
A Distribution-Dependent Analysis of Meta-Learning

Mikhail Konobeev¹ Ilja Kuzborskij² Csaba Szepesvári^{1,2}

Abstract

A key problem in the theory of meta-learning is to understand how the task distributions influence transfer risk, the expected error of a meta-learner on a new task drawn from the unknown task distribution. In this paper, focusing on fixed design linear regression with Gaussian noise and a Gaussian task (or parameter) distribution, we give distribution-dependent lower bounds on the transfer risk of *any* algorithm, while we also show that a *novel, weighted* version of the so-called biased regularized regression method is able to match these lower bounds up to a fixed constant factor. Notably, the weighting is derived from the covariance of the Gaussian task distribution. Altogether, our results provide a precise characterization of the difficulty of meta-learning in this Gaussian setting. While this problem setting may appear simple, we show that it is rich enough to unify the “parameter sharing” and “representation learning” streams of meta-learning; in particular, representation learning is obtained as the special case when the covariance matrix of the task distribution is unknown. For this case we propose to adopt the EM method, which is shown to enjoy efficient updates in our case. The paper is completed by an empirical study of EM. In particular, our experimental results show that the EM algorithm can attain the lower bound as the number of tasks grows, while the algorithm is also successful in competing with its alternatives when used in a representation learning context.

1. Introduction

In meta-learning, a learner uses data from past tasks in an attempt to speed up learning on future tasks. Whether a speedup is possible depends on whether the new task is

¹Computing Science Department, University of Alberta, Edmonton, Alberta, Canada ²DeepMind, London, United Kingdom. Correspondence to: Mikhail Konobeev <konobeev.michael@gmail.com>.

“similar” to the previous ones. In the formal framework of statistical *meta-learning* of Baxter (2000), the learner is given a sequence of training “sets”. The data in each set is independently sampled from an unknown distribution specific to the set, or task, while each such *task distribution* is independently sampled from an unknown meta-distribution, which we shall just call the *environment*. The learner’s *transfer risk* then is its expected prediction loss on a *target* task freshly sampled from the environment. *Can a learner achieve smaller transfer risk by using data from the possibly unrelated tasks? What are the limits of reducing transfer risk?*

As an instructive example, consider a popular approach where each of the n tasks is associated with ground truth parameters $\theta_i \in \mathbb{R}^d$, each of which is assumed to lie close to an unknown vector α that characterizes the environment. To estimate the unknown parameter vector of the last task, one possibility is to employ *biased regularization* (Yang et al., 2007; Kuzborskij & Orabona, 2013; Pentina & Lampert, 2014), that is, solve the optimization problem

$$\min_{\theta} \left\{ \hat{\mathcal{L}}_n(\theta) + \frac{\lambda}{2} \|\theta - \hat{\alpha}\|^2 \right\}$$

where $\hat{\mathcal{L}}_n(\cdot)$ is the empirical loss on the n th task, $\lambda > 0$ is a regularization parameter that governs the strength of the regularization term that biases the solution towards $\hat{\alpha}$, an estimate of α . Here, $\hat{\alpha}$ could be obtained, for example, by averaging parameters estimated on previous tasks (Denevi et al., 2018). This procedure implements the maxim “learn on a new task, but stay close to what is already learned”, which is the basis of many successful meta-learning algorithms, including the above, and MAML (Finn et al., 2017).

Early theoretical work in the area focused on studying the *generalization gap*, which is the difference between the transfer risk and its empirical counterpart. Maurer (2005) gives an upper bound on this gap for a concrete algorithm which is similar to the biased regularization approach discussed above. While these bounds are reassuring, they need further work to fully quantify the benefit of meta-learning, i.e., the gap between the risk of a standard (non-meta) learner and the transfer risk of a meta-learner. Numerous other works have shown bounds on the generalization gap when using biased regularization, in one-shot learning (Kuzborskij & Orabona, 2016), meta-learning (Pentina

& Lampert, 2014), and sequential learning of tasks (Denevi et al., 2018; 2019; Khodak et al., 2019a;b; Finn et al., 2019). While some of these works introduced a dependence on the environment distribution, or on the “regularity” of the sequence of tasks as appropriate, *they still leave open the question whether the shown dependence is best possible.*

In summary, the main weakness of the cited literature is the *lack of (problem dependent) lower bounds*: To be able to separate “good” meta-learning methods from “poor” ones, one needs to know the best achievable performance in a given problem setting. In learning theory, the most often used lower bounds are *distribution-free* or *problem independent*. In the context of meta learning, the distribution refers to the distribution over the tasks, or the environment. The major limitation of a distribution-free approach is that if the class of environments is sufficiently rich, all that the bound will tell us is that the best standard learner will have similar performance to that of the best meta-learner since the worst-case environment will be one where the tasks are completely unrelated. As an example, for a linear regression setting with d -dimensional parameter vectors, Lucas et al. (2021) gives the worst-case lower bound $\Omega(d/((2r)^{-d}M + m))$ for parameter identification where the error is measured in the squared Euclidean distance. Here, M is the total number of data points in the identically-sized training sets, m is the number of data points in the training set of the target task, and $r \geq 1$ is the radius of the ball that contains the parameter vectors.¹ It follows that as $r \rightarrow \infty$, the lower bound reduces to that of linear regression and we see that any method that ignores the tasks is competitive with the best meta-learning method. The pioneering works of Maurer (2009); Maurer et al. (2016) avoid this pathology by introducing empirical quantities that capture task relatedness in the context of linear regression with a common low-dimensional representation.

The bounds can be refined and the pathological limit can be avoided by restricting the set of environments. This approach is taken by Du et al. (2020) and Tripuraneni et al. (2020) who also consider linear regression where the tasks share a common low-dimensional representation. Their main results show that natural algorithms can take advantage of this extra structure. In addition, Tripuraneni et al. (2020) also shows a lower bound on the transfer risk which is matched by their method up to logarithmic factors and problem dependent “conditioning” constants.

¹This result is stated in Theorem 5 in their paper and the setting is meta linear regression. For readability, we dropped some constants, such as label noise variance and slightly generalized the cited result by introducing r , which is taken to be $r = 1$ in their paper. Indeed, the analysis in the paper is not hard to modify to get the dependence shown on r .

Our contributions. In the present paper we revisit the framework underlying biased regularized regression. In particular, we propose to study the case when the unknown parameter vectors for the tasks are generated from a normal distribution with some mean and covariance matrix. First, we consider the case when the mean is unknown while the covariance matrix is known. For this case, in the context of fixed-design linear regression, we prove distribution-dependent lower and upper bounds, which essentially match each other. The lower bound is a direct lower limit on the transfer risk of *any* meta-learning method. The upper bound is proven for a version of a *weighted* biased regularized least-squares regression. Here, the parameters are biased towards the maximum likelihood estimate of the unknown common mean of the task parameter vectors, and the weighting is done with respect to the inverse covariance matrix of the distribution over the task parameter vectors. We show that the maximum likelihood estimator can be efficiently computed, which implies that the entire procedure is efficient. As opposed to the work of Tripuraneni et al. (2020), the gap between the lower and upper bounds is a universal constant, regardless of the other parameters of the meta-learning task. The matching lower and upper bounds together provide a *precise and fine-grained characterization of the benefits of meta-learning*. Our algorithm shows how one should combine datasets of different cardinalities and suggest specific ways of tuning biased regularized regression based on the noise characteristics of the data and the task structure. Our lower bounds are based on a rigorously proven novel observation, which may be of interest on its own. According to this observation, any predictor can be treated as a plug-in method that first estimates the unknown task distribution parameters. Hence, to prove a lower bound for the transfer risk, it suffices to do so for plug-in estimators.

