Scalable Bayesian Meta-Learning through Generalized Implicit Gradients

Yilang Zhang, Bingcong Li, Shijian Gao, Georgios B. Giannakis

Dept. of ECE, University of Minnesota, Minneapolis, MN, USA
{zhan7453,lixx5599,gao00379,georgios}@umn.edu

Abstract

Meta-learning owns unique effectiveness and swiftness in tackling emerging tasks with limited data. Its broad applicability is revealed by viewing it as a bi-level optimization problem. The resultant algorithmic viewpoint however, faces scalability issues when the inner-level optimization relies on gradient-based iterations. Implicit differentiation has been considered to alleviate this challenge, but it is restricted to an isotropic Gaussian prior, and only favors *deterministic* meta-learning approaches. This work markedly mitigates the scalability bottleneck by cross-fertilizing the benefits of implicit differentiation to *probabilistic* Bayesian meta-learning. The novel implicit Bayesian meta-learning (iBaML) method not only broadens the scope of learnable priors, but also quantifies the associated uncertainty. Furthermore, the ultimate complexity is well controlled regardless of the inner-level optimization trajectory. Analytical error bounds are established to demonstrate the precision and efficiency of the generalized implicit gradient over the explicit one. Extensive numerical tests are also carried out to empirically validate the performance of the proposed method.

1 Introduction

Over the past decade, deep learning (DL) has garnered huge attention from theory, algorithms, and application viewpoints. The underlying success of DL is mainly attributed to the massive datasets, with which large-scale and highly expressive models can be trained. On the other hand, the stimulus of DL, namely data, can be scarce. Nevertheless, in several real-world tasks, such as object recognition and concept comprehension, humans can perform exceptionally well even with very few data samples. This prompts the natural question: *How can we endow DL with human's unique intelligence?* By doing so, DL's data reliance can be alleviated and the subsequent model training can be streamlined. Several trials have been emerging in those "stimulus-lacking" domains, including speech recognition (Miao, Metze, and Rawat 2013), medical imaging (yang et al. 2016), and robot manipulation (Hansen and Wang 2021).

A systematic framework has been explored in recent years to address the aforementioned question, under the terms *learning-to-learn* or *meta-learning* (Thrun 1998). In brief, meta-learning extracts task-invariant prior information from a given family of correlated (and thus informative) tasks. Domain-generic knowledge can therein be acquired as an inductive bias and transferred to new tasks outside the set of given ones (Thrun and Pratt 2012; Grant et al. 2018), making it feasible to learn unknown models/tasks even with minimal training samples. One representative example is that of an edge extractor, which can act as a common prior owing to its presence across natural images. Thus, using it can prune degrees of freedom from a number of image classification models. The prior extraction in conventional meta-learning is more of a hand-crafted art; see e.g., (Schmidhuber 1993; Bengio, Bengio, and Cloutier 1995; Schmidhuber, Zhao, and Wiering 1996). This rather "cumbersome art" has been gradually replaced by data-driven approaches. For parametric models of the task-learning process (Santoro et al. 2016; Mishra et al. 2018), the task-invariant "sub-model" can then be shared across different tasks with prior information embedded in the model weights. One typical model is that of recurrent neural networks (RNNs), where task-learning is captured by recurrent cells. However, the resultant black-box learning setup faces interpretability challenges.

As an alternative to model-committed approaches, model-agnostic meta-learning (MAML) transforms task-learning to optimizing the task-specific model parameters, while the prior amounts to initial parameters per task-level optimization, that are shared across tasks and can be learned through differentiable meta-level optimization (Finn, Abbeel, and Levine 2017). Building upon MAML, optimization-based meta-learning has been advocated to ameliorate its performance; see e.g. (Li et al. 2017; Bertinetto et al. 2019; Flennerhag et al. 2020; Abbas et al. 2022). In addition, performance analyses have been reported to better understand the behavior of these optimization-based algorithms (Franceschi et al. 2018; Fallah, Mokhtari, and Ozdaglar 2020; Wang, Sun, and Li 2020; Chen and Chen 2022).

Interestingly, the learned initialization can be approximately viewed as the mean of an implicit Gaussian prior over the task-specific parameters (Grant et al. 2018). Inspired by this interpretation, Bayesian methods have been advocated for meta-learning to further allow for uncertainty quantification in the model parameters. Different from its deterministic counterpart, Bayesian meta-learning seeks a prior distribution over the model parameters that

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

best explains the data. Exact Bayesian inference however, is barely tractable as the posterior is often non-Gaussian, which prompts pursuing approximate inference methods; see e.g., (Yoon et al. 2018; Grant et al. 2018; Finn, Xu, and Levine 2018; Ravi and Beatson 2019).

MAML and its variants have appealing empirical performance, but optimizing the meta-learning loss with backpropagation is challenging due to the high-order derivatives involved. This incurs complexity that grows linearly with the number of task-level optimization steps, which renders the corresponding algorithms barely scalable. For this reason, scalability of meta-learning algorithms is of paramount importance. One remedy is to simply ignore the high-order derivatives, and rely on first-order updates only (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018). Alternatively, the so-termed implicit (i)MAML relies on implicit differentiation to eliminate the explicit backpropagation. However, the proximal regularization term in iMAML is confined to be a simple isotropic Gaussian prior, which limits model expressiveness (Rajeswaran et al. 2019).

In this paper, we develop a novel implicit Bayesian meta-learning (iBaML) approach that offers the desirable scalability, expressiveness, and performance quantification, and thus broadens the scope and appeal of meta-learning to real application domains. The contribution is threefold.

i) iBaML enjoys complexity that is invariant to the number K of gradient steps in task-level optimization. This fundamentally breaks the complexity-accuracy tradeoff, and makes Bayesian meta-learning affordable with more sophisticated task-level optimization algorithms.

ii) Rather than an isotropic Gaussian distribution, iBaML allows for learning more expressive priors. As a Bayesian approach, iBaML can quantify uncertainty of the estimated model parameters.

iii) Through both analytical and numerical performance studies, iBaML showcases its complexity and accuracy merits over the state-of-the-art Bayesian meta-learning methods. In a large K regime, the time and space complexity can be reduced even by an order of magnitude.

2 Preliminaries and problem statement

This section outlines the meta-learning formulation in the context of supervised few-shot learning, and touches upon the associated scalability issues.

2.1 Meta-learning setups

Suppose we are given datasets $\mathcal{D}_t := \{(\mathbf{x}_t^n, y_t^n)\}_{n=1}^{N_t}$, each of cardinality $|\mathcal{D}_t| = N_t$ corresponding to a task indexed by $t \in \{1, \ldots, T\}$, where $\mathbf{x}_t^n$ is an input vector, and $y_t^n \in \mathbb{R}$ denotes its label. Set $\mathcal{D}_t$ is disjointly partitioned into a training set $\mathcal{D}_t^{\rm tr}$ and a validation set $\mathcal{D}_t^{\rm val}$, with $|\mathcal{D}_t^{\rm tr}| = N_t^{\rm tr}$ and $|\mathcal{D}_t^{\rm val}| = N_t^{\rm val}$ for $\forall t$. Typically, N_t is limited, and often much smaller than what is required by supervised DL tasks. However, it is worth stressing that the number of tasks T can be considerably large. Thus, $\sum_{t=1}^T N_t$ can be sufficiently large for learning a prior parameter vector shared by all tasks; e.g., using deep neural networks.

A key attribute of meta-learning is to estimate such a task-invariant prior information parameterized by the meta-parameter $\boldsymbol{\theta}$ based on training data *across* tasks. Subsequently, $\boldsymbol{\theta}$ and $\mathcal{D}_t^{\rm tr}$ are used to perform task- or *inner-level optimization* to obtain the task-specific parameter $\boldsymbol{\theta}_t \in \mathbb{R}^d$. The estimate of $\boldsymbol{\theta}_t$ is then evaluated on $\mathcal{D}_t^{\rm val}$ (and potentially also $\mathcal{D}_t^{\rm tr}$) to produce a validation loss. Upon minimizing this loss summed over all the training tasks w.r.t. $\boldsymbol{\theta}$, this meta- or *outer-level optimization* yields the task-invariant estimate of $\boldsymbol{\theta}$. Note that the dimension of $\boldsymbol{\theta}_t$ is not necessarily identical to that of $\boldsymbol{\theta}$; see e.g. (Li et al. 2017; Bertinetto et al. 2019; Lee et al. 2019). As we will see shortly, this nested structure can be formulated as a bi-level optimization problem. This formulation readily suggests application of meta-learning to settings such as hyperparameter tuning that also relies on a similar bi-level optimization (Franceschi et al. 2018).

This bi-level optimization is outlined next for both deterministic and probabilistic Bayesian meta-learning variants.

Optimization-based meta-learning. For each task t, let $\check{\mathcal{L}}_t^{\rm tr}(\boldsymbol{\theta}_t)$ and $\check{\mathcal{L}}_t^{\rm val}(\boldsymbol{\theta}_t)$ denote the losses over $\mathcal{D}_t^{\rm tr}$ and $\mathcal{D}_t^{\rm val}$, respectively. Further, let $\hat{\boldsymbol{\theta}}$ be the meta-parameter estimate, and $\mathcal{R}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_t)$ the regularizer of the learning cost per task t. Optimization-based meta-learning boils down to

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{t=1}^T \check{\mathcal{L}}_t^{\rm val}(\hat{\boldsymbol{\theta}}_t(\boldsymbol{\theta})) \quad (1)$$
$$\text{s.to } \hat{\boldsymbol{\theta}}_t(\boldsymbol{\theta}) = \operatorname*{argmin}_{\boldsymbol{\theta}_t} \check{\mathcal{L}}_t^{\rm tr}(\boldsymbol{\theta}_t) + \mathcal{R}(\boldsymbol{\theta}, \boldsymbol{\theta}_t),\ t = 1, \ldots, T.$$

The regularizer $\mathcal{R}$ can be either implicit (as in iMAML) or explicit (as in MAML). Further, the task-invariant meta-parameter is calibrated by $\mathcal{R}$ in order to cope with overfitting. Indeed, an over-parameterized neural network could easily overfit $\mathcal{D}_t^{\rm tr}$ to produce a tiny $\check{\mathcal{L}}_t^{\rm tr}$ yet a large $\check{\mathcal{L}}_t^{\rm val}$.

As reaching global minima can be infeasible especially with highly nonconvex neural networks, a practical alternative is an estimator $\hat{\boldsymbol{\theta}}_t$ produced by a function $\hat{\mathcal{A}}_t(\boldsymbol{\theta})$ representing an optimization algorithm, such as gradient descent (GD), with a prefixed number K of iterations. Thus, a tractable version of (1) is

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{t=1}^T \check{\mathcal{L}}_t^{\rm val}(\hat{\boldsymbol{\theta}}_t(\boldsymbol{\theta})) \quad (2)$$
$$\text{s.to } \hat{\boldsymbol{\theta}}_t(\boldsymbol{\theta}) = \hat{\mathcal{A}}_t(\boldsymbol{\theta}),\ \ t = 1, \ldots, T$$

As an example, $\hat{\mathcal{A}}_t$ can be an one-step gradient descent initialized by $\hat{\boldsymbol{\theta}}$ with implicit priors ($\mathcal{R}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_t) = 0$) (Finn, Abbeel, and Levine 2017; Grant et al. 2018), which yields the per task parameter estimate

$$\hat{\boldsymbol{\theta}}_t = \hat{\mathcal{A}}_t(\boldsymbol{\theta}) = \boldsymbol{\theta} - \alpha \nabla \check{\mathcal{L}}_t^{\rm tr}(\boldsymbol{\theta}),\ \ t = 1, \ldots, T \quad (3)$$

where α is the learning rate of GD, and we use the compact gradient notation $\nabla \check{\mathcal{L}}_t^{\rm tr}(\boldsymbol{\theta}) := \nabla_{\boldsymbol{\theta}_t} \check{\mathcal{L}}_t^{\rm tr}(\boldsymbol{\theta}_t)\big|_{\boldsymbol{\theta}_t = \boldsymbol{\theta}}$ hereafter. For later use, we also define $\mathcal{A}_t^*$ the (unknown) oracle function that generates the global optimum $\boldsymbol{\theta}_t^*$.

