of a squared exponential kernel. If we consider expressive classes of GP priors (e.g., our setup where the mean and kernel function are neural networks), such a complexity penalty could be insufficient to avoid meta-overfitting. Indeed, this is what we also observe in our experiments (see Sec. 6).

C. PACOH-based Meta-Learning of GP priors

In this section, we provide further details on the three variants of PACOH, introduced in Section 5 of the paper and employed in our experiments. Following Section 6, we instantiate our framework with GP base learners.Since we are interested in meta-learning, we define the mean and kernel function both as parametric functions. Similar to Wilson et al. (2016) and Fortuin and Rätsch (2019), we instantiate m_θ and k_θ as neural networks, where the parameter vector θ can be meta-learned. To ensure the positive-definiteness of the kernel, we use the neural network as feature map $\Phi_\theta(x)$ on top of which we apply a squared exponential (SE) kernel. Accordingly, the parametric kernel reads as $k_\theta(x, x') = \frac{1}{2} \exp\left(-||\Phi_\theta(x) - \Phi_\theta(x')||_2^2\right)$. Both $m_\theta(x)$ and $\Phi_\theta(x)$ are fully-connected neural networks with 4 layers with each 32 neurons and $\tanh$ non-linearities. The parameter vector θ represents the weights and biases of both neural networks. As hyper-prior we choose a zero-mean isotropic Gaussian, that is, $\mathcal{P}(\theta) = \mathcal{N}(0, \sigma_{\mathcal{P}}^2 I)$.

C.1. PACOH-MAP

A maximum a-posteriori (MAP) approximation of $\mathcal{Q}^*$ is the simplest way to obtain a practical meta-learning algorithm from our PAC-Bayesian theory. In particular, it approximates the $\mathcal{Q}^*(P)$ by a Dirac measure $\delta_P(P^*)$ on the prior P^* that maximizes $\mathcal{Q}^*$:

$$P^* = \underset{P \in \mathcal{M}(\mathcal{H})}{\arg\max}\, \mathcal{Q}^*(P) \ . \tag{68}$$

We can restate (68) as minimizing the following objective:

$$J^{\text{MAP}}(P) = -\frac{\tilde{m}}{\tilde{m} + n} \sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P) - \ln \mathcal{P}(P) \ . \tag{69}$$

When we use GP base learners, as described above, the meta-learning objective reduces to the GP's marginal log-likelihood plus L_2-regularization.

$$J^{\text{MAP}}(\theta) = -\frac{\tilde{m}}{\tilde{m} + n} \sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P_\theta) - \frac{1}{2\sigma_{\mathcal{P}}^2} ||\theta||_2^2 \tag{70}$$

To minimize $J^{\text{MAP}}(\theta)$, we use mini-batch gradient descent. Note that here, mini-batches are created on the meta-level. That is, we sample mini-batches of four tasks to compute the gradients of $J^{\text{MAP}}(\theta)$ and use all data-points in the respective

datasets S_i. Since weight-decay is equivalent to the L_2-regularization in (70), we use AdamW as optimizer (Loshchilov and Hutter, 2019) to optimize the first term in (70).

C.2. PACOH-VI

When aiming to approximate $\mathcal{Q}^*$ via variational inference, we consider a family $\mathcal{F} = \{\mathcal{Q}_\gamma | \gamma \in \Gamma\} \subset \mathcal{M}(\mathcal{M}(\mathcal{H}))$ of parametric distributions with parameter vector γ and try to solve the following minimization problem:

$$\tilde{\mathcal{Q}} = \arg\min_{\gamma \in \Gamma} D_{KL}(\mathcal{Q}_\gamma || \mathcal{Q}^*) \ . \tag{71}$$

In our setup, where P_θ is the GP-prior, this optimization problem can be re-stated as minimizing the negative ELBO:

$$J^{\text{VI}}(\gamma) = -\mathbb{E}_{\theta \sim \mathcal{Q}_\gamma} \left[\frac{\tilde{m}}{\tilde{m}+n} \sum_{i=1}^{n} \frac{1}{m_i} \ln Z(S_i, P_\theta) + \ln \mathcal{P}(\theta) - \ln \mathcal{Q}_\gamma(\theta) \right] \ . \tag{72}$$