In the last part of the paper we consider the case when the covariance matrix of the task parameter vector distribution is unknown. Importantly, this case can be seen as a way of *unifying* the representation learning stream of meta-learning with the parameter sharing stream. In particular, if the covariance matrix is such that $d - s$ of its eigenvalues tend to zero, while the other eigenvalues s are allowed to take on arbitrarily large values, the problem becomes essentially the same as the representation learning problem of Du et al. (2020); Tripuraneni et al. (2020).

While we provide no theoretical analysis for this case, we give a detailed description of how the Expectation-Maximization (EM) algorithm can be used to tackle this problem. In particular, we show that in this special case the EM algorithm enjoys an efficient *implementation*: we show how to implement the iterative steps in the loop of the EM algorithm in an efficient way. The steps of this algorithm are given as closed-form expressions, which are both intuitive and straightforward to implement. We demonstrate

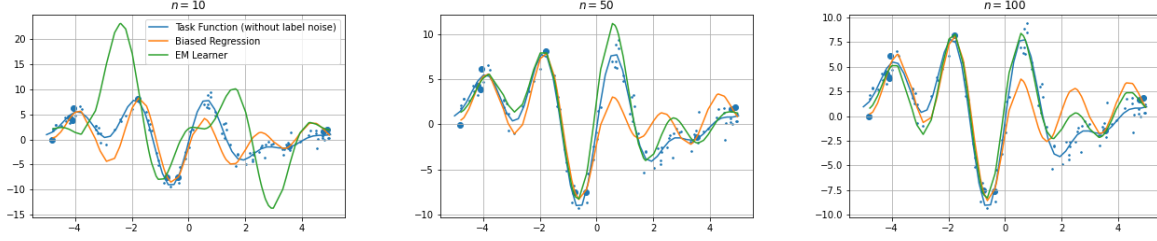


Figure 1. Examples of predictions on the synthetic, ‘Fourier’ meta-learning problem. Training data is shown in bold, small dots show test data. We also show the predictions for two learners (at every input) and the target function. The column correspond to outputs obtained training on $n \in \{10, 50, 100\}$ tasks. Our new algorithm, EM learner, performs quite well.

the effectiveness of the resulting procedure on a number of synthetic and real benchmarks; Fig. 1 shows an example on a synthetic benchmark problem, comparing our EM algorithm with the earlier cited (unweighted) ‘‘biased regression’’ procedure. As can be seen from the figure, the EM based learner is significantly more effective. Further experiments suggest that the EM learner is almost as effective as the optimal biased weighted regularized regression procedure that is given the unknown parameters. We found that the EM learner is also competitive as a representation learning algorithm by comparing it to the algorithm of Tripuraneni et al. (2020) that is based on the ‘‘method-of-moments’’ technique.

2. Setup

In the statistical approach to *meta-learning* (Baxter, 1998; 2000) the learner observes a sequence of *training tuples* $\mathcal{D} = (D_i)_{i=1}^n$, distributed according to a *random* sequence of *task distributions* $(P_i)_{i=1}^n$, i.e. $D_i \sim P_i$, and furthermore task distributions are sampled independently from each other from a fixed and unknown *environment* distribution \mathcal{P} . The focus of this paper is *linear regression with a fixed design* and therefore each training tuple $D_i = ((\mathbf{x}_{i,1}, Y_{i,1}), \dots, (\mathbf{x}_{i,m_i}, Y_{i,m_i}))$ consists of m_i fixed training *inputs* from \mathbb{R}^d and corresponding random, real-valued *targets* satisfying

$$Y_{i,j} = \boldsymbol{\theta}_i^\top \mathbf{x}_{i,j} + \varepsilon_{i,j}, \quad (1)$$

$$\text{where } \varepsilon_{i,j} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2), \boldsymbol{\theta}_i \stackrel{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\alpha}, \boldsymbol{\Sigma}),$$

while $(\varepsilon_{i,j})_{i,j}$ and $(\boldsymbol{\theta}_i)_i$ are also independent from each other.² A meta-learning environment in this setting is thus given by $\boldsymbol{\alpha}$ and the noise parameters $(\sigma^2, \boldsymbol{\Sigma})$. Initially, we will assume that $(\sigma^2, \boldsymbol{\Sigma})$ is known, while $\boldsymbol{\alpha}$ (just like $(\boldsymbol{\theta}_i)_i$) is unknown. The learner observes \mathcal{D} and needs to produce a prediction of the value

$$Y = \boldsymbol{\theta}_n^\top \mathbf{x} + \varepsilon$$

²Technically, P_i consists of Dirac deltas on inputs.

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and where $\mathbf{x} \in \mathbb{R}^d$ is a fixed (non-random) point. Our theoretical results will trivially extend to the case when the learner needs to produce predictions for a sequence of input points or a fixed distribution over these, as often considered in meta-learning literature (Denevi et al., 2018; Du et al., 2020). The (random) transfer risk of the learner is defined as

$$\mathcal{L}(\mathbf{x}) = \mathbb{E} \left[(Y - \hat{Y})^2 \mid \mathcal{D} \right].$$

The setting described above coincides with the standard fixed-design linear regression setup for $n = 1$ and $\boldsymbol{\Sigma} \rightarrow \mathbf{0}$, for which the behavior of risk is well understood. In contrast, the question that meta-learning poses is whether having $n > 1$, one can design a predictor that achieves lower risk compared to the approach that only uses the target data. Naturally, this is of a particular interest in the small sample regime when for all tasks, $m_i \ll n$, that is when facing scarcity of the training data but having many tasks. Broadly speaking, this reduces to understanding the behavior of the risk in terms of the interaction between the number of tasks n , their sample sizes (m_1, \dots, m_n) , and the task structure given by the noise parametrization $(\sigma^2, \boldsymbol{\Sigma})$.

3. Sufficiency of Meta-mean Prediction

In this section we show that there is no loss of generality in considering ‘‘plug-in’’ predictors that predict first the unknown meta-mean $\boldsymbol{\alpha}$. We also show that biased regularized least-squares estimator belongs to this family. We start with some general remarks and notation.

Throughout the rest of the paper, for real symmetric matrices \mathbf{A} and \mathbf{B} , we use $\mathbf{A} \succeq \mathbf{B}$ to indicate that the matrix $\mathbf{A} - \mathbf{B}$ is Positive Semi-Definite (PSD). For $\mathbf{x} \in \mathbb{R}^d$ and PSD matrix \mathbf{A} , we let $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^\top \mathbf{A} \mathbf{x}}$. We use $\|\mathbf{x}\|$ to denote the 2-norm of \mathbf{x} . In the following we will use matrix notation aggregating inputs, targets, and parameters over multiple tasks. In particular, let the cumulative sample size of all tasks be $M = m_1 + \dots + m_n$ and introduce aggregates

for inputs and targets as follows:

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_{i,1}^\top \\ \vdots \\ \mathbf{x}_{i,m_i}^\top \end{bmatrix}, \Psi = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_n \end{bmatrix}, \mathbf{Y}_i = \begin{bmatrix} Y_{i,1} \\ \vdots \\ Y_{i,m_i} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_n \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{X}_n \end{bmatrix}, \Theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}.$$

The matrix representation allows us to compactly state the regression model simultaneously over all tasks. In particular, for the M -dimensional noise vector $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$:

$$\mathbf{Y} = \mathbf{X}\Theta + \varepsilon \Leftrightarrow \mathbf{Y} \sim \mathcal{N}(\Psi\alpha, \mathbf{K}) \quad (2)$$

where α is a meta-mean of model (1) and \mathbf{K} is the *marginal* covariance matrix defined as $\mathbf{K} = \mathbf{X}(\mathbf{I} \otimes \Sigma)\mathbf{X}^\top + \sigma^2 \mathbf{I}$ where \otimes stands for the Kronecker product. Note that the above equivalence comes from a straightforward observation that a linear map \mathbf{X}_i applied to the Gaussian r.v. θ_i is itself Gaussian with mean $\mathbb{E}[Y_i] = \mathbf{X}_i \mathbb{E}[\theta_i] = \mathbf{X}_i \alpha$ and covariance $\mathbf{X}_i \Sigma \mathbf{X}_i^\top + \sigma^2 \mathbf{I}$ which follows from the property that for any random vector ξ with covariance matrix \mathbf{C} , and matrix \mathbf{A} of appropriate dimensions, covariance matrix of $\mathbf{A}\xi$ is $\mathbf{A}\mathbf{C}\mathbf{A}^\top$, ultimately giving Eq. (2).