Bayesian meta-learning. The probabilistic approach to meta-learning takes a Bayesian view of the (now random) vector $\boldsymbol{\theta}_t$ per task t. The task-invariant vector $\boldsymbol{\theta}$ is still deterministic, and parameterizes the prior probability density function (pdf) $p(\boldsymbol{\theta}_t;\boldsymbol{\theta})$. Task-specific learning seeks the posterior pdf $p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})$, where $\mathbf{X}_t^{\rm tr} := [\mathbf{x}_t^1,\ldots,\mathbf{x}_t^{N_t^{\rm tr}}]$ and $\mathbf{y}_t^{\rm tr} := [y_t^1,\ldots,y_t^{N_t^{\rm tr}}]^\top$ ($^\top$ denotes transposition), while the objective per task t is to maximize the conditional likelihood $p(\mathbf{y}_t^{\rm val}|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm val},\mathbf{X}_t^{\rm tr},\boldsymbol{\theta}) = \int p(\mathbf{y}_t^{\rm val}|\boldsymbol{\theta}_t;\mathbf{X}_t^{\rm val})p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})d\boldsymbol{\theta}_t$. Along similar lines followed by its deterministic optimization-based counterpart, Bayesian meta-learning amounts to

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{t=1}^{T} \int p(\mathbf{y}_t^{\rm val}|\boldsymbol{\theta}_t;\mathbf{X}_t^{\rm val})p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})d\boldsymbol{\theta}_t$$
$$\text{s.to}\quad p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta}) \propto p(\mathbf{y}_t^{\rm tr}|\boldsymbol{\theta}_t;\mathbf{X}_t^{\rm tr})p(\boldsymbol{\theta}_t;\boldsymbol{\theta}),\ \forall t \quad (4)$$

where we used that datasets are independent across tasks, and Bayes' rule in the second line. Through the posterior $p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})$, Bayesian meta-learning quantifies the uncertainty of task-specific parameter estimate $\hat{\boldsymbol{\theta}}_t$, thus assessing model robustness. When the posterior of $\boldsymbol{\theta}_t$ is replaced by its maximum a posteriori point estimator $\hat{\boldsymbol{\theta}}_t^{\rm map}$, meaning $p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta}) = \delta_D[\boldsymbol{\theta}_t - \hat{\boldsymbol{\theta}}_t^{\rm map}]$ with δ_D denoting Dirac's delta, it turns out that (4) reduces to (1).

Unfortunately, the posterior in (4) can be intractable with nonlinear models due to the difficulty of finding analytical solutions. To overcome this, we can resort to the widely adopted approximate variational inference (VI); see e.g. (Finn, Xu, and Levine 2018; Ravi and Beatson 2019; Nguyen, Do, and Carneiro 2020). VI searches over a family of tractable distributions for a surrogate that best matches the true posterior $p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})$. This can be accomplished by minimizing the KL-divergence between the surrogate pdf $q(\boldsymbol{\theta}_t;\mathbf{v}_t)$ and the true one, where $\mathbf{v}_t$ determines the variational distribution. Considering that the dimension of $\boldsymbol{\theta}_t$ can be fairly high, both the prior and surrogate posterior are often set to be Gaussian ($\mathcal{N}$) with diagonal covariance matrices. Specifically, we select the prior as $p(\boldsymbol{\theta}_t;\boldsymbol{\theta}) = \mathcal{N}(\mathbf{m},\mathbf{D})$ with covariance $\mathbf{D} = \text{diag}(\mathbf{d})$ and $\boldsymbol{\theta} := [\mathbf{m}^\top,\mathbf{d}^\top]^\top \in \mathbb{R}^d \times \mathbb{R}^d_{>0}$, and the surrogate posterior as $q(\boldsymbol{\theta}_t;\mathbf{v}_t) = \mathcal{N}(\mathbf{m}_t,\mathbf{D}_t)$ with $\mathbf{D}_t = \text{diag}(\mathbf{d}_t)$ and $\mathbf{v}_t := [\mathbf{m}_t^\top,\mathbf{d}_t^\top]^\top \in \mathbb{R}^d \times \mathbb{R}^d_{>0}$.

To ensure tractable numerical integration over $q(\boldsymbol{\theta}_t;\mathbf{v}_t)$, the meta-learning loss is often relaxed to an upper bound of $\sum_{t=1}^T -\log p(\mathbf{y}_t^{\rm val}|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm val},\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})$. Common choices include applying Jensen's inequality (Nguyen, Do, and Carneiro 2020) or an extra VI (Finn, Xu, and Levine 2018; Ravi and Beatson 2019) on (4). For notational convenience, here we will denote this upper bound by $\mathcal{L}_t^{\rm val}(\mathbf{v}_t,\boldsymbol{\theta})$. With VI and a relaxed (upper bound) objective, (4) becomes

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathcal{L}_t^{\rm val}(\mathbf{v}_t^*(\boldsymbol{\theta}),\boldsymbol{\theta}) \quad (5)$$
$$\text{s.to}\ \mathbf{v}_t^*(\boldsymbol{\theta}) = \operatorname*{argmin}_{\mathbf{v}_t} \text{KL}\left(q(\boldsymbol{\theta}_t;\mathbf{v}_t)\big\|p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})\right)\ \forall t,$$

where $\mathcal{L}_t^{\rm val}$ depends on $\boldsymbol{\theta}$ in two ways: i) via the intermediate variable $\mathbf{v}_t^*$; and, ii) by acting directly on $\mathcal{L}_t^{\rm val}$. Note that (5) is general enough to cover the case where $\mathcal{L}_t^{\rm val}$ is constructed using both $\mathcal{D}_t^{\rm val}$ and $\mathcal{D}_t^{\rm tr}$; see e.g., (Ravi and Beatson 2019). Similar to optimization-based meta-learning, the difficulty in reaching global optima prompts one to substitute $\mathbf{v}_t^*$ with a sub-optimum $\hat{\mathbf{v}}_t$ obtained through an algorithm $\hat{\mathcal{A}}_t(\boldsymbol{\theta})$; i.e.,

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t(\boldsymbol{\theta}),\boldsymbol{\theta})$$
$$\text{s.to}\quad \hat{\mathbf{v}}_t(\boldsymbol{\theta}) = \hat{\mathcal{A}}_t(\boldsymbol{\theta}),\quad t=1,\ldots,T. \quad (6)$$

2.2 Scalability issues in meta-learning

Delay and memory resources required for solving (2) and (6) are arguably the major challenges that meta-learning faces. Here we will elaborate on these challenges in the optimization-based setup, but the same argument carries over to Bayesian meta-learning too.

Consider minimizing the meta-learning loss in (2) using gradient-based iteration such as Adam (Kingma and Ba 2015). In the $(r+1)$-st iteration, gradients must be computed for a batch $\mathcal{B}^r \subset \{1,\ldots,T\}$ of tasks. Letting $\hat{\boldsymbol{\theta}}_t^r := \hat{\mathcal{A}}_t(\hat{\boldsymbol{\theta}}^r)$, where $\hat{\boldsymbol{\theta}}^r$ denotes the meta-parameter in the r-th iteration, the chain rule yields the so-termed meta-gradient

$$\nabla_{\boldsymbol{\theta}}\check{\mathcal{L}}_t^{\rm val}(\hat{\boldsymbol{\theta}}_t^r(\boldsymbol{\theta}))\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^r} = \nabla\hat{\mathcal{A}}_t(\hat{\boldsymbol{\theta}}^r)\nabla\check{\mathcal{L}}_t^{\rm val}(\hat{\boldsymbol{\theta}}_t^r),\ \ t\in\mathcal{B}^r \quad (7)$$

where $\nabla\hat{\mathcal{A}}_t(\hat{\boldsymbol{\theta}}^r)$ contains high-order derivatives. When $\hat{\mathcal{A}}_t$ is chosen as the one-step GD (cf. (3)), the meta-gradient is

$$\nabla\hat{\mathcal{A}}_t(\hat{\boldsymbol{\theta}}^r) = \mathbf{I}_d - \alpha\nabla^2\check{\mathcal{L}}_t^{\rm tr}(\hat{\boldsymbol{\theta}}^r),\quad t\in\mathcal{B}^r. \quad (8)$$

Fortunately, in this case the meta-gradient can still be computed through the Hessian-vector product (HVP), which incurs spatio-temporal complexity $\mathcal{O}(d)$.

In general, $\hat{\mathcal{A}}_t$ is a K-step GD for some $K>1$, which gives rise to high-order derivatives $\{\nabla^k\check{\mathcal{L}}_t^{\rm tr}(\hat{\boldsymbol{\theta}}^r)\}_{k=2}^{K+1}$ in the meta-gradient. The most efficient computation of the meta-gradient calls for recursive application of HVP K times, what incurs an overall complexity of $\mathcal{O}(Kd)$ in time, and $\mathcal{O}(Kd)$ in space requirements. Empirical wisdom however, favors a large K because it leads to improved accuracy in approximating the true meta-gradient $\nabla_{\boldsymbol{\theta}}\check{\mathcal{L}}_t^{\rm val}(\mathcal{A}_t^*(\boldsymbol{\theta}))\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^r}$. Hence, the linear increase of complexity with K will impede the scaling of optimization-based meta-learning algorithms.

When computing the meta-gradient, it should be underscored that the forward implementation of the K-step GD function has complexity $\mathcal{O}(Kd)$. However, the constant hidden in the $\mathcal{O}$ is much smaller compared to the HVP computation in the backward propagation. Typically, the constant is $1/5$ in terms of time and $1/2$ in terms of space; see (Griewank 1993; Rajeswaran et al. 2019). For this reason, we will focus on more efficient means of obtaining the meta-gradient function $\nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\rm val}(\hat{\mathcal{A}}_t(\boldsymbol{\theta}))$ for Bayesian meta-learning. It is also worth stressing that our results in the next section will hold for an arbitrary vector $\boldsymbol{\theta}\in\mathbb{R}^d\times\mathbb{R}^d_{>0}$ instead of solely the variable $\hat{\boldsymbol{\theta}}^r$ of the r-th iteration. Thus, we will use the general vector $\boldsymbol{\theta}$ when introducing our approach, while we will take its value at the point $\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^r$ when presenting our meta-learning algorithm.

3 Implicit Bayesian meta-learning

In this section, we will first introduce the proposed implicit Bayesian meta-learning (iBaML) method, which is built on top of implicit differentiation. Then, we will provide theoretical analysis to bound and compare the errors of explicit and implicit differentiation.

3.1 Implicit Bayesian meta-gradients

We start with decomposing the meta-gradient in Bayesian meta-learning (6) (henceforth referred to as Bayesian meta-gradient) using the chain rule

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t(\boldsymbol{\theta}),\boldsymbol{\theta}) = \nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t,\boldsymbol{\theta}) + \nabla_2\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t,\boldsymbol{\theta}),\ \ t=1,\ldots,T \quad (9)$$

where ∇_1 and ∇_2 denote the partial derivatives of a function w.r.t. its first and second arguments, respectively. The computational burden in (9) comes from the high-order derivatives present in the Jacobian $\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})$.

The key idea behind implicit differentiation is to express $\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})$ as a function of itself, so that it can be numerically obtained without using high-order derivatives. The following lemma formalizes how the implicit Jacobian is obtained in our setup. All proofs can be found in the Appendix.