In particular, we use fully-factorized Gaussians as the variational family, that is, we meta-learn $\gamma = (\mu_\mathcal{Q}, \sigma_\mathcal{Q})$. To estimate the gradients of $J^{\text{VI}}(\gamma)$ with respect to γ, we employ a pathwise gradient estimator, also known as reparametrization trick. This means that we sample a set of K prior parameters $\theta_k := \mu_\mathcal{Q} + \sigma_\mathcal{Q} \epsilon_k, \ \epsilon_k \sim \mathcal{N}(0, I)$ as well as a mini-batch of H datasets $S_1, ..., S_H$ and compute an unbiased gradient estimate of (72) as follows:

$$\nabla_\gamma J^{\text{VI}}(\gamma) \approx -\frac{1}{K} \sum_{k=1}^{L} \nabla_{\mu_\mathcal{Q}, \sigma_\mathcal{Q}} \left(\frac{n}{H} \cdot \frac{\tilde{m}}{\tilde{m}+n} \sum_{h=1}^{H} \frac{1}{m_h} \ln Z(S_h, P_{\theta_k}) + \ln \mathcal{P}(\theta_k) - \ln \mathcal{Q}_\gamma(\theta_k) \right) \ . \tag{73}$$

During gradient descent with $\nabla_\gamma J^{\text{VI}}(\gamma)$, we employ the adaptive learning rate method Adam. Due to the double stochasticity (mini-batches of tasks and mini-batches of $\theta_k \sim \mathcal{Q}_\gamma$), we found that in practice the gradient estimates of the marginal log-likelihood term in (73) are very noisy whereas the second and third term (meta-level KL-divergence) are subject to less variance. As a result, the less noisy gradients of the KL-divergence dominate during gradient-descent, pushing the VI posterior towards the prior which in turn leads to a higher entropy of $\mathcal{Q}_\gamma$ and even noisier gradient estimates for the marginal log-likelihood term. To counteract this explosion in hyper-posterior entropy, we add a weight $0 < \lambda < 1$ in front of $\ln \mathcal{P}(\theta) - \ln \mathcal{Q}_\gamma(\theta)$ which effectively down-scales the effect of $D_{KL}(\mathcal{Q}_\gamma || \mathcal{P})$ and improves results significantly. Here, we treat λ as a hyper-parameter.

C.3. PACOH-SVGD

SVGD (Liu and Wang, 2016) approximates $\mathcal{Q}^*$ as a set of particles $\hat{\mathcal{Q}} = \{P_1, ..., P_K\}$. In our described setup, each particle corresponds to the parameters of the GP prior, that is, $\hat{\mathcal{Q}} = \{\theta_1, ..., \theta_K\}$. Initially, we sample random priors $\theta_k \sim \mathcal{P}$ from our hyper-posterior. Then, the SVGD iteratively transports the set of particles to match $\mathcal{Q}^*$, by applying a form of functional gradient descent that minimizes $D_{KL}(\hat{\mathcal{Q}} | \mathcal{Q}^*)$ in the reproducing kernel Hilbert space induced by a kernel function $k(\cdot, \cdot)$. We choose a squared exponential kernel with length scale (hyper-)parameter ℓ, that is, $k(\theta, \theta') = \exp\left(-\frac{||\theta - \theta'||_2^2}{2\ell}\right)$. In each iteration, the particles are updated by

$$\theta_k \leftarrow \theta_k + \eta_t \phi^*(\theta_k) \ , \quad \text{with} \quad \phi^*(\theta) = \frac{1}{K} \sum_{l=1}^{K} \left[k(\theta_l, \theta) \nabla_{\theta_l} \ln \mathcal{Q}^*(\theta_l) + \nabla_{\theta_l} k(\theta_l, \theta) \right] \ .$$

Here, we can again estimate $\nabla_{\theta_l} \ln \mathcal{Q}^*(\theta_l)$ with a mini-batch of H datasets $S_1, ..., S_H$:

$$\nabla_{\theta_l} \ln \mathcal{Q}^*(\theta_l) = \frac{n}{H} \cdot \frac{\tilde{m}}{\tilde{m}+n} \sum_{h=1}^{H} \frac{1}{m_h} \nabla_{\theta_l} \ln Z(S_h, P_{\theta_l}) + \nabla_{\theta_l} \ln \mathcal{P}(\theta_l) \ .$$