3.1. Plug-In Predictors and their Sufficiency

Both our lower and upper bounds will be derived from analyzing a family of ‘‘plug-in’’ predictors that aim to estimate θ_n through estimating α . As we shall see, weighted biased regularization is also member of this family.

The said family is motivated by applying the well-known bias-variance decomposition to the risk of an arbitrary predictor $A : \text{supp}(P_1) \times \dots \times \text{supp}(P_n) \times \mathbb{R}^d \rightarrow \mathbb{R}$. Namely,

$$\begin{aligned} \mathcal{L}(\mathbf{x}) &= \mathbb{E}[(Y - A(\mathcal{D}, \mathbf{x}))^2 \mid \mathcal{D}] \\ &= \mathbb{E}[(\mathbb{E}[Y \mid \mathcal{D}] - A(\mathcal{D}, \mathbf{x}))^2 + \mathbb{V}[Y \mid \mathcal{D}] \mid \mathcal{D}] \end{aligned}$$

where we used the law of total expectation and the fact that for any r.v. ξ , $\mathbb{E}[\xi^2] = \mathbb{E}[\xi]^2 + \mathbb{V}[\xi]$. Since the variance term does not depend on A , it follows that the prediction problem reduces to predicting the posterior mean $\mathbb{E}[Y \mid \mathcal{D}]$, which, in our setting, can be given in closed form:

Proposition 3.1. *Let $Y = \theta_n^\top \mathbf{x} + \varepsilon$ for $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and some $\mathbf{x} \in \mathbb{R}^d$. Then,*

$$\begin{aligned} \mathbb{E}[Y \mid \mathcal{D}] &= \mathbf{x}^\top \mathcal{T} \left(\Sigma^{-1} \alpha + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{Y}_n \right) \\ \text{where } \mathcal{T} &= \left(\Sigma^{-1} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{X}_n \right)^{-1}. \end{aligned}$$

Proof. See Appendix A. \square

Since the only unknown parameter here is the meta-mean α , we expect that good predictors will just estimate the meta-mean and use the above formula. That is, these predictors take the form $(\mathcal{D}, \mathbf{x}) \mapsto \mathbf{x}^\top \hat{\theta}_n(\alpha(\mathcal{D}, \mathbf{x}))$, where

$$\hat{\theta}_n(\mathbf{a}) = \mathcal{T} \left(\Sigma^{-1} \mathbf{a} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{Y}_n \right) \quad \mathbf{a} \in \mathbb{R}^d, \quad (3)$$

giving our family of *plug-in predictors*. In fact, *there is no loss in generality by considering only predictors of the above form*. Indeed, given some predictor A and $\mathbf{x} \neq \mathbf{0}$, we can solve $A(\mathcal{D}, \mathbf{x}) = \mathbf{x}^\top \hat{\theta}_n(\alpha)$ for α . One solution is given by $\alpha(\mathcal{D}, \mathbf{x}) = \Sigma \mathcal{T}^{-1} \mathbf{x} c$ where $c = \frac{1}{\|\mathbf{x}\|^2} (A(\mathcal{D}, \mathbf{x}) - \sigma^{-2} \mathbf{x}^\top \mathcal{T} \mathbf{X}_n^\top \mathbf{Y}_n)$. Hence, to prove a lower bound for any regressor A , it will be enough to prove it for algorithms that estimate α .

One special estimator of α is the Maximum Likelihood Estimator (MLE) estimator, and, thanks to (2), can be obtained via $\hat{\alpha}^{\text{MLE}} = \arg \max_{\alpha \in \mathbb{R}^d} \ln p^G(\mathbf{Y}; \Psi \alpha, \mathbf{K})$, where $p^G(\mathbf{x}; \boldsymbol{\mu}, \Sigma) \propto e^{-\frac{1}{2} \|\mathbf{x} - \boldsymbol{\mu}\|_{\Sigma^{-1}}^2}$ is a Gaussian PDF. Some standard calculations give us

$$\hat{\alpha}^{\text{MLE}} = (\Psi^\top \mathbf{K}^{-1} \Psi)^{-1} \Psi^\top \mathbf{K}^{-1} \mathbf{Y}. \quad (4)$$

3.2. Weighted Biased Regularization

Biased regularization is a popular transfer learning technique which commonly appears in the regularized formulations of the empirical risk minimization problems, where one aims at minimizing the empirical risk (such as the mean squared error) while forcing the solution to stay close to some bias variable \mathbf{b} . Here we consider the *Weighted Biased Regularized Least Squares (WBRLS)* formulation defined w.r.t. bias \mathbf{b} and some PSD matrix Γ :

$$\begin{aligned} \hat{\theta}_n^{\text{WBRLS}} &= \arg \min_{\theta \in \mathbb{R}^d} \left\{ \hat{\mathcal{L}}_n(\theta) + \frac{\lambda}{2} \|\theta - \mathbf{b}\|_{\Gamma}^2 \right\} \\ \text{where } \hat{\mathcal{L}}_n(\theta) &= \sum_{j=1}^{m_n} (Y_{n,j} - \theta^\top \mathbf{x}_{n,j})^2. \end{aligned}$$

Remarkably, an estimate $\hat{\theta}_n^{\text{WBRLS}}$ produced by WBRLS is *equivalent* to estimator $\hat{\theta}_n(\hat{\alpha})$ of Eq. (3) for the choice of $\mathbf{b} = \hat{\alpha}$, $\Gamma = \Sigma^{-1}$, and $\lambda = \sigma^2$. Thus, WBRLS is a special member of the family chosen in the previous section.

To see the equivalence, owing to the convenient least-squares formulation, we observe that

$$\hat{\theta}_n^{\text{WBRLS}} = (\mathbf{X}_n^\top \mathbf{X}_n + \lambda \Gamma)^{-1} (\mathbf{X}_n^\top \mathbf{Y}_n + \lambda \Gamma \mathbf{b})$$

and from here the equivalence follows by substitution. A natural question commonly arising in such formulations is how to set the bias term \mathbf{b} . One choice can be $\mathbf{b} = \hat{\alpha}^{\text{MLE}}$ and in the following we will see that it is an optimal one.

4. Problem-Dependent Bounds

We now present our main results, which are essentially matching lower and upper bounds. The upper bounds concern the parameter estimator that uses the MLE estimate of α , while the lower bounds apply to *any* method. We also present a more precise lower bound that applies to estimators that are built on *unbiased* meta-mean estimators $\hat{\alpha}$. As we shall see that plug-in predictors based on MLE will exactly match this lower bound. The general lower bounds are also quite precise: They differ from this lower bound only by a (relatively small) universal constant. We also give a high-probability variant of the same general lower bound.

Theorem 4.1. *Let $\mathbf{x} \in \mathbb{R}^d$ and consider the linear regression model (1). Let $\hat{\alpha}$ be any unbiased estimator of α based on \mathcal{D} . Then the transfer risk $\mathcal{L}(\mathbf{x})$ of the predictor that predicts $\hat{Y} = \mathbf{x}^\top \hat{\theta}_n(\hat{\alpha})$ satisfies*

$$\mathbb{E}[\mathcal{L}(\mathbf{x})] \geq \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2 \quad (5)$$

where $\mathbf{M} = \mathcal{T} \Sigma^{-1} (\Psi^\top \mathbf{K}^{-1} \Psi)^{-1} \Sigma^{-1} \mathcal{T}$.