Lemma 1. *Consider the Bayesian meta-learning problem in* (5)*, and let* $\bar{\mathbf{v}}_t := [\bar{\mathbf{m}}_t^\top, \bar{\mathbf{d}}_t^\top]^\top$ *be a local minimum of the task-level KL-divergence generated by* $\bar{\mathcal{A}}_t(\boldsymbol{\theta})$*. Also, let* $\mathcal{L}_t^{\rm tr}(\mathbf{v}_t) := \mathbb{E}_{q(\boldsymbol{\theta}_t;\mathbf{v}_t)}[-\log p(\mathbf{y}_t^{\rm tr}|\boldsymbol{\theta}_t;\mathbf{X}_t^{\rm tr})]$ *denote the expected negative log-likelihood (nll) on* $\mathcal{D}_t^{\rm tr}$*. If* $\mathbf{H}_t(\bar{\mathbf{v}}_t) := \nabla^2\mathcal{L}_t^{\rm tr}(\bar{\mathbf{v}}_t) + \begin{bmatrix}\mathbf{D}^{-1} & \mathbf{0}_d\\ \mathbf{0}_d & \frac{1}{2}\big(\mathbf{D}^{-1}+2\,\mathrm{diag}\,\big(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\rm tr}(\bar{\mathbf{v}}_t)\big)\big)^2\end{bmatrix}$ *is invertible, then it holds for* $\forall t\in\{1,\ldots,T\}$ *that*

$$\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta}) = \begin{bmatrix}\mathbf{D}^{-1} & \mathbf{0}_d\\ -\mathrm{diag}\,\big(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\rm tr}(\bar{\mathbf{v}}_t)\big)\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2}\end{bmatrix}\mathbf{H}_t^{-1}(\bar{\mathbf{v}}_t). \quad (10)$$

Two remarks are now in order regarding the technical assumption, and connections with iMAML. For notational brevity, define the block matrix

$$\mathbf{G}_t(\bar{\mathbf{v}}_t) := \begin{bmatrix}\mathbf{D}^{-1} & \mathbf{0}_d\\ -\mathrm{diag}\,\big(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\rm tr}(\bar{\mathbf{v}}_t)\big)\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2}\end{bmatrix}. \quad (11)$$

Remark 1. The invertibility of $\mathbf{H}_t(\bar{\mathbf{v}}_t)$ in Lemma 1 is assumed to ensure uniqueness of $\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})$. Without this assumption, it turns out that $\bar{\mathbf{v}}_t$ can be a singular point, belonging to a subspace where any point is also a local minimum. The Bayesian meta-gradients (9) of the points in this subspace form a set

$$\bar{\mathcal{G}}_t = \Big\{\mathbf{G}_t(\bar{\mathbf{v}}_t)\big(\mathbf{H}_t^\dagger(\bar{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t,\boldsymbol{\theta})+\mathbf{u}\big) + \nabla_2\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t,\boldsymbol{\theta}) \,\Big|\, \forall\mathbf{u}\in\mathrm{Null}\big(\mathbf{H}_t(\bar{\mathbf{v}}_t)\big)\Big\} \quad (12)$$

where † represents pseudo-inverse, and $\mathrm{Null}(\cdot)$ stands for the null space. Upon replacing $\mathbf{H}_t^{-1}(\bar{\mathbf{v}}_t)$ with $\mathbf{H}_t^\dagger(\bar{\mathbf{v}}_t)$, one can generalize Lemma 1, and forgo the invertibility assumption.

Algorithm 1: Implicit Bayesian meta-learning (iBaML)

Inputs: tasks $\{1,\ldots,T\}$ with their $\mathcal{D}_t^{\rm tr}$ and $\mathcal{D}_t^{\rm val}$, and meta-learning rate β.
Initialization: initialize $\hat{\boldsymbol{\theta}}^0$ randomly, and iteration counter $r=0$.
repeat
  Sample a batch $\mathcal{B}^r\subset\{1,\ldots,T\}$ of tasks;
  for $t\in\mathcal{B}^r$ **do**
    Compute task-level sub-optimum $\hat{\mathbf{v}}_t^r = \hat{\mathcal{A}}_t(\hat{\boldsymbol{\theta}}^r)$ using e.g. K-step GD;
    Approximate $\hat{\mathbf{u}}_t^r \approx \mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t^r)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t^r,\hat{\boldsymbol{\theta}}^r)$ with L-step CG;
    Compute meta-level gradient $\hat{\mathbf{g}}_t^r = \mathbf{G}_t(\hat{\mathbf{v}}_t^r)\hat{\mathbf{u}}_t^r + \nabla_2\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t^r,\hat{\boldsymbol{\theta}}^r)$ using (17);
  end for
  Update $\hat{\boldsymbol{\theta}}^{r+1} = \hat{\boldsymbol{\theta}}^r - \beta\frac{1}{|\mathcal{B}^r|}\sum_{t\in\mathcal{B}^r}\hat{\mathbf{g}}_t^r$;
  $r=r+1$;
until convergence
Output: $\hat{\boldsymbol{\theta}}^r$.

Remark 2. To recognize how Lemma 1 links iBaML with iMAML (Rajeswaran et al. 2019), consider the special case where the covariance matrices of the prior and local minimum are fixed as $\mathbf{D}\equiv\lambda^{-1}\mathbf{I}_d$ and $\bar{\mathbf{D}}_t\equiv\mathbf{0}_d$ for some constant λ. Since $\mathbf{d}=[\lambda^{-1},\ldots,\lambda^{-1}]\in\mathbb{R}^d$ is a constant vector, Lemma 1 boils down to

$$\nabla_{\mathbf{m}}\bar{\mathcal{A}}_t(\boldsymbol{\theta}) = \mathbf{D}^{-1}\big(\nabla_{\mathbf{m}}^2\mathcal{L}_t^{\rm tr}(\bar{\mathbf{v}}_t)+\mathbf{D}^{-1}\big)^{-1} = \big(\lambda^{-1}\nabla_{\mathbf{m}}^2\mathcal{L}_t^{\rm tr}(\bar{\mathbf{v}}_t)+\mathbf{I}_d\big)^{-1} \quad (13)$$

which coincides with Lemma 1 of (Rajeswaran et al. 2019). Hence, iBaML subsumes iMAML whose expressiveness is confined because $\mathbf{d}$ is fixed, while iBaML entails a learnable covariance matrix in the prior $p(\boldsymbol{\theta}_t;\boldsymbol{\theta})$. In addition, the uncertainty of iMAML's training posterior $p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\rm tr};\mathbf{X}_t^{\rm tr},\boldsymbol{\theta})$ can be more challenging to quantify than that in iBaML.

An immediate consequence of Lemma 1 is the so-called generalized implicit gradients. Suppose that $\hat{\mathcal{A}}_t$ involves a K sufficiently large for the sub-optimal point $\hat{\mathbf{v}}_t$ to be close to a local optimum $\bar{\mathbf{v}}_t$. The Bayesian meta-gradient (9) can then be approximated through

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t(\boldsymbol{\theta}),\boldsymbol{\theta}) \quad (14)$$
$$\approx \mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t,\boldsymbol{\theta}) + \nabla_2\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t,\boldsymbol{\theta}),\ \forall t.$$

The approximate implicit gradient in (14) is computationally expensive due to the matrix inversion $\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)$, which incurs complexity $\mathcal{O}(d^3)$. To relieve the computational burden, a key observation is that $\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t,\boldsymbol{\theta})$ is the solution of the optimization problem

$$\operatorname*{argmin}_{\mathbf{u}}\ \frac{1}{2}\mathbf{u}^\top\mathbf{H}_t(\hat{\mathbf{v}}_t)\mathbf{u} - \mathbf{u}^\top\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t,\boldsymbol{\theta}). \quad (15)$$

Given that the square matrix $\mathbf{H}_t(\hat{\mathbf{v}}_t)$ is by definition symmetric, problem (15) can be efficiently solved using the conjugate gradient (CG) iteration. Specifically, the complexity

of CG is dominated by the matrix-vector product $\mathbf{H}_t(\hat{\mathbf{v}}_t)\mathbf{p}$ (for some vector $\mathbf{p} \in \mathbb{R}^{2d}$), given by

$$\mathbf{H}_t(\hat{\mathbf{v}}_t)\mathbf{p} = \nabla^2 \mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\mathbf{p} + \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d & \frac{1}{2}\big(\mathbf{D}^{-1} + 2\,\mathrm{diag}\left(\nabla_{\hat{\mathbf{d}}_t} \mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)\big)^2 \end{bmatrix} \mathbf{p}. \tag{16}$$

The first term on the right-hand side of (16) is an HVP, and the second is the multiplication of a diagonal matrix with a vector. Note that with the diagonal matrix, the latter term boils down to a dot product, implying that the complexity of each CG iteration is as low as $\mathcal{O}(d)$. In practice, a small number of CG iterations suffices to produce an accurate estimate of $\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1 \mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta})$ thanks to its fast convergence rate (Van der Sluis and van der Vorst 1986; Winther 1980). In order to control the total complexity of iBaML, we set the maximum number of CG iterations to a constant L.

Having obtained an approximation of the matrix-inverse-vector product $\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1 \mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta})$, we proceed to estimate the Bayesian meta-gradient. Let $\hat{\mathbf{u}}_t := [\hat{\mathbf{u}}_{t,\mathbf{m}}^\top, \hat{\mathbf{u}}_{t,\mathbf{d}}^\top]^\top$ be the output of the CG method with subvectors $\hat{\mathbf{u}}_{t,\mathbf{m}},\ \hat{\mathbf{u}}_{t,\mathbf{d}} \in \mathbb{R}^d$. Then, it follows from (14) that

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}) \\ &\approx \mathbf{G}_t(\hat{\mathbf{v}}_t)\hat{\mathbf{u}}_t + \nabla_2 \mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) \\ &= \begin{bmatrix} \mathbf{D}^{-1}\hat{\mathbf{u}}_{t,\mathbf{m}} \\ -\,\mathrm{diag}\left(\nabla_{\hat{\mathbf{m}}_t} \mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)\mathbf{D}^{-1}\hat{\mathbf{u}}_{t,\mathbf{m}} + \frac{1}{2}\mathbf{D}^{-2}\hat{\mathbf{u}}_{t,\mathbf{d}} \end{bmatrix} \\ &\quad + \nabla_2 \mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) := \hat{\mathbf{g}}_t, \quad t = 1, \ldots, T \end{aligned}$$

where we also used the definition (11). Again, the diagonal-matrix-vector products in (17) can be efficiently computed through dot products, which incur complexity $\mathcal{O}(d)$. The step-by-step pseudocode of the iBaML is listed under Algorithm 1.

In a nutshell, the implicit Bayesian meta-gradient computation consumes $\mathcal{O}(Ld)$ time, regardless of the optimization algorithm $\hat{\mathcal{A}}_t$. One can even employ more complicated algorithms such as second-order matrix-free optimization (Martens and Grosse 2015; Botev, Ritter, and Barber 2017). In addition, as the time complexity does not depend on K, one can increase K to reduce the approximation error in (14). The space complexity of iBaML is only $\mathcal{O}(d)$ thanks to the iterative implementation of CG steps. These considerations explain how iBaML addresses the scalability issue of explicit backpropagation.

3.2 Theoretical analysis

This section deals with performance analysis of both explicit and implicit gradients in Bayesian meta-learning to further understand their differences. Similar to (Rajeswaran et al. 2019), our results will rely on the following assumptions.

Assumption 1. *Vector $\bar{\mathbf{v}}_t = \bar{\mathcal{A}}_t(\boldsymbol{\theta})$ is a local minimum of the KL-divergence in* (5).