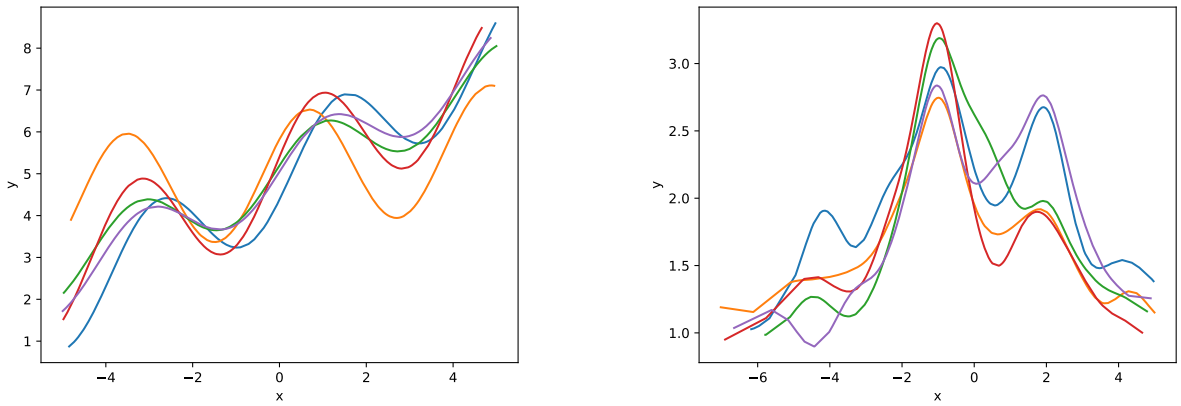
Importantly, $\nabla_{\theta_l} \ln \mathcal{Q}^*(\theta_l)$ does not depend on Z^{II} which makes SVGD tractable.

C.4. Meta-Testing / Target-Training with PACOH

Meta-learning with PACOH, as described above, gives us an approximation of $\mathcal{Q}^*$. In target-testing (see Figure 1), the base learner is instantiated with the meta-learned prior P_θ, receives a dataset $\tilde{S} = (\tilde{\mathbf{X}}, \tilde{\mathbf{y}})$ from an unseen task $\mathcal{D} \sim \mathcal{T}$ and

	Sinusoid	Cauchy	SwissFEL	Physionet
n	20	20	5	100
m_i	5	20	200	4 - 24

Table S1: Number of tasks n and samples per task m_i for the different meta-learning environments.



(a) Sinusoid tasks

(b) Cauchy tasks

Figure S1: Depiction of tasks (i.e., functions) sampled from the Sinusoid and Cauchy task environment, respectively. Note that the Cauchy task environment is two-dimensional ($\dim(\mathcal{X}) = 2$), while (b) displays a one-dimensional projection.

outputs a posterior Q as product of its inference. In our GP setup, Q is the GP posterior and the predictive distribution $\hat{p}(y^*|x^*, \tilde{\mathbf{X}}, \tilde{\mathbf{y}}, \theta)$ is a Gaussian (for details see Rasmussen and Williams, 2006).

Since the meta-learner outputs a distribution over priors, that is, the hyper-posterior $\mathcal{Q}$, we may obtain different predictions for different priors $P_\theta \sim \mathcal{Q}$, sampled from the hyper-posterior. To obtain a predictive distribution under our meta-learned hyper-posterior, we empirically marginalize $\mathcal{Q}$. That is, we draw a set of prior parameters $\theta_1, ..., \theta_K \sim \mathcal{Q}$ from the hyper-posterior, compute their respective predictive distributions $\hat{p}(y^*|x^*, \tilde{\mathbf{X}}, \tilde{\mathbf{y}}, \theta_k)$ and form an equally weighted mixture:

$$\hat{p}(y^*|x^*, \tilde{\mathbf{X}}, \tilde{\mathbf{y}}, \mathcal{Q}) = \mathbb{E}_{\theta\sim\mathcal{Q}}\left[\hat{p}(y^*|x^*, \tilde{\mathbf{X}}, \tilde{\mathbf{y}}, \theta)\right] \approx \frac{1}{K}\sum_{k=0}^{K} \hat{p}(y^*|x^*, \tilde{\mathbf{X}}, \tilde{\mathbf{y}}, \theta_k)\,, \quad \theta_k \sim \mathcal{Q} \tag{74}$$