Moreover, for all predictors we have

$$\mathbb{E}[\mathcal{L}(\mathbf{x})] \geq \frac{\mathbf{x}^\top \mathbf{M} \mathbf{x}}{16\sqrt{e}} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2. \quad (6)$$

Finally, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ for all predictors we have

$$\mathcal{L}(\mathbf{x}) \geq \frac{1}{2} \log \left(\frac{1}{4(1-\delta)} \right) \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2.$$

Proof. The main ideas of the proof are in Section 4.2, while the complete proof is given in Appendix B. \square

Note that the presented bounds are problem-dependent since they depend on a concrete task structure of the environment characterized by (Σ, σ^2) . While the strength of the above bound is its generality, this generality makes the interpretation of the result challenging. We return to the interpretation of this result momentarily, after presenting results for the transfer risk for the plug-in method that uses the (unbiased) MLE meta-mean estimator $\hat{\alpha}^{\text{MLE}}$ defined in Eq. (4).

Theorem 4.2. *For the estimator $\hat{\theta}_n(\hat{\alpha}^{\text{MLE}})$ and for any $\mathbf{x} \in \mathbb{R}^d$ we have*

$$\mathbb{E}[\mathcal{L}(\mathbf{x})] = \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2. \quad (7)$$

Moreover for the same estimator, with probability at least $1 - \delta$, $\delta \in (0, 1)$ we have

$$\mathcal{L}(\mathbf{x}) \leq 2 \log \left(\frac{2}{\delta} \right) \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2.$$

Proof. See Appendix C. \square

Note that Eq. (7) is an equality for the transfer risk and it matches the lower bound available for unbiased estimators. This result, together with our lower bound shows that (i) the predictors based on $\hat{\alpha}^{\text{MLE}}$ is *optimal, with matching constant* within the set of predictors that is based on unbiased estimators of α . It also follows that (ii) apart from a constant factor of $16\sqrt{e}$ of the transfer risk, this predictor is also optimal among *all* predictors.

4.1. Interpretation of the Results

The following two corollaries specialize the lower bound of Theorem 4.1 in a way that will make the results more transparent. Note that while we give these simplified expressions for the lower bound (specifically, Eq. (6)), these expressions also remain essentially true for the upper bound for the MLE estimator, since these differ only in minor details. The proofs of both corollaries are given in Appendix F.

Both specializations are concerned with the case when the inputs are *isotropic*, meaning that the input covariance matrix of task i is $\frac{m_i}{d} \mathbf{I}$.

In the first result, in addition, we assume a *spherical* task structure: $\Sigma = \tau^2 \mathbf{I}$. Thus, the coordinates of the parameter vectors θ_i are uncorrelated and share the same variance τ^2 .

Corollary 4.3. *Assume the same as in case of Eq. (6). In addition, let $\Sigma = \tau^2 \mathbf{I}$, suppose that $\mathbf{X}_i^\top \mathbf{X}_i = \frac{m_i}{d} \mathbf{I}$, and let $\|\mathbf{x}\| = 1$. Then,*

$$\frac{\mathbb{E}[\mathcal{L}(\mathbf{x})] - \sigma^2}{\sigma^2} \geq \frac{H_{\tau^2}}{16\sqrt{e}} \cdot \frac{d^2 \sigma^2}{n(\tau^2 m_n + d\sigma^2)^2} + \frac{d\tau^2}{\tau^2 m_n + d\sigma^2},$$

where H_z is a harmonic mean of a sequence $(z + \frac{d\sigma^2}{m_i})_{i=1}^n$.

In the above bound the first term vanishes as more tasks are added (n growing). On the other hand, to decrease the second term, m_n needs to increase. In particular, as $n \rightarrow \infty$, we get

$$\frac{\mathbb{E}[\mathcal{L}(\mathbf{x})] - \sigma^2}{\sigma^2} \geq \left(\frac{m_n}{d} + \frac{\sigma^2}{\tau^2} \right)^{-1} \quad (8)$$

where $\frac{m_n}{d} + \frac{\sigma^2}{\tau^2}$ can be interpreted as an *effective sample size*. Thus, while having infinitely many previous tasks have the potential to reduce the loss, the size of this effect is fixed and is related to the noise variance ratios. If $\tau^2 \rightarrow 0$, having infinitely many tasks will allow perfect prediction, but for any $\tau^2 > 0$, there is a limit on how much the data of previous tasks can help. Finally, for the case $n = 1$ and $\tau^2 = 0$ we recover the standard lower bound for linear setting $\mathbb{E}[\mathcal{L}(\mathbf{x})] - \sigma^2 = \Omega(d\sigma^2/m_1)$.

Our next result is concerned with “representation learning”, which corresponds to the case when Σ is a low rank PSD matrix.

Corollary 4.4. *Let the inputs be isotropic as before and $\|\mathbf{x}\| = 1$. Moreover, let Σ be a PSD matrix of rank $s \leq d$ with eigenvalues $\lambda_1 \geq \dots \geq \lambda_s > 0$,³ and suppose that $\|\mathbf{x}\|_{\mathbf{P}_s^\top \mathbf{P}_s}^2 = s/d$ where $\mathbf{P}_s = [\mathbf{u}_1, \dots, \mathbf{u}_s]^\top$ and $(\mathbf{u}_j)_{j=1}^s$ are unit length eigenvectors of Σ . Then,*

$$\frac{\mathbb{E}[\mathcal{L}(\mathbf{x})] - \sigma^2}{\sigma^2} \geq \frac{H_{\lambda_s}}{16\sqrt{e}} \cdot \frac{sd\sigma^2}{n(\lambda_1 m_n + d\sigma^2)^2} + \frac{s\lambda_s}{\lambda_s m_n + d\sigma^2}.$$

Note that the first term on the right-hand side of the last display scales with sd/n , where sd is the number of parameter in a matrix that would give the low-dimensional representation and the second term scales with s/m_n for $m_n \gg d\sigma^2/\lambda_s$. Somewhat surprisingly (given that here Σ is known), these essentially match the upper bounds due to Du et al. (2020); Tripuraneni et al. (2020), implying that their results are unimprovable.

4.2. Proof Sketches

Our lower and upper bounds on the risk are based on an identity that holds for the transfer risk of plug-in methods. The identity is essentially a bias-variance decomposition.

Lemma 4.5. *For $\hat{\theta}_n(\hat{\alpha})$ defined in Eq. (3), any task mean estimator $\hat{\alpha}$, and any $\mathbf{x} \in \mathbb{R}^d$ we have $\mathbb{E}[\mathcal{L}(\mathbf{x})] = \mathbb{E}[(\mathbf{x}^\top \mathcal{T} \Sigma^{-1} (\alpha - \hat{\alpha}))^2] + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2$.*

For the proof of this lemma we need the following proposition, which expands on Proposition 3.1. The proof is given in Appendix A:

Proposition 4.6. *Let $Y = \theta_n^\top \mathbf{x} + \varepsilon$ for $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and some $\mathbf{x} \in \mathbb{R}^d$. Then, $\mathbb{E}[Y | \mathcal{D}] = \mathbf{x}^\top \mathcal{T} (\Sigma^{-1} \alpha + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{Y}_n)$ and $\mathbb{V}[Y | \mathcal{D}] = \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2$.*