Assumption 2. *The meta-loss function $\mathcal{L}_t^{\mathrm{val}}(\mathbf{v}_t, \boldsymbol{\theta})$ is A_t-Lipschitz and B_t-smooth w.r.t. $\mathbf{v}_t$ while its partial gradient $\nabla_2 \mathcal{L}_t^{\mathrm{val}}(\mathbf{v}_t, \boldsymbol{\theta})$ is C_t-Lipschitz w.r.t. $\mathbf{v}_t$.*

Assumption 3. *The expected nll function $\mathcal{L}_t^{\mathrm{tr}}(\mathbf{v}_t)$ is D_t-smooth, and has a Hessian that is E_t-Lipschitz.*

Assumption 4. *Matrices $\mathbf{H}_t(\hat{\mathbf{v}}_t)$ and $\mathbf{H}_t(\bar{\mathbf{v}}_t)$ are both non-singular; that is, their smallest singular value $\sigma_t := \min\left\{\sigma_{\min}\big(\mathbf{H}_t(\hat{\mathbf{v}}_t)\big), \sigma_{\min}\big(\mathbf{H}_t(\bar{\mathbf{v}}_t)\big)\right\} > 0$.*

Assumption 5. *Prior variances are positive and bounded, meaning $0 < D_{\min} \leq [\mathbf{d}]_i \leq D_{\max},\ i = 1, \ldots, d$.*

Based on these assumptions, we can establish the following result.

Theorem 1 (Explicit Bayesian meta-gradient error bound)**.** *Consider the Bayesian meta-learning problem* (6)*. Let $\epsilon_t := \|\hat{\mathbf{v}}_t - \bar{\mathbf{v}}_t\|_2$ be the task-level optimization error, and $\delta_t := \|\nabla \hat{\mathcal{A}}_t(\boldsymbol{\theta}) - \mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\|_2$ the error in the Jacobian. Upon defining $\rho_t := \max\left\{\|\nabla_{\bar{\mathbf{v}}_t} \mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\|_\infty, \|\nabla_{\hat{\mathbf{v}}_t} \mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\|_\infty\right\}$, and with Assumptions 1-5 in effect, it holds for $t \in \{1, \ldots, T\}$ that*

$$\left\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathrm{val}}\big(\hat{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathrm{val}}\big(\bar{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big)\right\|_2 \leq F_t \epsilon_t + A_t \delta_t \tag{17}$$

where F_t is a constant dependent on ρ_t.

Theorem 1 asserts that the ℓ_2 error of the explicit Bayesian meta-gradient relative to the true depends on the task-level optimization error as well as the error in the Jacobian, where the former captures the Euclidean distance of the local minimum $\bar{\mathbf{v}}_t$ and its approximation $\hat{\mathbf{v}}_t$, while the latter characterizes how the sub-optimal function $\hat{\mathcal{A}}_t$ influences the Jacobian. Both errors can be reduced by increasing K in the task-level optimization, at the cost of time and space complexity for backpropagating $\nabla \hat{\mathcal{A}}_t(\boldsymbol{\theta})$. Ideally, one can have $\delta_t = 0$ when $\hat{\mathbf{v}}_t$ is a local optimum, and $\epsilon_t = 0$ when choosing $\bar{\mathbf{v}}_t = \hat{\mathbf{v}}_t$.

Next, we derive an error bound for implicit differentiation.

Theorem 2 (Implicit Bayesian meta-gradient error bound)**.** *Consider the Bayesian meta-learning problem* (6)*. Let $\epsilon_t := \|\hat{\mathbf{v}}_t - \bar{\mathbf{v}}_t\|_2$ be the task-level optimization error, and $\delta_t' := \|\hat{\mathbf{u}}_t - \mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1 \mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta})\|$ the CG error. Upon defining $\rho_t := \max\left\{\|\nabla_{\bar{\mathbf{v}}_t} \mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\|_\infty, \|\nabla_{\hat{\mathbf{v}}_t} \mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\|_\infty\right\}$, and with Assumptions 1-5 in effect, it holds for $t \in \{1, \ldots, T\}$ that*

$$\left\|\hat{\mathbf{g}}_t - \nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathrm{val}}\big(\bar{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big)\right\|_2 \leq F_t' \epsilon_t + G_t' \delta_t', \tag{18}$$

where F_t' and G_t' are constants dependent on ρ_t.

While the bound on implicit meta-gradient also depends on the task-level optimization error, the difference with Theorem 1 is highlighted in the CG error. The fast convergence of CG leads to a tolerable δ_t' even with a small L. As a result, one can opt for a large K to reduce task-level optimization error ϵ_t, and a small L to obtain a satisfactory approximation of the meta-gradient.

It is worth stressing that $\bar{\mathbf{v}}_t$ in Theorems 1 and 2 can denote *any* local optimum. It further follows by definition that both δ_t and δ_t' do not rely on the choice of local optima, yet ϵ_t does. One final remark is now in order.

Remark 3. Theorems 1 and 2 can be further simplified under the additional assumption that $\mathcal{L}_t^{\mathrm{tr}}(\mathbf{v}_t)$ is H_t-Lipschitz. In such a case, we have $\rho_t \leq H_t$, and thus the scalars F_t, F_t' and G_t' boil down to task-specific constants.

4 Numerical tests

Here we test and showcase on synthetic and real data the analytical novelties of this contribution. Our implementation relies on the PyTorch (Paszke et al. 2019), and experiments are run on a server equipped with an Intel Core i7-6700K CPU (4.00GHz), and an NVIDIA TITAN X GPU.

4.1 Synthetic data

Here we experiment on the errors between explicit and implicit gradients over a synthetic dataset. The data are generated using the Bayesian linear regression model

$$y_t^n = \langle \boldsymbol{\theta}_t, \mathbf{x}_t^n \rangle + e_t^n, \ \forall n, \quad t = 1, \ldots, T \tag{19}$$

where $\{\boldsymbol{\theta}_t\}_{t=1}^T$ are i.i.d. samples drawn from a distribution $p(\boldsymbol{\theta}_t; \hat{\boldsymbol{\theta}})$ that is unknown during meta-training, and e_t^n is the additive white Gaussian noise (AWGN) with known variance σ^2. Although the current training posterior $p(\boldsymbol{\theta}_t | y_t^{\mathrm{tr}}; \mathbf{X}_t^{\mathrm{tr}}, \boldsymbol{\theta})$ becomes tractable, we still focus on the VI approximation for uniformity. Within this rudimentary linear case, it can be readily verified that the task-level optimum $\mathbf{v}_t^* := [\mathbf{m}_t^{*\top}, \mathbf{d}_t^{*\top}]^\top$ of (5) is given by

$$\mathbf{m}_t^* = \Big(\frac{1}{\sigma^2} \mathbf{X}_t^{\mathrm{tr}} (\mathbf{X}_t^{\mathrm{tr}})^\top + \mathbf{D}^{-1} \Big)^{-1} \big(\mathbf{D}^{-1} \mathbf{m} + \frac{1}{\sigma^2} \mathbf{X}_t^{\mathrm{tr}} \mathbf{y}_t^{\mathrm{tr}} \big) \tag{20a}$$

$$\mathbf{d}_t^* = \Big(\frac{1}{2\sigma^2} \operatorname{diag} \big(\mathbf{X}_t^{\mathrm{tr}} (\mathbf{X}_t^{\mathrm{tr}})^\top \big) + \mathbf{d}^{-1} \Big)^{-1}, \ \ t = 1 \ldots, T \tag{20b}$$

where $\operatorname{diag}(\mathbf{M})$ is a vector collecting the diagonal entries of matrix $\mathbf{M}$. The true posterior in the linear case is $p(\boldsymbol{\theta}_t | \mathbf{y}_t^{\mathrm{tr}}; \mathbf{X}_t^{\mathrm{tr}}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{m}_t^*, \big(\frac{1}{2\sigma^2} (\mathbf{X}_t^{\mathrm{tr}} (\mathbf{X}_t^{\mathrm{tr}})^\top) + \mathbf{d}^{-1} \big)^{-1})$, implying that the posterior covariance matrix is essentially approximated by its diagonal counterpart $\mathbf{D}_t^*$ in VI. Lemma 1 and (9) imply that the oracle meta-gradient is

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathrm{val}}(\mathbf{v}_t^*(\boldsymbol{\theta}), \boldsymbol{\theta}) \\ &= \mathbf{G}_t(\mathbf{v}_t^*) \mathbf{H}_t^{-1}(\mathbf{v}_t^*) \nabla_1 \mathcal{L}_t^{\mathrm{val}}(\mathbf{v}_t^*, \boldsymbol{\theta}) + \nabla_2 \mathcal{L}_t^{\mathrm{val}}(\mathbf{v}_t^*, \boldsymbol{\theta}), \ \forall t. \end{aligned} \tag{21}$$

As a benchmark meta-learning algorithm, we selected the amortized Bayesian meta-learning (ABML) in (Ravi and Beatson 2019). The metric used for performance assessment is the normalized root-mean-square error (NRMSE) between the true meta-gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathrm{val}}(\mathbf{v}_t^*(\boldsymbol{\theta}), \boldsymbol{\theta})$, and the estimated meta-gradients $\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta})$ and $\hat{\mathbf{g}}_t$; see also the Appendix for additional details on the numerical test.

Figure 1 depicts the NRMSE as a function of K for the first iteration of ABML, that is at the point $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^0$. For explicit and implicit gradients, the NRMSE decreases as K increases, while the former outperforms the latter for $K \leq 5$, and the vice-versa for $K > 5$. These observations confirm our analytical results. Intuitively, factors $F_t \epsilon_t$ and $F_t' \epsilon_t$ caused by imprecise task-level optimization dominate the upper bounds for small K, thus resulting in large NRMSE. Besides, implicit gradients are more sensitive to task-level optimization errors. One conjecture is that iBaML is developed based on Lemma 1, where the matrix inversion can be sensitive to $\bar{\mathbf{v}}_t$'s variation. Despite that the conditioning

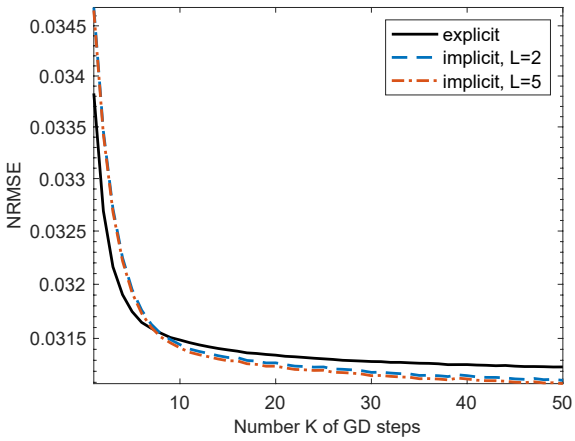


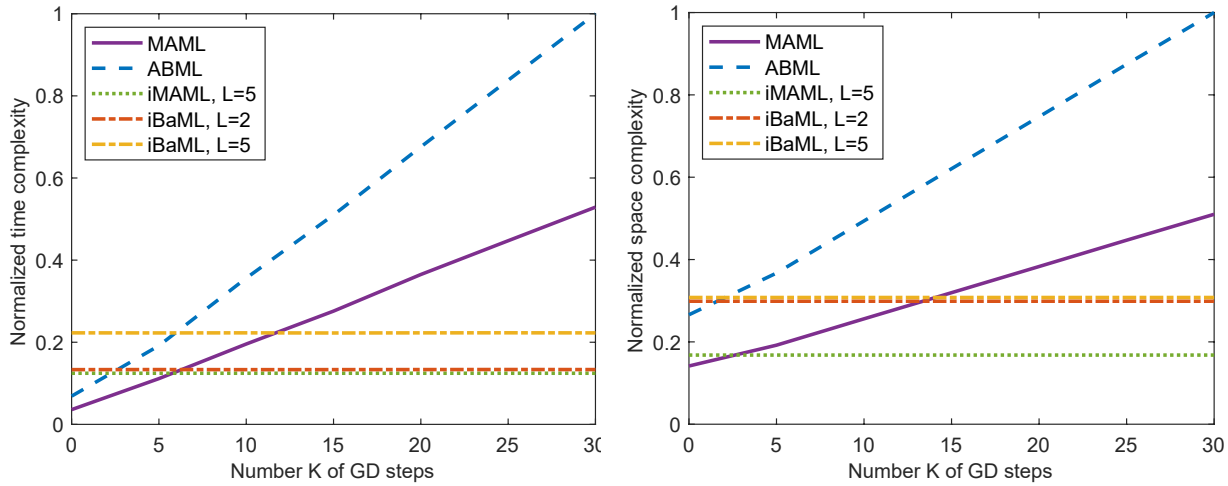
Figure 1: Gradient error comparison on synthetic dataset.

number κ of $\mathbf{X}_t^{\mathrm{tr}}$ takes on a large value purposely so that ϵ_t decreases slowly with K, a small K suffices to capture accurately implicit gradients. The main reason is that the CG error δ_t' can become sufficiently small even with only $L = 2$ steps, while δ_t remains large because GD converges slowly.