Since we are concerned with GPs, (74) coincides with a mixture of Gaussians. As one would expect, the mean prediction under $\mathcal{Q}$ (i.e., the expectation of (74)), is the average of the mean predictions corresponding to the sampled prior parameters $\theta_1, ..., \theta_K$. In case of PACOH-VI, we sample $K = 100$ priors from the variational hyper-posterior $\tilde{\mathcal{Q}}$. For PACOH-SVGD, samples from the hyper-posterior correspond to the $K = 10$ particles. PACOH-MAP can be viewed as a special case of SVGD with $K = 1$, that is, only one particle. Thus, $\hat{p}(y^*|x^*, \tilde{\mathbf{X}}, \tilde{\mathbf{y}}, \mathcal{Q}) \approx \hat{p}(y^*|x^*, \tilde{\mathbf{X}}, \tilde{\mathbf{y}}, \theta^{MAP})$ is a single Gaussian.

D. Experiments

D.1. Meta-Learning Environments

In this section, we provide further details on the meta-learning environments used in Section 6. Information about the numbers of tasks and samples in the respective environments can be found in Table S1.

D.1.1. Sinusoids

Each task of the sinusoid environment corresponds to a parametric function

$$f_{a,b,c,\beta}(x) = \beta * x + a * \sin(1.5 * (x - b)) + c\,, \tag{75}$$

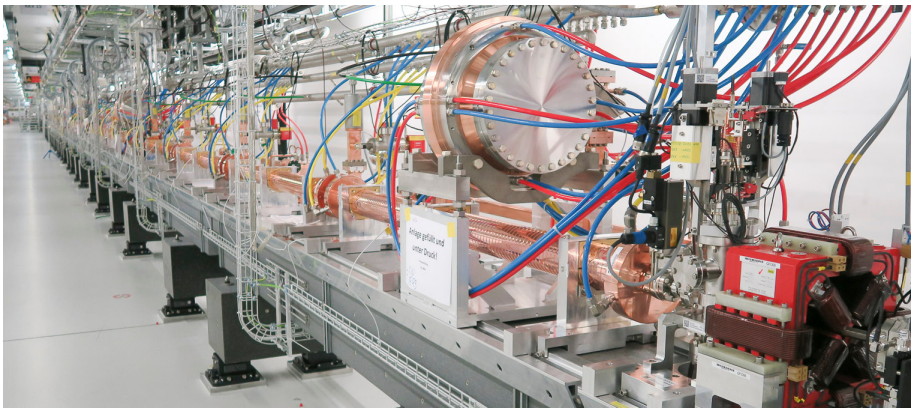


Figure S2: Accelerator of the Swiss Free-Electron Laser (SwissFEL).

which, in essence, consists of an affine as well as a sinusoid function. Tasks differ in the function parameters (a, b, c, β) that are sampled from the task environment $\mathcal{T}$ as follows:

$$a \sim \mathcal{U}(0.7, 1.3), \quad b \sim \mathcal{N}(0, 0.1^2), \quad c \sim \mathcal{N}(5.0, 0.1^2), \quad \beta \sim \mathcal{N}(0.5, 0.2^2) \; . \tag{76}$$

Figure S1a depicts functions $f_{a,b,c,\beta}$ with parameters sampled according to (76). To draw training samples from each task, we draw x uniformly from $\mathcal{U}(-5, 5)$ and add Gaussian noise with standard deviation 0.1 to the function values $f(x)$:

$$x \sim \mathcal{U}(-5, 5) \; , \qquad y \sim \mathcal{N}(f_{a,b,c,\beta}(x), 0.1^2) \; . \tag{77}$$

D.1.2. Cauchy

Each task of the Cauchy environment can be interpreted as a two dimensional mixture of Cauchy distributions plus a function sampled from a Gaussian process prior with zero mean and SE kernel function $k(x, x') = \exp\left(\frac{||x-x'||_2^2}{2l}\right)$ with $l = 0.2$. The (unnormalized) mixture of Cauchy densities is defined as:

$$m(x) = \frac{6}{\pi \cdot (1 + ||x - \mu_1||_2^2)} + \frac{3}{\pi \cdot (1 + ||x - \mu_2||_2^2)} \; , \tag{78}$$

with $\mu_1 = (-1, -1)^\top$ and $\mu_2 = (2, 2)^\top$.