Proof of Lemma 4.5. Using the law of total expectation and that for a r.v. ξ we have $\mathbb{E}[\xi^2] = \mathbb{E}[\xi]^2 + \mathbb{V}[\xi]$,

$$\begin{aligned} \mathcal{L}(\mathbf{x}) &= \mathbb{E}[(Y - \hat{\theta}_n(\hat{\alpha})^\top \mathbf{x})^2 | \mathcal{D}] \\ &= \mathbb{E}\left[\left(\mathbb{E}[Y | \mathcal{D}] - \hat{\theta}_n(\hat{\alpha})^\top \mathbf{x}\right)^2 + \mathbb{V}[Y | \mathcal{D}] \mid \mathcal{D}\right] \\ &= \mathbb{E}\left[(\mathbf{x}^\top \mathcal{T} \Sigma^{-1} (\alpha - \hat{\alpha}))^2 \mid \mathcal{D}\right] + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2, \end{aligned}$$

where identities for $\mathbb{E}[Y | \mathcal{D}]$ and $\mathbb{V}[Y | \mathcal{D}]$ come from Proposition 3.1 and identity for $\hat{\theta}_n(\hat{\alpha})$ is due to (3). \square

Thus, to establish universal lower bounds we need to lower bound $\mathbb{E}[(\mathbf{x}^\top \mathcal{T} \Sigma^{-1} (\alpha - \hat{\alpha}))^2]$ for any choice of estimator $\hat{\alpha}$, which in combination with Lemma 4.5 will prove Theorem 4.1. Here, relying on the Cramér-Rao inequality (Theorem B.1), we only prove a lower bound for unbiased estimators, while the general case, whose proof uses Le Cam’s method, is left to Appendix B.2.

³When $s < d$, we replace Σ^{-1} with its pseudo-inverse Σ^\dagger .

Lemma 4.7. *For any unbiased estimator $\hat{\alpha}$ of α in Eq. (2) we have $\mathbb{E}[(\mathbf{x}^\top \mathcal{T} \Sigma^{-1} (\alpha - \hat{\alpha}))^2] \geq \mathbf{x}^\top \mathbf{M} \mathbf{x}$.*

Proof. Recall that according to the equivalence (2), $\mathbf{Y} \sim \mathcal{N}(\Psi\alpha, \mathbf{K})$ and the unknown parameter is α . To compute the Fisher information matrix we first observe that $\nabla_\alpha \ln p^G(\mathbf{Y}; \Psi\alpha, \mathbf{K}) = \Psi^\top \mathbf{K}^{-1}(\mathbf{Y} - \Psi\alpha)$ and

$$\begin{aligned} \mathbf{F} &= \mathbb{E}\left[\nabla_\alpha \ln p^G(\mathbf{Y}; \Psi\alpha, \mathbf{K}) \nabla_\alpha \ln p^G(\mathbf{Y}; \Psi\alpha, \mathbf{K})^\top\right] \\ &= \Psi^\top \mathbf{K}^{-1} \mathbb{E}[(\mathbf{Y} - \Psi\alpha)(\mathbf{Y} - \Psi\alpha)^\top] \mathbf{K}^{-1} \Psi \\ &= \Psi^\top \mathbf{K}^{-1} \Psi. \end{aligned}$$

Thus, by the Cramér-Rao inequality we have $\mathbb{E}[(\alpha - \hat{\alpha})(\alpha - \hat{\alpha})^\top] \succeq (\Psi^\top \mathbf{K}^{-1} \Psi)^{-1}$. Finally, left-multiplying by $\mathbf{x}^\top \mathcal{T} \Sigma^{-1}$ and right-multiplying the above by $\Sigma^{-1} \mathcal{T} \mathbf{x}$ gives us the statement. \square

5. Learning with Unknown Task Structure

So far we have assumed that parameters (σ^2, Σ) characterizing the structure of environment are known, which limits the applicability of the predictor (though does not limit the lower bound). Staying within our framework, a natural idea is to estimate all the environment parameters $\mathcal{E} = (\alpha, \sigma^2, \Sigma)$ by maximizing the data marginal log-likelihood

$$J(\mathcal{D}, \mathcal{E}') = \ln \int_{\mathbb{R}^{nd}} p(\mathcal{D} | \vartheta) d p(\vartheta | \mathcal{E}')$$

over \mathcal{E}' , where $p(\mathcal{D}, \Theta, \mathcal{E})$ stands for the joint distribution in the model (1). The above problem is non-convex. As such, we propose to use EM procedure (Dempster et al., 1977), which is known to be a reasonable algorithm for similar settings.⁴ EM can be derived as a procedure that maximizes a lower bound on $J(\mathcal{D}, \mathcal{E}')$: Jensen’s inequality gives us that for any probability measure q on \mathbb{R}^{nd} , $J(\mathcal{D}, \mathcal{E}') \geq \int \ln \left(\frac{p(\vartheta, \mathcal{D} | \mathcal{E}')}{q(\vartheta)} \right) dq(\vartheta)$. This is then maximized in \mathcal{E}' and q in an alternating fashion: Letting $\hat{\mathcal{E}}_t$ to be a parameter estimate at step t , we maximize the lower bound in q for a fixed $\mathcal{E}' = \hat{\mathcal{E}}_t$, and then obtain $\hat{\mathcal{E}}_{t+1}$ by maximizing the lower bound in \mathcal{E}' for a fixed previously obtained solution in q . Maximization in q gives us $q(\vartheta) = p(\vartheta | \mathcal{D}, \hat{\mathcal{E}}_t)$, while maximization in \mathcal{E}' yields

$$\hat{\mathcal{E}}_{t+1} \in \arg \max_{\mathcal{E}'} \int \ln (p(\vartheta, \mathcal{D} | \mathcal{E}')) d p(\vartheta | \mathcal{D}, \hat{\mathcal{E}}_t). \quad (9)$$

After some calculations (cf. Appendix D), this gives Algorithm 1. During the E-step (lines 4-5), the algorithm computes the parameters of the posterior distribution $\mathcal{N}(\theta_i | \hat{\mu}_{t,i}, \hat{\mathcal{T}}_{t,i})$ relying on $\hat{\mathcal{E}}_t$, and during the M-step

⁴While the marginal distribution is available in analytic form by Eq. (2), we focus on EM because, in preliminary experiments, direct optimization proved to be numerically unstable.

(lines 7–9) it estimates $\hat{\mathcal{E}}_{t+1}$ based on $(\hat{\boldsymbol{\mu}}_{t,i}, \hat{\mathcal{T}}_{t,i})$. We propose to detect convergence (not shown) by checking the relative difference between successive parameter values.

Algorithm 1 EM procedure to estimate $(\boldsymbol{\alpha}, \sigma^2, \boldsymbol{\Sigma})$

Input: Initial parameter estimates $\hat{\mathcal{E}}_1 = (\hat{\boldsymbol{\alpha}}_1, \hat{\sigma}_1^2, \hat{\boldsymbol{\Sigma}}_1)$

Output: Final parameter estimates $\hat{\mathcal{E}}_t = (\hat{\boldsymbol{\alpha}}_t, \hat{\sigma}_t^2, \hat{\boldsymbol{\Sigma}}_t)$

```

1:  $\hat{\mathcal{T}}_{1,i} \leftarrow \mathbf{0}, \hat{\boldsymbol{\mu}}_{1,i} \leftarrow \mathbf{0} \quad i \in [n]$ 
2: repeat
3:   for  $i = 1, \dots, n$  do ▷ E-step
4:      $\hat{\mathcal{T}}_{t,i} \leftarrow \left( \hat{\boldsymbol{\Sigma}}_t^{-1} + \hat{\sigma}_t^{-2} \mathbf{X}_i^\top \mathbf{X}_i \right)^{-1}$ 
5:      $\hat{\boldsymbol{\mu}}_{t,i} \leftarrow \hat{\mathcal{T}}_{t,i} \left( \hat{\boldsymbol{\Sigma}}_t^{-1} \hat{\boldsymbol{\alpha}}_t + \hat{\sigma}_t^{-2} \mathbf{X}_i^\top \mathbf{Y}_i \right)$ 
6:   end for
7:    $\hat{\boldsymbol{\alpha}}_t \leftarrow \frac{1}{n} \sum_{i=1}^n \hat{\boldsymbol{\mu}}_{t,i}$  ▷ M-step
8:    $\hat{\boldsymbol{\Sigma}}_t \leftarrow \frac{1}{n} \sum_{i=1}^n \left( \hat{\mathcal{T}}_{t,i} + (\hat{\boldsymbol{\mu}}_{t,i} - \hat{\boldsymbol{\alpha}}_t)(\hat{\boldsymbol{\mu}}_{t,i} - \hat{\boldsymbol{\alpha}}_t)^\top \right)$ 
9:    $\hat{\sigma}_t^2 \leftarrow \frac{1}{n} \sum_{i=1}^n \frac{1}{m_i} \left( \hat{\mathcal{L}}_i(\hat{\boldsymbol{\mu}}_{t,i}) + \text{tr} \left( \mathbf{X}_i \hat{\mathcal{T}}_{t,i} \mathbf{X}_i^\top \right) \right)$ 
10:   $t \leftarrow t + 1$ 
11: until Convergence (see discussion)
```