4.2 Real data

Next, we conduct tests to assess the performance of iBaML on real datasets. We consider one of the most widely used few-shot dataset for classification *mini*ImageNet (Vinyals et al. 2016). This dataset consists of natural images categorized in 100 classes, with 600 samples per class. All images are cropped to have size of 84×84. We adopt the dataset splitting suggested by (Ravi and Larochelle 2017), where 64, 16 and 20 disjoint classes are used for meta-training, meta-validation and meta-testing, respectively. The setups of the numerical test follow from the standard W-class S^{tr}-shot few-shot learning protocol in (Vinyals et al. 2016). In particular, each task has W randomly selected classes, and each class contains S^{tr} training images and S^{val} validation images. In other words, we have $N^{\mathrm{tr}} = S^{\mathrm{tr}} W$ and $N^{\mathrm{val}} = S^{\mathrm{val}} W$. We further adopt the typical choices with $W = 5$, $S^{\mathrm{tr}} \in \{1, 5\}$, and $S^{\mathrm{val}} = 15$. It should be noted that the training and validation sets are also known as support and query sets in the context of few-shot learning.

We first empirically compare the computational complexity (time and space) for explicit versus implicit gradients on the 5-class 1-shot *mini*ImageNet dataset. Here we are only interested in backward complexity, so the delay and memory requirements for forward pass of $\hat{\mathcal{A}}_t$ is excluded. Figure 2(a) plots the time complexity of explicit and implicit gradients against K. It is observed that the time complexity of explicit gradient grows linearly with K, while the implicit one increases only with L but not K. Moreover, the explicit and implicit gradients have comparable time complexity when $K = L$. As far as space complexity, Figure 2(b) illustrates that memory usage with explicit gradients is proportional to K. In contrast, the memory used in the implicit gradient algorithms is nearly invariant across K values. Such a memory-saving property is important when meta-learning is employed with models of growing degrees of freedom.



(a) Time complexity (b) Space complexity

Figure 2: Time and space complexity comparisons for meta-gradients computation on 5-class 1-shot *mini*ImageNet dataset.

Method	nll	accuracy
MAML, $K = 5$	0.967 ± 0.017	$63.1 \pm 0.92\%$
ABML, $K = 5$	0.957 ± 0.016	$62.8 \pm 0.74\%$
iBaML, $K = 5$	0.965 ± 0.018	$63.2 \pm 0.74\%$
iBaML, $K = 10$	0.947 ± 0.017	$64.0 \pm 0.75\%$
iBaML, $K = 15$	0.943 ± 0.017	$64.0 \pm 0.74\%$

Table 1: Test negative log-likelihood (nll) and accuracy comparison on 5-class 5-shot *mini*ImageNet dataset. The $\pm$ sign indicates the 95% confidence interval.

Furthermore, one may also notice from both figures that MAML and iMAML incur about 50% time/space complexities of ABML and iBaML. This is because non-Bayesian approaches only optimize the mean vector of the Gaussian prior, whose dimension is d, while the probabilistic methods cope with both the mean and diagonal covariance matrix of the pdf with corresponding dimension $2d$. This increase in dimensionality doubles the space-time complexity in gradient computations.

Next, we demonstrate the effectiveness of iBaML in reducing the Bayesian meta-learning loss. The test is conducted on the 5-class 5-shot *mini*ImageNet. The model is a standard 4-layer 32-channel convolutional neural network, and the chosen baseline algorithms are MAML (Finn, Abbeel, and Levine 2017) and ABML (Ravi and Beatson 2019); see also the Appendix for alternative setups. Due to the large number of training tasks, it is impractical to compute the exact meta-training loss. As an alternative, we adopt the 'test nll' (averaged over $1,000$ test tasks) as our metric, and also report their corresponding accuracy. For fairness, we set $L = 5$ when implementing the implicit gradients so that the time complexity is similar to explicit one with $K = 5$. The results are listed in Table 1. It is observed that both nll and accuracy improve with K, implying that the meta-learning loss can be effectively reduced by trading a small error in gradient estimation.

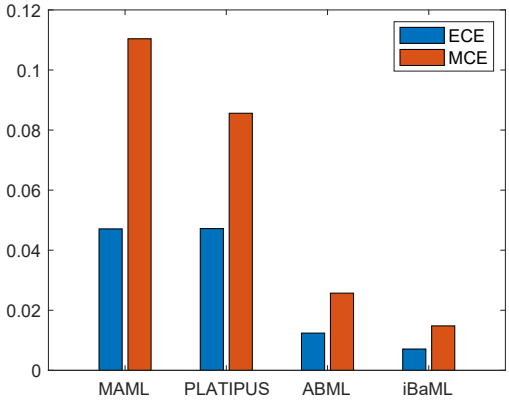


Figure 3: Calibration errors on 5-class 1-shot *mini*ImageNet.

To quantify the uncertainties embedded in state-of-the-art meta-learning methods, Figure 3 plots the expected/maximum calibration errors (ECE/MCE) (Naeini, Cooper, and Hauskrecht 2015). It can be seen that iBaML is once again the most competitive among tested approaches.

5 Conclusions

This paper develops a novel so-termed iBaML approach to enhance the scalablity of Bayesian meta-learning. At the core of iBaML is an estimate of meta-gradients using implicit differentiation. Analysis reveals that the estimation error is upper bounded by task-level optimization and CG errors, and these two can be significantly reduced with only a slight increase in time complexity. In addition, the required computational complexity is invariant to the task-level optimization trajectory, what allows iBaML to deal with complicated task-level optimization. Besides analytical performance, extensive numerical tests on synthetic and real datasets are also conducted and demonstrate the appealing merits of iBaML over competing alternatives.

Acknowledgments

This work was supported in part by NSF grants 2220292, 2212318, 2126052, and 2128593.

References

Abbas, M.; Xiao, Q.; Chen, L.; Chen, P.-Y.; and Chen, T. 2022. Sharp-MAML: Sharpness-Aware Model-Agnostic Meta Learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 10–32. PMLR.

Bengio, S.; Bengio, Y.; and Cloutier, J. 1995. On the Search for New Learning Rules for ANNs. *Neural Processing Letters*, 2(4): 26–30.

Bertinetto, L.; Henriques, J. F.; Torr, P.; and Vedaldi, A. 2019. Meta-learning with Differentiable Closed-Form Solvers. In *Proceedings of International Conference on Learning Representations*.

Botev, A.; Ritter, H.; and Barber, D. 2017. Practical Gauss-Newton Optimisation for Deep Learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 557–565. PMLR.

Chen, L.; and Chen, T. 2022. Is Bayesian Model-Agnostic Meta Learning Better than Model-Agnostic Meta Learning, Provably? In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, 1733–1774. PMLR.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. On the Convergence Theory of Gradient-Based Model-Agnostic Meta-Learning Algorithms. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, 1082–1092. PMLR.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 1126–1135. PMLR.

Finn, C.; Xu, K.; and Levine, S. 2018. Probabilistic Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Flennerhag, S.; Rusu, A. A.; Pascanu, R.; Visin, F.; Yin, H.; and Hadsell, R. 2020. Meta-Learning with Warped Gradient Descent. In *Proceedings of International Conference on Learning Representations*.

Franceschi, L.; Frasconi, P.; Salzo, S.; Grazzi, R.; and Pontil, M. 2018. Bilevel Programming for Hyperparameter Optimization and Meta-Learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 1568–1577. PMLR.

Grant, E.; Finn, C.; Levine, S.; Darrell, T.; and Griffiths, T. 2018. Recasting Gradient-Based Meta-Learning as Hierarchical Bayes. In *Proceedings of International Conference on Learning Representations*.

Griewank, A. 1993. Some bounds on the complexity of gradients, Jacobians, and Hessians. In *Complexity in numerical optimization*, 128–162. World Scientific.

Hansen, N.; and Wang, X. 2021. Generalization in Reinforcement Learning by Soft Data Augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 13611–13617.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of International Conference on Learning Representations*.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-Learning With Differentiable Convex Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Martens, J.; and Grosse, R. 2015. Optimizing Neural Networks with Kronecker-factored Approximate Curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 2408–2417. Lille, France: PMLR.

Miao, Y.; Metze, F.; and Rawat, S. 2013. Deep maxout networks for low-resource speech recognition. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 398–403. IEEE.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A Simple Neural Attentive Meta-Learner. In *International Conference on Learning Representations*.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty Ninth International Conference on Artificial Intelligence and Statistics*, 2901–2907. PMLR.

Nguyen, C.; Do, T.-T.; and Carneiro, G. 2020. Uncertainty in Model-Agnostic Meta-Learning using Variational Inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

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

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Rajeswaran, A.; Finn, C.; Kakade, S. M.; and Levine, S. 2019. Meta-Learning with Implicit Gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ravi, S.; and Beatson, A. 2019. Amortized Bayesian Meta-Learning. In *Proceedings of International Conference on Learning Representations*.

Ravi, S.; and Larochelle, H. 2017. Optimization as a Model for Few-Shot Learning. In *Proceedings of International Conference on Learning Representations*.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, 1842–1850. New York, New York, USA: PMLR.

Schmidhuber, J. 1993. A Neural Network that Embeds its Own Meta-Levels. In *IEEE International Conference on Neural Networks*, 407–412 vol.1.

Schmidhuber, J.; Zhao, J.; and Wiering, M. 1996. Simple Principles of Metalearning. *Technical report IDSIA*, 69: 1–23.

Thrun, S. 1998. *Lifelong Learning Algorithms*, 181–209. Boston, MA: Springer US. ISBN 978-1-4615-5529-2.

Thrun, S.; and Pratt, L. 2012. *Learning to Learn*. Springer Science & Business Media.

Van der Sluis, A.; and van der Vorst, H. A. 1986. The rate of convergence of conjugate gradients. *Numerische Mathematik*, 48(5): 543–560.

Vinyals, O.; Blundell, C.; Lillicrap, T.; kavukcuoglu, k.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Wang, H.; Sun, R.; and Li, B. 2020. Global Convergence and Generalization Bound of Gradient-Based Meta-Learning with Deep Neural Nets. *arXiv preprint arXiv:2006.14606*.