Functions from the task environments are sampled as follows:

$$f(x) = m(x) + g(x) \; , \qquad g \sim \mathcal{GP}(0, k(x, x')) \; . \tag{79}$$

Figure S1b depicts a one-dimensional projection of functions sampled according to (79). To draw training samples from each task, we draw x from a truncated normal distribution and add Gaussian noise with standard deviation 0.05 to the function values $f(x)$:

$$x := \min\{\max\{\tilde{x}, 2\}, -3\} \; , \;\; \tilde{x} \sim \mathcal{N}(0, 2.5^2) \; , \qquad y \sim \mathcal{N}(f(x), 0.05^2) \; . \tag{80}$$

D.1.3. SwissFEL

Free-electron lasers (FELs) accelerate electrons to very high speed in order to generate shortly pulsed laser beams with wavelengths in the X-ray spectrum. These X-ray pulses can be used to map nanometer scale structures, thus facilitating experiments in molecular biology and material science. The accelerator and the electron beam line of a FEL consist of multiple magnets and other adjustable components, each of which has several parameters that experts adjust to maximize the pulse energy (Kirschner et al., 2019a). Due do different operational modes, parameter drift, and changing (latent) conditions, the laser's pulse energy function, in response to its parameters, changes across time. As a result, optimizing the laser's parameters is a recurrent task.

Overall, our meta-learning environment consists of different parameter optimization runs (i.e., tasks) on the SwissFEL, an 800 meter long laser located in Switzerland (Milne et al., 2017). A picture of the SwissFEL is shown in Figure S2. The input space, corresponding to the laser's parameters, has 12 dimensions whereas the regression target is the pulse energy (1-dimensional). For details on the individual parameters, we refer to Kirschner et al. (2019b). For each run, we have around 2000 data points. Since these data-points are generated with online optimization methods, the data are non-i.i.d. and get successively less diverse throughout the optimization. As we are concerned with meta-learning with limited data and want to avoid issues with highly dependent data points, we only take the first 400 data points per run and split them into training and test subsets of size 200. Overall, we have 9 runs (tasks) available. 5 of those runs are used for meta-training and the remaining 4 runs are used for meta-testing.

D.1.4. PHYSIONET

The 2012 Physionet competition (Silva et al., 2012) published an open-access data set of patient stays on the intensive care unit (ICU). Each patient stay consists of a time series over 48 hours, where up to 37 clinical variables are measured. The original task in the competition was binary classification of patient mortality, but due to the large number of missing values (around 80 % across all features), the data set is also popular as a test bed for time series prediction methods, especially using Gaussian processes (Fortuin and Rätsch, 2019; Fortuin et al., 2019).

In this work, we treat each patient as a separate task and the different clinical variables as different environments. We use the Glasgow coma scale (GCS) and hematocrit value (HCT) as environments for our study, since they are among the most frequently measured variables in this data set. From the dataset, we remove all patients where less than four measurements of CGS (and HCT respectively) are available. From the remaining patients we use 100 patients for meta-training and 500 patients each for meta-validation and meta-testing. Here, each patient corresponds to a task. Since the number of available measurements differs across patients, the number of training points m_i ranges between 4 and 24.

D.2. Experimental Methodology

In the following, we describe our experimental methodology and provide details on how the empirical results reported in Section 6 were generated. Overall, evaluating a meta-learner consists of two phases, *meta-training* and *meta-testing*. The latter can be further sub-divided into *target training* and *target testing*. Figure 1 illustrates these different stages for our PAC-Bayesian meta-learning framework.