6. Experiments

In this section we present experiments⁵ designed to verify three hypotheses: (i) Under ideal circumstances, the predictor $\mathbf{x}^\top \hat{\boldsymbol{\theta}}_n(\hat{\boldsymbol{\alpha}}^{\text{MLE}})$ is superior to its alternatives, including biased, but unweighted regression; (ii) The EM-algorithm reliably recovers unknown parameters of the environment and is also suitable for representation learning; (iii) our distribution-dependent lower bound Eq. (5) is numerically sharp. In addition, we briefly report on experiments with a real-world dataset.

Baselines. We consider two non-meta-learning baselines, that is `Linear Regression (All)` — Ordinary Least Squares (OLS) fitted on $\mathcal{D}^{\setminus n} = (D_i)_{i=1}^{n-1}$, which excludes the newly observed task, and `Linear Regression (Task)` — OLS fitted on a newly encountered task D_n . Next, we consider meta-learning algorithms. We report performance of the *unweighted* Biased Regression procedure with bias set to the least squares solution $(\sum_{i \neq n} \mathbf{X}_i^\top \mathbf{X}_i)^{-1} \sum_{i \neq n} \mathbf{X}_i^\top \mathbf{Y}_i$ and λ found by cross-validation (cf. Appendix E). Note that the bias and the regularization coefficient are found on $\mathcal{D}^{\setminus n}$, while D_n is used for the final fitting. EM Learner is estimator (3) with all environment parameters found by Algorithm 1 on $\mathcal{D}^{\setminus n}$. The convergence threshold was set to 10^{-6} while the maximum number of iterations was set to 10^3 . Finally, we report numerical values of Eq. (5) as `Known Covariance Lower Bound`. For all of the experiments we show aver-

⁵Implementation of all of our experiments and the required dataset are provided in the supplementary material.

ages and standard deviations of the mean test errors computed over 30 independent runs of that experiment.

Synthetic Experiments. We conduct synthetic experiments on datasets with Fourier generated features and features sampled from a d -dimensional unit sphere. In all of the synthetic experiments we have $\boldsymbol{\alpha} = \mathbf{0}, \sigma^2 = 1$ and $\boldsymbol{\Sigma}$ generated by computing $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top + \eta \mathbf{I}$ where $L_{ij} = \mathbb{I}\{i \geq j\} Z_{ij}$ with Z_{ij} and η sampled from the standard normal distribution. Test error is computed on 100 test tasks using 10 examples for training and 100 examples for testing.

For the Fourier-features, we sample a value $u \sim \mathcal{U}(-5, 5)$ and compute features by evaluating $d = 11$ Fourier basis functions at u : $x_j = \mathbb{I}\{1 \leq j \leq 5\} \sin(\frac{j}{5}\pi u) + \mathbb{I}\{6 \leq j \leq 10\} \cos(\frac{j-5}{5}\pi u) + \mathbb{I}\{j = 11\}$, where $\mathbb{I}\{E\} = 1$ if E is true and $\mathbb{I}\{E\} = 0$ otherwise. Examples of these tasks and results of meta-learning on some of these were shown in Fig. 1. In Fig. 2 we show the test errors for various meta-learners while varying the number of tasks n and task sizes m . For the ‘spherical’ data, the same is shown in Fig. 3. Here, we generate \mathbf{x} from a $d = 42$ dimensional unit sphere. In both experiments for sufficiently large number of training tasks the EM-based learner approaches the optimal estimator even when the number of examples per task is less than the dimensionality of that task.

In the context of the ‘Fourier’ dataset, we also experimented with generating low-rank $\boldsymbol{\Sigma}$, corresponding to the challenge of learning a low-dimensional representation, shared across the tasks. We found that the EM-based meta learner stays competitive in this setting. To save space, the results are presented in Appendix G.

Real Dataset Experiment. We also conducted experiments on a real world dataset containing information about students in 139 schools in years 1985-1987 (Dua & Graff, 2017, School Dataset). We adapt the dataset to a meta-learning problem with the goal to predict the exam score of students based on the student-specific and school-specific features. After one-hot encoding of the categorical values there are $d = 27$ features for each student. We randomly split schools into two subsets: The first, consisting of 100 schools, forms $\mathcal{D}^{\setminus n}$ (used for training the bias, λ selection, and EM). The second subset consists of 39 schools, where each school is further split into 80%/20% for the final training and testing of the meta-learners. Results are given in Fig. 4. We can see that while both Biased Regression and EM Learner outperform regression, their performance is very similar. This could be attributed to the fact that the features mostly contain weakly relevant information, which is confirmed by inspecting the coefficient vector.

Representation learning experiments I. Our next figure (Fig. 5) shows the outcomes of experiments for the Fourier task but when $\boldsymbol{\Sigma}$ is low-rank. As can be seen, the EM based

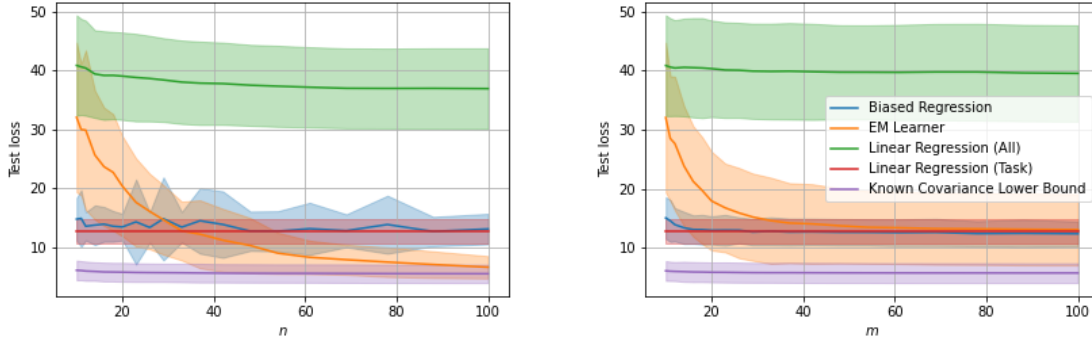


Figure 2. Test errors on Fourier synthetic experiment with changing number of tasks n and number of samples per task m . When one of the parameters changes, the other one is set to 10.