Winther, R. 1980. Some Superlinear Convergence Results for the Conjugate Gradient Method. *SIAM Journal on Numerical Analysis*, 17(1): 14–17.

yang, y.; Sun, J.; Li, H.; and Xu, Z. 2016. Deep ADMM-Net for Compressive Sensing MRI. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Yoon, J.; Kim, T.; Dia, O.; Kim, S.; Bengio, Y.; and Ahn, S. 2018. Bayesian Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Appendix

A.1 Proof of Lemma 1

Lemma 1 (Restated)**.** *Consider the Bayesian meta-learning problem* (5)*. Let* $\bar{\mathbf{v}}_t := [\bar{\mathbf{m}}_t^\top, \bar{\mathbf{d}}_t^\top]^\top$ *be a local minimum of the task-level KL-divergence generated by* $\bar{\mathcal{A}}_t(\boldsymbol{\theta})$*; and,* $\mathcal{L}_t^{\mathrm{tr}}(\mathbf{v}_t) := \mathbb{E}_{q(\boldsymbol{\theta}_t;\mathbf{v}_t)}[-\log p(\mathbf{y}_t^{\mathrm{tr}}|\boldsymbol{\theta}_t;\mathbf{X}_t^{\mathrm{tr}})]$ *the expected negative log-likelihood (nll) on* $\mathcal{D}_t^{\mathrm{tr}}$*. If* $\mathbf{H}_t(\bar{\mathbf{v}}_t) := \nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + \begin{bmatrix}\mathbf{D}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d & \frac{1}{2}\big(\mathbf{D}^{-1} + 2\,\mathrm{diag}\big(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\big)\big)^2\end{bmatrix}$ *is invertible, it then holds for* $t \in \{1, \ldots, T\}$ *that*

$$\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta}) = \begin{bmatrix}\mathbf{D}^{-1} & \mathbf{0}_d \\ -\,\mathrm{diag}(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t))\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2}\end{bmatrix}\mathbf{H}_t^{-1}(\bar{\mathbf{v}}_t). \tag{22}$$

Proof. We first write out the evidence lower bound (ELBO) of the VI in (4).

$$\mathrm{KL}\left(q(\boldsymbol{\theta}_t;\mathbf{v}_t)\middle\|p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})\right) = \int q(\boldsymbol{\theta}_t;\mathbf{v}_t)\log\frac{q(\boldsymbol{\theta}_t;\mathbf{v}_t)}{p(\boldsymbol{\theta}_t|\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})} = \int q(\boldsymbol{\theta}_t;\mathbf{v}_t)\log\frac{q(\boldsymbol{\theta}_t;\mathbf{v}_t)p(\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})}{p(\mathbf{y}_t^{\mathrm{tr}},\boldsymbol{\theta}_t;\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})}$$
$$= \int q(\boldsymbol{\theta}_t;\mathbf{v}_t)\log\frac{q(\boldsymbol{\theta}_t;\mathbf{v}_t)p(\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})}{p(\mathbf{y}_t^{\mathrm{tr}}|\boldsymbol{\theta}_t;\mathbf{X}_t^{\mathrm{tr}})p(\boldsymbol{\theta}_t;\boldsymbol{\theta})} = \mathbb{E}_{q(\boldsymbol{\theta}_t;\mathbf{v}_t)}[-\log p(\mathbf{y}_t^{\mathrm{tr}}|\boldsymbol{\theta}_t;\mathbf{X}_t^{\mathrm{tr}})] + \mathbb{E}_{q(\boldsymbol{\theta}_t;\mathbf{v}_t)}\Big[\log\frac{q(\boldsymbol{\theta}_t;\mathbf{v}_t)}{p(\boldsymbol{\theta}_t;\boldsymbol{\theta})}\Big]$$
$$+ \mathbb{E}_{q(\boldsymbol{\theta}_t;\mathbf{v}_t)}[\log p(\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})] = \mathcal{L}_t^{\mathrm{tr}}(\mathbf{v}_t) + \mathrm{KL}\left(q(\boldsymbol{\theta}_t;\mathbf{v}_t)\middle\|p(\boldsymbol{\theta}_t;\boldsymbol{\theta})\right) + \log p(\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta}) = -\mathrm{ELBO} + \log p(\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})$$

where $\mathrm{ELBO} := -\mathcal{L}_t^{\mathrm{tr}}(\mathbf{v}_t) - \mathrm{KL}\left(q(\boldsymbol{\theta}_t;\mathbf{v}_t)\middle\|p(\boldsymbol{\theta}_t;\boldsymbol{\theta})\right)$. Minimizing the KL divergence amounts to maximizing the ELBO.

From the definitions $\boldsymbol{\theta} := [\mathbf{m}^\top, \mathbf{d}^\top]^\top$ and $\bar{\mathbf{v}}_t := \bar{\mathcal{A}}_t(\boldsymbol{\theta}) = [\bar{\mathbf{m}}_t^\top, \bar{\mathbf{d}}_t^\top]^\top$, we can write the desired gradient as a block matrix

$$\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta}) = \begin{bmatrix}\nabla_{\mathbf{m}}\bar{\mathbf{m}}_t & \nabla_{\mathbf{m}}\bar{\mathbf{d}}_t \\ \nabla_{\mathbf{d}}\bar{\mathbf{m}}_t & \nabla_{\mathbf{d}}\bar{\mathbf{d}}_t\end{bmatrix} \tag{23}$$

where with a slight abuse in notation $\nabla_{\mathbf{m}}\bar{\mathcal{A}}_t(\boldsymbol{\theta}) = [\nabla_{\mathbf{m}}\bar{\mathbf{m}}_t, \nabla_{\mathbf{m}}\bar{\mathbf{d}}_t]$ and $\nabla_{\mathbf{d}}\bar{\mathcal{A}}_t(\boldsymbol{\theta}) = [\nabla_{\mathbf{d}}\bar{\mathbf{m}}_t, \nabla_{\mathbf{d}}\bar{\mathbf{d}}_t]$ denote partial gradients. The next step is to express $\nabla_{\mathbf{m}}\bar{\mathcal{A}}_t(\boldsymbol{\theta})$ as a function of itself to leverage the implicit differentiation.

Since $\bar{\mathbf{v}}_t$ is a local minimum of $\mathrm{KL}\left(q(\boldsymbol{\theta}_t;\mathbf{v}_t)\middle\|p(\boldsymbol{\theta}_t;\mathbf{y}_t^{\mathrm{tr}};\mathbf{X}_t^{\mathrm{tr}},\boldsymbol{\theta})\right)$, it maximizes the ELBO. The first-order necessary condition for optimality thus yields

$$-\nabla\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) - \nabla_{\bar{\mathbf{v}}_t}\,\mathrm{KL}\left(q(\boldsymbol{\theta}_t;\bar{\mathbf{v}}_t)\middle\|p(\boldsymbol{\theta}_t;\boldsymbol{\theta})\right) = \mathbf{0}. \tag{24}$$

Upon defining $\bar{\mathbf{D}}_t := \mathrm{diag}(\bar{\mathbf{d}}_t)$, the KL-divergence of Gaussian distributions can be written as

$$\mathrm{KL}\left(q(\boldsymbol{\theta}_t;\bar{\mathbf{v}}_t)\middle\|p(\boldsymbol{\theta}_t;\boldsymbol{\theta})\right) = \frac{1}{2}\Big(\mathrm{tr}(\mathbf{D}^{-1}\bar{\mathbf{D}}_t) - n + (\mathbf{m}-\bar{\mathbf{m}}_t)^\top\mathbf{D}^{-1}(\mathbf{m}-\bar{\mathbf{m}}_t) + \log\frac{|\mathbf{D}|}{|\bar{\mathbf{D}}_t|}\Big)$$
$$= \frac{1}{2}\sum_{i=1}^{d}\Big(\frac{[\bar{\mathbf{d}}_t]_i}{[\mathbf{d}]_i} - 1 + \frac{([\mathbf{m}]_i - [\bar{\mathbf{m}}_t]_i)^2}{[\mathbf{d}]_i} + \log[\mathbf{d}]_i - \log[\bar{\mathbf{d}}_t]_i\Big), \tag{25}$$

and after plugging (25) into (24) and rearranging terms, we arrive at

$$\bar{\mathbf{m}}_t = \mathbf{m} - \mathbf{D}\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) \tag{26}$$

and

$$\bar{\mathbf{d}}_t = \Big(\mathbf{d}^{-1} + 2\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\Big)^{-1} \tag{27}$$

where we used $\mathbf{v}^{-1}$ to represent the element-wise inverse of a general vector $\mathbf{v}$.

Then, taking gradient w.r.t. $\boldsymbol{\theta} = [\mathbf{m}^\top, \mathbf{d}^\top]^\top$ on both sides of (26), and employing the chain rule results in

$$\nabla_{\mathbf{m}}\bar{\mathbf{m}}_t = \mathbf{I}_d - \big(\nabla_{\mathbf{m}}\bar{\mathbf{m}}_t\nabla^2_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + \nabla_{\mathbf{m}}\bar{\mathbf{d}}_t\nabla_{\bar{\mathbf{d}}_t}\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\big)\mathbf{D} \tag{28}$$

and

$$\nabla_{\mathbf{d}}\bar{\mathbf{m}}_t = -\,\mathrm{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) - \big(\nabla_{\mathbf{d}}\bar{\mathbf{m}}_t\nabla^2_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + \nabla_{\mathbf{d}}\bar{\mathbf{d}}_t\nabla_{\bar{\mathbf{d}}_t}\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\big)\mathbf{D}. \tag{29}$$

Applying the same operation to (27), yields

$$\nabla_{\mathbf{m}}\bar{\mathbf{d}}_t = -2\big(\nabla_{\mathbf{m}}\bar{\mathbf{d}}_t\nabla^2_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + \nabla_{\mathbf{m}}\bar{\mathbf{m}}_t\nabla_{\bar{\mathbf{m}}_t}\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\big)\bar{\mathbf{D}}_t^2 \tag{30}$$

and

$$\nabla_{\mathbf{d}}\bar{\mathbf{d}}_t = -\big(-\mathbf{D}^{-2} + 2\nabla_{\mathbf{d}}\bar{\mathbf{d}}_t\nabla^2_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + 2\nabla_{\mathbf{d}}\bar{\mathbf{m}}_t\nabla_{\bar{\mathbf{m}}_t}\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\big)\bar{\mathbf{D}}_t^2. \tag{31}$$

So far, we have written the four blocks of $\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})$ as a function of themselves through implicit differentiation. Hence, the last step is to solve for these four blocks from the linear equations (28)-(31).

Directly solving this linear system of equations will produce complicated results. The trick here is to reformulate them into a compact matrix form:

$$
\begin{aligned}
&\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta}) \\
&= \left(\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & -\mathbf{D}^{-2} \end{bmatrix} - \begin{bmatrix} \nabla_{\mathbf{m}}\bar{\mathbf{m}}_t & \nabla_{\mathbf{m}}\bar{\mathbf{d}}_t \\ \nabla_{\mathbf{d}}\bar{\mathbf{m}}_t & \nabla_{\mathbf{d}}\bar{\mathbf{d}}_t \end{bmatrix} \begin{bmatrix} \nabla^2_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) & \nabla_{\bar{\mathbf{m}}_t}\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) \\ \nabla_{\bar{\mathbf{d}}_t}\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) & \nabla^2_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & -2\mathbf{I}_d \end{bmatrix}\right) \\
&\quad \times \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & -\bar{\mathbf{D}}_t^2 \end{bmatrix} \\
&= \left(\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & -\mathbf{D}^{-2} \end{bmatrix} - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & -2\mathbf{I}_d \end{bmatrix}\right)\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & -\bar{\mathbf{D}}_t^2 \end{bmatrix} \\
&= \left(\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & \mathbf{D}^{-2} \end{bmatrix} - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & 2\mathbf{I}_d \end{bmatrix}\right)\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \bar{\mathbf{D}}_t^2 \end{bmatrix}.
\end{aligned} \tag{32}
$$

Now, the matrix equation can be readily solved to obtain

$$
\begin{aligned}
\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta}) &= \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & \mathbf{D}^{-2} \end{bmatrix}\left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & 2\mathbf{I}_d \end{bmatrix} + \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \bar{\mathbf{D}}_t^{-2} \end{bmatrix}\right)^{-1} \\
&= \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & \mathbf{D}^{-2} \end{bmatrix}\left(\left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d & \frac{1}{2}\bar{\mathbf{D}}_t^{-2} \end{bmatrix}\right)\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & 2\mathbf{I}_d \end{bmatrix}\right)^{-1} \\
&= \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2} \end{bmatrix}\left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d & \frac{1}{2}\bar{\mathbf{D}}_t^{-2} \end{bmatrix}\right)^{-1} \\
&= \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2} \end{bmatrix}\left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) + \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d & \frac{1}{2}\left(\mathbf{D}^{-1} + 2\operatorname{diag}\left(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\right)^2 \end{bmatrix}\right)^{-1} \\
&= \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ -\operatorname{diag}(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t))\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2} \end{bmatrix}\mathbf{H}_t^{-1}(\bar{\mathbf{v}}_t)
\end{aligned} \tag{33}
$$

where the fourth equality comes from (27).