Meta-training The meta-learner is provided with a set of datasets $S_1, ..., S_2$ and optimizes its respective meta-objective, for instance in case of PACO-VI, the negative ELBO with respect to $\mathcal{Q}^*$.

Meta-testing This phase aims to evaluate the empirical generalization properties of the meta-learned prior knowledge. In particular, we evaluate how well the base learner, instatiated with the prior knowledge, performs on multiple unseen tasks $\tau = (\mathcal{D}, m) \sim \mathcal{T}$. For that, two steps are required: In the *target-training* phase, the base learner is given a training dataset $\tilde{S} \sim \mathcal{D}$ and performs (normal) inference. Then, in *target-testing*, we evaluate its predictions on a test dataset $\tilde{S}^* \sim \mathcal{D}$ from the same task. For PACOH, MLL, GP, and NP, the respective predictor outputs a probability distribution $\hat{p}(y^*|x^*, \tilde{S})$ for the x^* in $\tilde{S}^*$. The respective mean prediction corresponds to the expectation of $\hat{p}$, that is $\hat{y} = \hat{\mathbb{E}}(y^*|x^*, \tilde{S})$. In the case of MAML, only a mean prediction is available. Based on the mean predictions, we compute the *root mean-squared error (RMSE)*:

$$\text{RMSE} = \sqrt{\frac{1}{|\tilde{S}^*|} \sum_{(x^*,y^*)\in S^*} (y^* - \hat{y})^2} \ . \tag{81}$$

Similarly, we compute the *average log-likelihood*:

$$\text{LL} = \frac{1}{|\tilde{S}^*|} \sum_{(x^*,y^*)\in S^*} \ln \hat{p}(y^*|x^*, \tilde{S}) \ , \tag{82}$$

and the *calibration error* (see Appendix D.2.1).

The described meta-training and meta-testing procedure is repeated for five random seeds that influence both the initialization and gradient-estimates of the concerned algorithms. The averages and standard deviations reported in Figure 2 as well as Tables 1, 2 and 2 are based on the results obtained for different seeds.

D.2.1. CALIBRATION ERROR

The concept of calibration applies to probabilistic predictors that, given a new target input x_i, produce a probability distribution $\hat{p}(y_i|x_i)$ over predicted target values y_i. Corresponding to the predictive density, we denote a predictor's cumulative density function (CDF) as $\hat{F}(y_j|x_j) = \int_{-\infty}^{y_j} \hat{p}(y|x_i)dy$. For confidence levels $0 \leq q_h < ... < q_H \leq 1$, we can compute the corresponding empirical frequency

$$\hat{q}_h = \frac{|\{y_j \mid \hat{F}(y_j|x_j) \leq q_h, j = 1, ..., m\}|}{m} \,, \tag{83}$$

based on dataset $S = \{(x_i, y_i)\}_{i=1}^m$ of m samples. If we have calibrated predictions we would expect that $\hat{q}_h \to q_h$ as $m \to \infty$. Similar to (Kuleshov et al., 2018), we can define the calibration error as a function of residuals $\hat{q}_h - q_h$, in particular,

$$\text{calib-err} = \sqrt{\frac{1}{H} \sum_{h=1}^{H} (\hat{q}_h - q_h)^2} \,. \tag{84}$$

In our experiments, we compute (84) with $K = 20$ equally spaced confidence levels between 0 and 1.

D.3. Hyper-Parameter Selection

For each of the meta-environments and algorithms, we ran a separate hyper-parameter search to select the hyper-parameters. In particular, we use the `hyperopt`[2] package (Bergstra et al., 2013) which performs Bayesian optimization based on regression trees. As optimization metric, we employ the average log-likelihood, evaluated on a separate validation set of tasks.