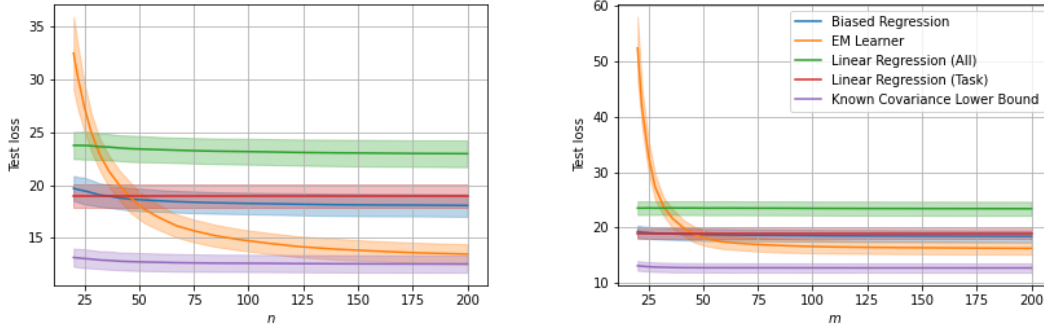


Figure 3. Test error on spherical synthetic experiment with changing number of tasks n and number of samples per task m . When one of the parameters changes, the other one is set to 40.

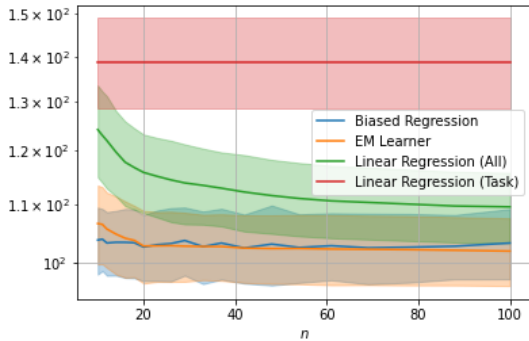


Figure 4. Test error on the School Dataset. Up to 100 schools are used for fitting environment-related parameters (see text for details) and the remaining 39 are used as the target task.

learner excels in exploiting the low-rank structure. For this experiment we have the same setup as for the Fourier experiment, but the covariance matrix Σ is generated by computing $\Sigma = \mathbf{L}\mathbf{L}^\top$ where \mathbf{L} is a $d \times r$ matrix with $r = \lfloor d/2 \rfloor = 5$ and elements $L_{i,j} \sim \mathcal{N}(0, 1)$. Note that in this case we can write $\theta_i = \mathbf{B}\mathbf{w}_i$ for some matrix \mathbf{B} of size $d \times r$ and vector \mathbf{w}_i of size r sampled from multivariate normal

distribution. Thus, if the matrix \mathbf{B} is known or estimated during training, one can project the features $\mathbf{x}_{i,j}$ onto a lower-dimensional space by computing $\mathbf{B}^\top \mathbf{x}_{i,j}$ to speed up the adaptation to new tasks by running least-squares regression to estimate \mathbf{w}_i instead of θ_i .

In addition to the baselines described in the main text, we compared EM Learner with two additional baselines: one is based on the Method of Moments (MoM) estimator from Tripuraneni et al. (2020) (not shown on the figure), and another which we refer to as Oracle Representation. We omit displaying the error of the method of moments estimator since for the features generated as in this experiment it is not able to perform estimation of the subspace and leads to test errors with values around 60. At the same time, as shown on Fig. 5 (left) we observe that EM Learner can outperform Oracle Representation which assumes the knowledge of the covariance matrix Σ from which it computes the subspace matrix \mathbf{B} and uses it to obtain lower-dimensional representation of the features when adapting to a new task via least-squares, as described above. This is possible because the coefficients estimated by EM are biased toward α which does not happen with least squares regression in the lower

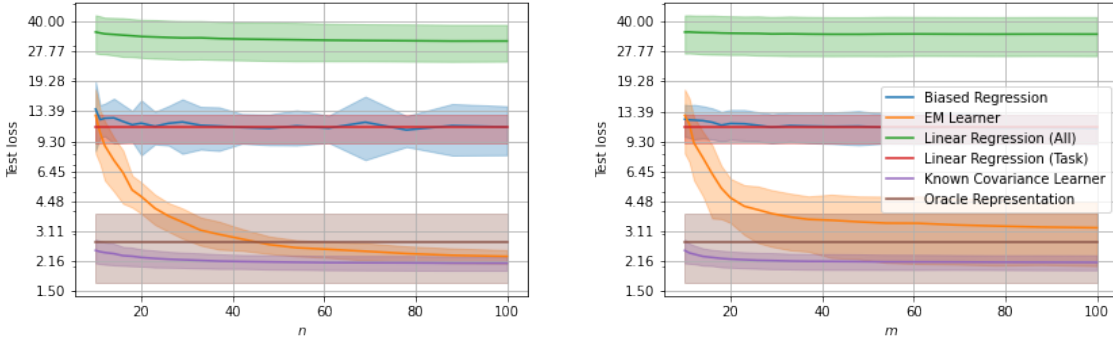


Figure 5. Test error when the task covariance matrix is low-rank. As usual, on the left the number of tasks is changed, on the right, the number of training datapoints (per task). When one parameter is varied, the other is set to the value of 10.

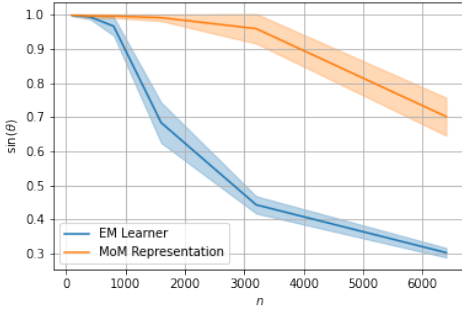


Figure 6. Max-correlation $d_{\max}(\hat{\mathbf{B}}, \mathbf{B})$ between the estimated matrix $\hat{\mathbf{B}}$ (by the respective algorithm) and the ground truth matrix \mathbf{B} while increasing number of tasks n .

dimensional subspace and this is beneficial, especially when the number of test-task training examples is small.

Representation learning experiments II. To validate our implementation of the MoM estimator of Tripuraneni et al. (2020) and to investigate more whether EM is preferable to the MoM estimator beyond the setting that is ideal for the EM method we considered the experimental setup of Tripuraneni et al. (2020).

To explain the setup, we recall that the MoM estimator computes an estimate $\hat{\mathbf{B}}$ of the ground truth matrix \mathbf{B} . Tripuraneni et al. (2020) proves results for the *max-correlation* between $\hat{\mathbf{B}}$ and \mathbf{B} , and also reports experimentally measured max-correlation values between the ground truth and the MoM computed matrix. The max-correlation between matrices \mathbf{A} and \mathbf{A}' is based on the definition of *principal angles* and is equal to $d_{\max}(\mathbf{A}, \mathbf{A}') = \sqrt{1 - \cos^2(\mathbf{A}, \mathbf{A}')}$ where $\cos(\mathbf{A}, \mathbf{A}') = \max_{\mathbf{u} \in \text{span}(\mathbf{A}): \|\mathbf{u}\|=1} \max_{\mathbf{v} \in \text{span}(\mathbf{A}'):\|\mathbf{v}\|=1} \mathbf{u}^\top \mathbf{v}$. Intuitively, max-correlation captures how well the subspaces spanned by matrices \mathbf{A} and \mathbf{A}' are aligned.

To compare our EM estimator to MoM we run the EM estimator as described in Algorithm 1, and once the final estimate $\hat{\Sigma}$ is obtained, we reduce its rank by clipping eigenvalues $\lambda_{s+1} \geq \dots \geq \lambda_d$ to 0.

We follow the experimental setup of Tripuraneni et al. (2020), that is, inputs are generated as $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{I}_d)$, while the regression model is given by Eq. (1) with $(\sigma^2, \Sigma) = (1, \frac{1}{s} \mathbf{B} \mathbf{B}^\top)$. Here, columns of $\mathbf{B} \in \mathbb{R}^{d \times s}$ are sampled from a uniform distribution on a unit d -sphere. Finally, the number of examples per previously observed task is set as $m_1 = \dots = m_{n-1} = 5$, the representation rank is $s = 5$, the input dimension is $d = 100$, and the experiment is repeated 30 times. Since we only estimate the subspace matrix we do not use the data from the test task $(\mathbf{X}_n, \mathbf{y}_n)$.