□

A.2 Proof of Theorem 1

Theorem 1 (Explicit meta-gradient error bound, restated)**.** *Consider the Bayesian meta-learning problem in* (6)*. Let* $\epsilon_t := \|\hat{\mathbf{v}}_t - \bar{\mathbf{v}}_t\|_2$ *be the task-level optimization error, and* $\delta_t := \|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta}) - \mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\|_2$ *the error of the Jacobian. Upon defining* $\rho_t := \max\left\{\|\nabla_{\bar{\mathbf{v}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\|_\infty, \|\nabla_{\hat{\mathbf{v}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\|_\infty\right\}$*, and with Assumptions 1-5 in effect, it holds for* $t \in \{1, \ldots, T\}$ *that*

$$
\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\mathrm{val}}\big(\hat{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big) - \nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\mathrm{val}}\big(\bar{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big)\right\|_2 \leq F_t\epsilon_t + A_t\delta_t, \tag{34}
$$

where the scalar F_t *depends on* ρ_t*.*

Proof. First, it follows by definition (9) of Bayesian meta-gradient that

$$
\begin{aligned}
&\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\mathrm{val}}\big(\hat{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big) - \nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\mathrm{val}}\big(\bar{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big)\right\|_2 \\
&\leq \left\|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\right\|_2 + \left\|\nabla_2\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla_2\mathcal{L}_t^{\mathrm{val}}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\right\|_2 \\
&\leq \left\|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\right\|_2 + C_t\epsilon_t \\
&\leq \left\|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta})\right\|_2 \\
&\quad + \left\|\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\mathrm{val}}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\right\|_2 + C_t\epsilon_t \\
&\leq \left\|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\right\|_2\left\|\nabla_1\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta})\right\|_2 + \left\|\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\right\|_2\left\|\nabla_1\mathcal{L}_t^{\mathrm{val}}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla_1\mathcal{L}_t^{\mathrm{val}}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\right\|_2 + C_t\epsilon_t \\
&\leq A_t\left\|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\right\|_2 + B_t\epsilon_t\left\|\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\right\|_2 + C_t\epsilon_t,
\end{aligned} \tag{35}
$$

where Assumption 2 was used in the second and last inequalities. What remains is to bound $\|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\|_2$ and $\|\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\|_2$.

Using Lemma 1 with Assumption 1, we obtain

$$\begin{aligned}
\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta}) &= \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2} \end{bmatrix} \mathbf{H}_t^{-1}(\bar{\mathbf{v}}_t) \\
&= \begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ 2\mathbf{D}^2\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & 2\mathbf{D}^2 \end{bmatrix}^{-1} \mathbf{H}_t^{-1}(\bar{\mathbf{v}}_t) \\
&= \Bigg(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ 2\mathbf{D}^2\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & 2\mathbf{D}^2 \end{bmatrix} \\
&\quad + \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d & \frac{1}{2}\left(\mathbf{D}^{-1} + 2\operatorname{diag}\left(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\right)^2 \end{bmatrix}\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ 2\mathbf{D}^2\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & 2\mathbf{D}^2 \end{bmatrix}\Bigg)^{-1} \\
&= \Bigg(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & 2\mathbf{D}^2 \end{bmatrix}\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & \mathbf{I}_d \end{bmatrix} \\
&\quad + \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \left(\mathbf{I}_d + 2\operatorname{diag}\left(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\mathbf{D}\right)^2 \end{bmatrix}\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & \mathbf{I}_d \end{bmatrix}\Bigg)^{-1} \\
&:= \left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\bar{\mathbf{P}}_t + \bar{\mathbf{Q}}_t\bar{\mathbf{R}}_t\right)^{-1},
\end{aligned} \tag{36}$$

where the third equality is from the definition of $\mathbf{H}_t(\bar{\mathbf{v}}_t)$. Likewise, we also have

$$\begin{aligned}
\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t) &= \Bigg(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & 2\mathbf{D}^2 \end{bmatrix}\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{I}_d \end{bmatrix} \\
&\qquad + \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \left(\mathbf{I}_d + 2\operatorname{diag}\left(\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)\mathbf{D}\right)^2 \end{bmatrix}\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{I}_d \end{bmatrix}\Bigg)^{-1} \\
&:= \left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\hat{\mathbf{P}}_t + \hat{\mathbf{Q}}_t\hat{\mathbf{R}}_t\right)^{-1}.
\end{aligned} \tag{37}$$

Upon defining $\Delta := \left(\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\right)^{-1} - \left(\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\right)^{-1}$, and adding intermediate terms, we arrive at

$$\begin{aligned}
\|\Delta\|_2 &= \left\|\left(\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\right)^{-1} - \nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\hat{\mathbf{P}}_t - \bar{\mathbf{Q}}_t\hat{\mathbf{R}}_t + \nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\hat{\mathbf{P}}_t + \bar{\mathbf{Q}}_t\hat{\mathbf{R}}_t - \left(\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\right)^{-1}\right\|_2 \\
&= \left\|\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\left(\bar{\mathbf{P}}_t - \hat{\mathbf{P}}_t\right) + \bar{\mathbf{Q}}_t\left(\bar{\mathbf{R}}_t - \hat{\mathbf{R}}_t\right) + \left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) - \nabla^2\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)\hat{\mathbf{P}}_t + \left(\bar{\mathbf{Q}}_t - \hat{\mathbf{Q}}_t\right)\hat{\mathbf{R}}_t\right\|_2 \\
&\le \left\|\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\left(\bar{\mathbf{P}}_t - \hat{\mathbf{P}}_t\right)\right\|_2 + \left\|\bar{\mathbf{Q}}_t(\bar{\mathbf{R}}_t - \hat{\mathbf{R}}_t)\right\|_2 + \left\|\left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) - \nabla^2\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)\hat{\mathbf{P}}_t\right\|_2 + \left\|\left(\bar{\mathbf{Q}}_t - \hat{\mathbf{Q}}_t\right)\hat{\mathbf{R}}_t\right\|_2.
\end{aligned} \tag{38}$$

Next, we will bound the four summands in (43). Using Assumption 3, it follows that

$$\begin{aligned}
\left\|\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\left(\bar{\mathbf{P}}_t - \hat{\mathbf{P}}_t\right)\right\|_2 &\le \left\|\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right\|_2\left\|\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & 2\mathbf{D}^2 \end{bmatrix}\right\|_2\left\|\begin{bmatrix} \mathbf{0}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) - \nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{0}_d \end{bmatrix}\right\|_2 \\
&\le D_t\max\left\{D_{\max}, 2D_{\max}^2\right\}\|\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) - \nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\|_\infty \\
&\le D_t^2\max\left\{D_{\max}, 2D_{\max}^2\right\}\|\bar{\mathbf{m}}_t - \hat{\mathbf{m}}_t\|_\infty \\
&\le D_t^2\max\left\{D_{\max}, 2D_{\max}^2\right\}\epsilon_t,
\end{aligned} \tag{39}$$

and

$$\begin{aligned}
\left\|\left(\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) - \nabla^2\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)\hat{\mathbf{P}}_t\right\|_2 &\le \left\|\nabla^2\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t) - \nabla^2\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right\|_2\left\|\begin{bmatrix} \mathbf{D} & \mathbf{0}_d \\ \mathbf{0}_d & 2\mathbf{D}^2 \end{bmatrix}\right\|_2\left\|\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{I}_d \end{bmatrix}\right\|_2 \\
&\le E_t\epsilon_t\max\left\{D_{\max}, 2D_{\max}^2\right\}\left\|\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{I}_d \end{bmatrix}\right\|_2 \\
&= E_t\epsilon_t\max\left\{D_{\max}, 2D_{\max}^2\right\}\left(1 + \left\|\begin{bmatrix} \mathbf{0}_d & \mathbf{0}_d \\ \operatorname{diag}\left(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{0}_d \end{bmatrix}\right\|_2\right) \\
&= E_t\max\left\{D_{\max}, 2D_{\max}^2\right\}(1 + \|\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\|_\infty)\epsilon_t \\
&\le E_t\max\left\{D_{\max}, 2D_{\max}^2\right\}(1 + \rho_t)\epsilon_t.
\end{aligned} \tag{40}$$

Letting $\mathbf{v}^2$ denote the element-wise square of a general vector $\mathbf{v}$, we have for the second term that

$$\begin{aligned}
\left\|\bar{\mathbf{Q}}_t(\bar{\mathbf{R}}_t-\hat{\mathbf{R}}_t)\right\|_2 &\le \left\|\begin{bmatrix}\mathbf{I}_d & \mathbf{0}_d\\ \mathbf{0}_d & \left(\mathbf{I}_d+2\operatorname{diag}\left(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\mathbf{D}\right)^2\end{bmatrix}\right\|_2 \left\|\begin{bmatrix}\mathbf{0}_d & \mathbf{0}_d\\ \operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)-\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{0}_d\end{bmatrix}\right\|_2\\
&= \max\left\{1,\left\|\left(\mathbf{1}_d+2\mathbf{d}\cdot\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)^2\right\|_\infty\right\}\left\|\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)-\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right\|_\infty\\
&\le \max\left\{1,\left\|\mathbf{1}_d+2\mathbf{d}\cdot\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right\|_\infty^2\right\}D_t\|\bar{\mathbf{m}}_t-\hat{\mathbf{m}}_t\|_\infty\\
&\le D_t(1+2\max\left\{D_{\max},2D_{\max}^2\right\}\rho_t)^2\epsilon_t,
\end{aligned}\tag{41}$$

where for the fourth inequality we employed Assumption 3, and the definition $\mathbf{1}_d:=[1,\dots,1]^\top\in\mathbb{R}^d$.