The scripts for reproducing the hyper-parameter search are included in our code repository[3] For the results, reported in the Tables 1, 2, and S2, we provide the selected hyper-parameters and detailed evaluation results under `[Link will be added upon acceptance]`

D.4. Further Experimental Results

Meta-overfitting To further demonstrate the issue of meta-overfitting, we conducted experiments where we varied the regularization intensity and analyzed its impact on the meta-test performance. In particular, we consider the two synthetic environments as well as real-world data from the SwissFEL (see Sec. 6.1). We report results for PACOH-MAP with a zero-mean isotropic Gaussian hyper-prior $\mathcal{P}(\theta) = \mathcal{N}(0, \sigma_{\mathcal{P}}^2 I)$. In this case, the regularization imposed by $D_{KL}(\mathcal{Q}||\mathcal{P})$ coincides with weight decay (that is, L_2-regularization) on the neural network parameters θ. Its strength, controlled by the weight-decay multiplier, corresponds to $\sigma_{\mathcal{P}}^2$ (Hanson and Pratt, 1989). We show the optimal regularization parameters for PACOH-MAP on the different environments in Figure S3. The fact that these optima are non-zero for all environments provides additional evidence of the effect of meta-overfitting.

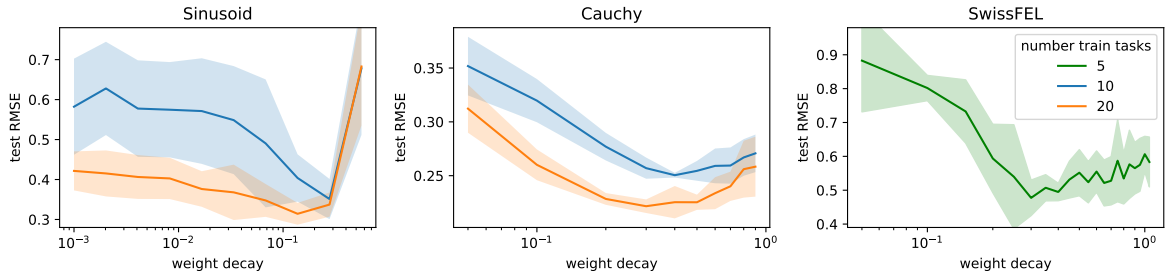


Figure S3: Meta-test RMSE and its standard deviation for PACOH-MAP across different regularization intensities (increasing from left to right), corresponding to $\sigma_{\mathcal{P}}^2$. The optimal performance is achieved at a non-zero regularization, suggesting that the proposed method can effectively combat meta-overfitting. The effect is more pronounced with fewer training tasks.

In order to investigate whether the phenomenon of meta-overfitting, which we have observed consistently for PACOH-MAP and MLL, is also relevant to other meta-learning methods, we also report the meta-train test error and the meta-test test

[2] http://hyperopt.github.io/hyperopt/

[3] [Link will be added upon acceptance]

error across different numbers of tasks. The results, analogous to Figure 2, are plotted in Figure S4, showing a significant difference between the meta-train and meta-test error that vanishes as the number of tasks becomes larger. Once more, this supports our claim that meta-overfitting is a relevant issue and should be addressed in a principled manner.

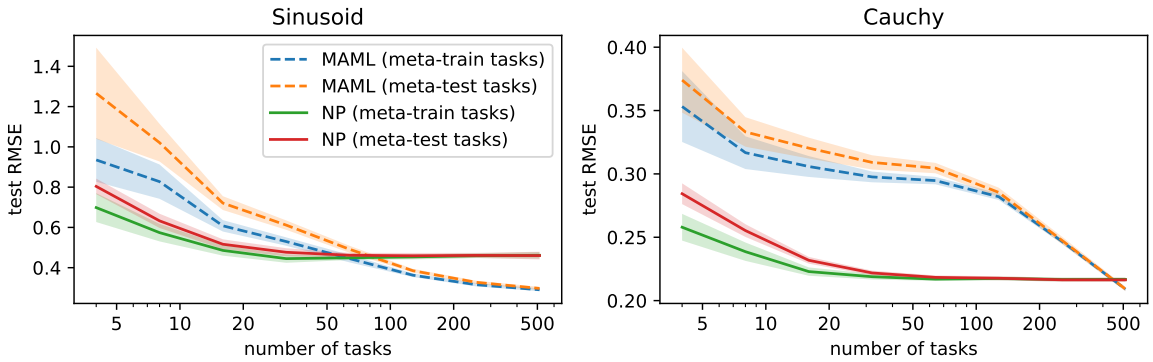


Figure S4: Test RMSE measured on the meta-training tasks and the meta-testing tasks as a function of the number of meta-training tasks for MAML and NPs. The performance gap between the meta-train and meta-test tasks clearly demonstrates overfitting on the meta-level for both methods.

Benchmark comparison across different numbers of tasks In addition to the performance comparison in Table 1, we assessed the performance of the different methods on our synthetic environments as a function of the number of meta-training tasks (Fig. S5). As expected and already seen in the meta-overfitting experiments, the performance gap between the PACOH methods and the baselines is especially pronounced for a small number of meta-training tasks, when meta-overfitting is a more severe problem. However, even for larger numbers of tasks, PACOH-MAP and PACOH-SVGD seem to outperform most baselines. Due to the high computational complexity of MLAP, getting increasingly burdensome as the number of task grows, we did not include MLAP in this experiment.

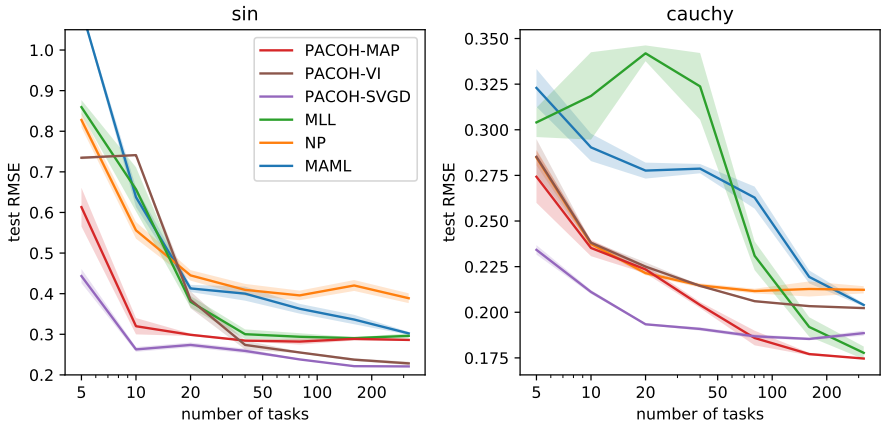


Figure S5: Test RMSE measured on the different environments as a function of the number of meta-training tasks for the different variants of our method (PACOH) and the baselines. Our method outperforms the baselines consistently. This effect is especially pronounced when the number of meta-train tasks is small.

Benchmark comparison in terms of the test log-likelihood Complementary to the the tables in Section 6, we report average test log-likelihoods in Table S2. These results reflect the same findings as discussed in Section 6.

	Sinusoid	Cauchy	SwissFEL	Physionet-GCS	Physionet-HCT
GP	-5.011 ± 0.000	$\mathbf{0.180 \pm 0.001}$	-0.891 ± 0.009	$< -10^6$	-6.054 ± 0.000
PACOH-MAP (ours)	0.127 ± 0.439	0.116 ± 0.010	-0.621 ± 0.085	-1.740 ± 0.005	-2.331 ± 0.001
PACOH-VI (ours)	0.236 ± 0.028	0.140 ± 0.023	-0.4460 ± 0.025	$\mathbf{-1.709 \pm 0.003}$	-2.335 ± 0.003
PACOH-SVGD (ours)	$\mathbf{0.336 \pm 0.022}$	$\mathbf{0.179 \pm 0.009}$	$\mathbf{-0.338 \pm 0.128}$	$\mathbf{-1.711 \pm 0.032}$	$\mathbf{-2.326 \pm 0.002}$
MLAP	-7.364 ± 0.187	0.133 ± 0.364	-1.059 ± 0.471	-2.392 ± 0.009	-2.590 ± 0.049
MLL	-12.87 ± 2.420	-144.9 ± 6.620	-175.8 ± 66.365	-1.739 ± 0.007	-2.350 ± 0.0060
NP	-1.108 ± 0.187	-0.011 ± 0.028	-8.053 ± 2.472	-4.235 ± 0.293	-2.711 ± 0.050
MAML	N/A	N/A	N/A	N/A	N/A

Table S2: Benchmark study of different meta-learning methods as well as a standard GP as baseline. Across 5 meta-learning environments, we report the average test log-likelihood and its standard deviation. Our proposed method achieves the best performance across tasks.