For the last term, it holds that

$$\begin{aligned}
&\left\|(\bar{\mathbf{Q}}_t-\hat{\mathbf{Q}}_t)\hat{\mathbf{R}}_t\right\|_2\\
&\le \left\|\begin{bmatrix}\mathbf{0}_d & \mathbf{0}_d\\ \mathbf{0}_d & \left(\mathbf{I}_d+2\operatorname{diag}\left(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)\mathbf{D}\right)^2-\left(\mathbf{I}_d+2\operatorname{diag}\left(\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)\mathbf{D}\right)^2\end{bmatrix}\right\|_2 \left\|\begin{bmatrix}\mathbf{I}_d & \mathbf{0}_d\\ \operatorname{diag}\left(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right) & \mathbf{I}_d\end{bmatrix}\right\|\\
&= \left\|\left(\mathbf{1}_d+2\mathbf{d}\cdot\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)^2-\left(\mathbf{1}_d+2\mathbf{d}\cdot\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)^2\right\|_\infty\left(1+\left\|\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right\|_\infty\right)\\
&\le \left\|\left(\mathbf{1}_d+2\mathbf{d}\cdot\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right)^2-\left(\mathbf{1}_d+2\mathbf{d}\cdot\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\right)^2\right\|_\infty(1+\rho_t)\\
&= \left\|2\big(\mathbf{1}_d+\mathbf{d}\cdot(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)+\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t))\big)\cdot 2\big(\mathbf{d}\cdot(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)-\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t))\big)\right\|_\infty(1+\rho_t)\\
&\le 4\left\|\mathbf{1}_d+\mathbf{d}\cdot(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)+\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t))\right\|_\infty\left\|\mathbf{d}\cdot(\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)-\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t))\right\|_\infty(1+\rho_t)\\
&\le 4(1+\max\left\{D_{\max},2D_{\max}^2\right\}\|\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)+\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\|_\infty)\max\left\{D_{\max},2D_{\max}^2\right\}\|\nabla_{\bar{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)-\nabla_{\hat{\mathbf{d}}_t}\mathcal{L}_t^{\mathrm{tr}}(\hat{\mathbf{v}}_t)\|_\infty(1+\rho_t)\\
&\overset{(a)}{\le} 4(1+2\max\left\{D_{\max},2D_{\max}^2\right\}\rho_t)\max\left\{D_{\max},2D_{\max}^2\right\}D_t\|\mathbf{d}_t-\hat{\mathbf{d}}_t\|_\infty(1+\rho_t)\\
&\le 4D_t\max\left\{D_{\max},2D_{\max}^2\right\}(1+2\max\left\{D_{\max},2D_{\max}^2\right\}\rho_t)(1+\rho_t)\epsilon_t,
\end{aligned}\tag{42}$$

where (a) utilizes Assumption 3.

Combining (38)-(42), we arrive at

$$\begin{aligned}
\|\Delta\|_2 \le&\big\{D_t^2\max\left\{D_{\max},2D_{\max}^2\right\}+E_t\max\left\{D_{\max},2D_{\max}^2\right\}(1+\rho_t)+D_t(1+2\max\left\{D_{\max},2D_{\max}^2\right\}\rho_t)^2\\
&+4D_t\max\left\{D_{\max},2D_{\max}^2\right\}(1+2\max\left\{D_{\max},2D_{\max}^2\right\}\rho_t)(1+\rho_t)\big\}\epsilon_t\\
:=&\ F_t^{\Delta}\epsilon_t.
\end{aligned}\tag{43}$$

Further, we can use Assumption 4 to establish one of the desired upper bounds

$$\begin{aligned}
\|\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\|_2 &\le \left\|\begin{bmatrix}\mathbf{D}^{-1} & \mathbf{0}_d\\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & \frac{1}{2}\mathbf{D}^{-2}\end{bmatrix}\right\|_2\left\|\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\right\|_2\\
&\le \left\|\begin{bmatrix}\mathbf{I}_d & \mathbf{0}_d\\ -\operatorname{diag}\left(\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\right) & \mathbf{I}_d\end{bmatrix}\right\|_2\left\|\begin{bmatrix}\mathbf{D}^{-1} & \mathbf{0}_d\\ \mathbf{0}_d & \frac{1}{2}\mathbf{D}^{-2}\end{bmatrix}\right\|_2\sigma_t^{-1}\\
&= \left(1+\|\nabla_{\bar{\mathbf{m}}_t}\mathcal{L}_t^{\mathrm{tr}}(\bar{\mathbf{v}}_t)\|_\infty\right)\max\{\|\mathbf{d}^{-1}\|_\infty,\|\frac{1}{2}\mathbf{d}^{-2}\|_\infty\}\sigma_t^{-1}\\
&\le (1+\rho_t)\max\left\{D_{\min}^{-1},\frac{1}{2}D_{\min}^{-2}\right\}\sigma_t^{-1},
\end{aligned}\tag{44}$$

and likewise

$$\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\|_2\le(1+\rho_t)\max\left\{D_{\min}^{-1},\frac{1}{2}D_{\min}^{-2}\right\}\sigma_t^{-1}.\tag{45}$$

Through (43) and (45), we can also establish the other upper bound as

$$\begin{aligned}
\|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})-\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\|_2 &\le \|\nabla\hat{\mathcal{A}}_t(\boldsymbol{\theta})-\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\|_2+\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)-\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\|_2\\
&= \delta_t+\|\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\Delta\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\|_2\\
&\le \delta_t+\|\nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\|_2\|\Delta\|_2\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\|_2\\
&\le \delta_t+(1+\rho_t)^2\min\left\{D_{\min},2D_{\min}^2\right\}^{-2}\sigma_t^{-2}F_t^{\Delta}\epsilon_t.
\end{aligned}\tag{46}$$

Finally, relating (44) and (46) to (35) completes the proof of the theorem. □

A.3 Proof of Theorem 2

Theorem 2 (implicit gradient error bound, restated)**.** *Consider the Bayesian meta-learning problem in* (6)*. Let* $\epsilon_t := \|\hat{\mathbf{v}}_t - \bar{\mathbf{v}}_t\|_2$ *be the task-level optimization error, and* $\delta_t' := \|\hat{\mathbf{u}}_t - \mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1 \mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta})\|$ *the CG error. Upon defining* $\rho_t := \max\big\{\|\nabla_{\bar{\mathbf{v}}_t}\mathcal{L}_t^{\rm tr}(\bar{\mathbf{v}}_t)\|_\infty, \|\nabla_{\hat{\mathbf{v}}_t}\mathcal{L}_t^{\rm tr}(\hat{\mathbf{v}}_t)\|_\infty\big\}$*, and with Assumptions 1-5 in effect, it holds for* $t \in \{1, \ldots, T\}$ *that*

$$\big\|\hat{\mathbf{g}}_t - \nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\rm val}\big(\bar{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big)\big\|_2 \leq F_t'\epsilon_t + G_t'\delta_t', \tag{47}$$

where F_t' *and* G_t' *are scalars not dependent on* ρ_t*.*

Proof. From (9) and (17), we deduce that

$$\begin{aligned}
&\big\|\hat{\mathbf{g}}_t - \nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\rm val}\big(\bar{\mathbf{v}}_t(\boldsymbol{\theta}), \boldsymbol{\theta}\big)\big\|_2 \\
&\leq \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\hat{\mathbf{u}}_t - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 + \big\|\nabla_2\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla_2\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 \\
&\overset{(a)}{\leq} \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\hat{\mathbf{u}}_t - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 + C_t\epsilon_t \\
&= \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\hat{\mathbf{u}}_t - \mathbf{G}_t(\bar{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\bar{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 + C_t\epsilon_t \\
&\leq \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\big(\hat{\mathbf{u}}_t - \mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta})\big)\big\|_2 \\
&\quad + \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 + C_t\epsilon_t \\
&\overset{(b)}{\leq} (1+\rho_t)\max\big\{D_{\min}^{-1}, \frac{1}{2}D_{\min}^{-2}\big\}\delta_t' + \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 + C_t\epsilon_t,
\end{aligned} \tag{48}$$

where (a) comes from Assumption 2, and (b) uses that

$$\begin{aligned}
\big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\big\|_2 &= \left\|\begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ -\operatorname{diag}\big(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\rm tr}(\hat{\mathbf{v}}_t)\big)\mathbf{D}^{-1} & \frac{1}{2}\mathbf{D}^{-2}\end{bmatrix}\right\|_2 \\
&\leq \left\|\begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ -\operatorname{diag}\big(\nabla_{\hat{\mathbf{m}}_t}\mathcal{L}_t^{\rm tr}(\hat{\mathbf{v}}_t)\big) & \mathbf{I}_d\end{bmatrix}\right\|_2 \left\|\begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d & \frac{1}{2}\mathbf{D}^{-2}\end{bmatrix}\right\|_2 \\
&\leq (1+\rho_t)\max\big\{D_{\min}^{-1}, \frac{1}{2}D_{\min}^{-2}\big\}.
\end{aligned} \tag{49}$$

To bound $\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\|_2$, we again add intermediate terms to arrive at

$$\begin{aligned}
&\big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 \\
&\leq \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 \\
&\quad + \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 \\
&\leq \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t)\big\|_2\big\|\nabla_1\mathcal{L}_t^{\rm val}(\hat{\mathbf{v}}_t, \boldsymbol{\theta}) - \nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 + \big\|\mathbf{G}_t(\hat{\mathbf{v}}_t)\mathbf{H}_t^{-1}(\hat{\mathbf{v}}_t) - \nabla\bar{\mathcal{A}}_t(\boldsymbol{\theta})\big\|_2\big\|\nabla_1\mathcal{L}_t^{\rm val}(\bar{\mathbf{v}}_t, \boldsymbol{\theta})\big\|_2 \\
&\leq (1+\rho_t)\max\big\{D_{\min}^{-1}, \frac{1}{2}D_{\min}^{-2}\big\}\sigma_t^{-1}B_t\epsilon_t + (1+\rho_t)^2\max\big\{D_{\min}^{-2}, \frac{1}{4}D_{\min}^{-4}\big\}\sigma_t^{-2}F_t^{\Delta}\epsilon_t A_t \\
&= \big(B_t(1+\rho_t)\max\big\{D_{\min}^{-1}, \frac{1}{2}D_{\min}^{-2}\big\}\sigma_t^{-1} + A_t(1+\rho_t)^2\max\big\{D_{\min}^{-2}, \frac{1}{4}D_{\min}^{-4}\big\}\sigma_t^{-2}F_t^{\Delta}\big)\epsilon_t
\end{aligned} \tag{50}$$

where the third inequality follows from (45), (46) and Assumption 2.

Plugging (50) into (48) completes the proof of the theorem.

□

A.4 Detailed setups for numerical tests

Synthetic dataset Across all tests, the dimension $d = 32$, and the standard deviation of AWGN is $\sigma = 0.01$. Matrix $\mathbf{X}_t^{\rm tr}$ is randomly generated with condition number $\kappa = 20$, and the linear weights are randomly sampled from the oracle distribution $p(\boldsymbol{\theta}_t; \boldsymbol{\theta}^*) = \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$. The size of the training and validation sets are fixed as $|\mathcal{D}_t^{\rm tr}| = 32$ and $|\mathcal{D}_t^{\rm val}| = 64$ for $t \in \{1, \ldots, T\}$. The task-level optimization function $\hat{\mathcal{A}}_t$ is chosen to be the K-step GD with learning rate $\alpha = 0.01$. To run $\hat{\mathcal{A}}_t$ and compute the meta-loss in (6), the number of Monte Carlo (MC) samples is set to 64.

***Mini*ImageNet** The numerical tests on *mini*ImageNet follow the few-learning protocol described in (Vinyals et al. 2016; Finn, Abbeel, and Levine 2017). For meta-level optimization, the total number of iterations is $40,000$ with batch size $|\mathcal{B}^r| = 2$ and meta-learning rate $\beta = 0.001$. The meta-level prior of ABML is set to Gamma$(\mathbf{1}_d, 0.01 * \mathbf{1}_d)$ according to (Ravi and Beatson 2019). For task-level optimization, the learning rate is $\alpha = 0.01$. In addition, the number of MC runs is taken to be 5 for meta-training, and 10 for evaluation.

Furthermore, to ensure that the entries $[\mathbf{d}]_i$ and $[\mathbf{d}_t]_i$ of the variances are greater than 0, we instead optimize $\log[\mathbf{d}]_i$ and $\log[\mathbf{d}_t]_i$. This is possible because for a general d, it holds that $\nabla_{\log d} f(d) = \nabla_{\log d} d\nabla_d f(d) = d\nabla f(d)$.