We report our results in Fig. 6, plotting the max-correlation between $\hat{\mathbf{B}}$ found by the respective algorithm and \mathbf{B} , while increasing the number of tasks. We see that EM learner considerably outperforms MoM Representation in terms of the subspace estimation to the degree captured by max-correlation. While we suspect that the improvement is due to the joint optimization over the covariance of environment and the mean of the environment (the bias in biased regularization), the detailed understanding of this effect is left for the future work.

7. Conclusions

While ours is the first work to derive matching, distribution-dependent lower and upper bounds, much work remains to be done: our approach to derive meta-learning algorithms based on a probabilistic model should be applicable more broadly and could lead to further interesting developments in meta-learning. The most interesting narrower question is to theoretically analyze the EM algorithm. Doing this in the low-rank setting looks particularly interesting. We hope that our paper will inspire other researchers to do further work in this area.

References

- Baxter, J. Theoretical models of learning to learn. In S. Thrun, L. P. (ed.), *Learning to learn*, pp. 71–94. Springer, 1998.
- Baxter, J. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.
- Bretagnolle, J. and Huber, C. Estimation des densités: risque minimax. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 47(2):119–137, 1979.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.
- Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Learning to learn around a common mean. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 10169–10179, 2018.
- Denevi, G., Ciliberto, C., Grazi, R., and Pontil, M. Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning (ICML)*, 2019.
- Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. Few-shot learning via learning the representation, provably. arXiv:2002.09434, 2020.
- Dua, D. and Graff, C. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.
- Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. Online meta-learning. *International Conference on Machine Learning (ICML)*, 2019.
- Khodak, M., Balcan, M.-F., and Talwalkar, A. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning (ICML)*, pp. 424–433, 2019a.
- Khodak, M., Balcan, M.-F. F., and Talwalkar, A. S. Adaptive gradient-based meta-learning methods. *Advances in Neural Information Processing Systems*, 32:5917–5928, 2019b.
- Kuzborskij, I. and Orabona, F. Stability and Hypothesis Transfer Learning. In *International Conference on Machine Learning (ICML)*, pp. 942–950, 2013.
- Kuzborskij, I. and Orabona, F. Fast Rates by Transferring from Auxiliary Hypotheses. *Machine Learning*, pp. 1–25, 2016. ISSN 1573-0565. doi: 10.1007/s10994-016-5594-4.
- Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2018.
- Lucas, J., Ren, M., Kamení, I., Pitassi, T., and Zemel, R. Theoretical bounds on estimation error for meta-learning. In *International Conference on Learning Representations*, 2021.
- Maurer, A. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6(Jun):967–994, 2005.
- Maurer, A. Transfer bounds for linear feature learning. *Machine Learning*, 75(3):327–350, 2009.
- Maurer, A., Pontil, M., and Romera-Paredes, B. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 2016.
- Pentina, A. and Lampert, C. A pac-bayesian bound for lifelong learning. In *International Conference on Machine Learning (ICML)*, pp. 991–999, 2014.
- Tripuraneni, N., Jin, C., and Jordan, M. I. Provable meta-learning of linear representations. arXiv:2002.11684, 2020.
- Yang, J., Yan, R., and Hauptmann, A. G. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM international conference on Multimedia*, pp. 188–197, 2007.

A. Parameters of the Posterior Distribution

Recall that

$$p_n(\boldsymbol{\theta}_n | \mathcal{D}) \propto p^G(\mathbf{Y}_n | \mathbf{X}_n \boldsymbol{\theta}_n, \sigma^2 \mathbf{I}) p^G(\boldsymbol{\theta}_n | \boldsymbol{\alpha}, \boldsymbol{\Sigma}).$$

We first give a handy proposition for the posterior distribution over the parameters $\boldsymbol{\theta}_n$.

Proposition A.1.

$$\ln p_n(\boldsymbol{\theta}_n) = -\frac{1}{2}(\boldsymbol{\theta}_n - \boldsymbol{\mu}) \mathcal{T}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{\mu}) + \text{const}(\boldsymbol{\theta}_n)$$

where covariance is

$$\mathcal{T} = \left(\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{X}_n \right)^{-1}$$

and mean is

$$\boldsymbol{\mu} = \mathcal{T} \left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\alpha} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{Y}_n \right).$$

Proof. The log-likelihood is the following chain of identities:

$$\begin{aligned} \ln p_n(\boldsymbol{\theta}_n) &= -\frac{\sum_{j=1}^{m_n} (\mathbf{x}_{n,j}^\top \boldsymbol{\theta}_n - Y_{n,j})^2}{2\sigma^2} - \frac{1}{2}(\boldsymbol{\theta}_n - \boldsymbol{\alpha})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{\alpha}) + \text{const}(\boldsymbol{\theta}_n) \\ &= -\frac{\sum_{j=1}^{m_n} (\boldsymbol{\theta}_n^\top \mathbf{x}_{n,j} \mathbf{x}_{n,j}^\top \boldsymbol{\theta}_n - 2Y_{n,j} \mathbf{x}_{n,j}^\top \boldsymbol{\theta}_n)}{2\sigma^2} \\ &= -\frac{1}{2}(\boldsymbol{\theta}_n^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\theta}_n - 2\boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\theta}_n) + \text{const}(\boldsymbol{\theta}_n) \\ &= -\frac{1}{2} \left(\boldsymbol{\theta}_n^\top \mathcal{T}^{-1} \boldsymbol{\theta}_n - 2 \left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\alpha} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{Y}_n \right)^\top \boldsymbol{\theta}_n \right) + \text{const}(\boldsymbol{\theta}_n) \\ &= -\frac{1}{2}(\boldsymbol{\theta}_n - \boldsymbol{\mu}) \mathcal{T}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{\mu}) + \text{const}(\boldsymbol{\theta}_n). \end{aligned}$$

□

First note that the first consequence of Proposition A.1 is a MLE for $\boldsymbol{\theta}_n$,

$$\hat{\boldsymbol{\theta}}_n^{\text{MLE}} = \mathcal{T} \left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\alpha} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{Y}_n \right).$$

The second consequence is the following corollary which is obtained by taking $Y = \boldsymbol{\theta}_n^\top \mathbf{x} + \varepsilon$ and simply observing that the mean of a $p_n(\boldsymbol{\theta}_n | \mathcal{D})$ is $\boldsymbol{\mu}$, and so $\mathbb{E}[Y | \mathcal{D}] = \mathbf{x}^\top \boldsymbol{\mu}$ while the variance is $\mathbb{V}[Y | \mathcal{D}] = \mathbb{E}[(\mathbf{x}^\top \boldsymbol{\theta}_n + \varepsilon)^2 | \mathcal{D}] - \mathbb{E}[(\mathbf{x}^\top \boldsymbol{\theta}_n + \varepsilon) | \mathcal{D}]^2 = \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2$.

Proposition 4.6 (restated). *Let $Y = \boldsymbol{\theta}_n^\top \mathbf{x} + \varepsilon$ for $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and some $\mathbf{x} \in \mathbb{R}^d$. Then,*

$$\begin{aligned} \mathbb{E}[Y | \mathcal{D}] &= \mathbf{x}^\top \mathcal{T} \left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\alpha} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{Y}_n \right) \\ \text{and } \mathbb{V}[Y | \mathcal{D}] &= \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2. \end{aligned}$$

B. Proof of the Lower Bounds

Our task reduces to establishing lower bounds on

$$\mathbb{E} \left[(\mathbf{x}^\top \mathcal{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}))^2 \right] \tag{10}$$

for any choice of estimator $\hat{\boldsymbol{\alpha}}$, which in combination with Lemma 4.5 will prove Theorem 4.1. In the next section we first prove a lower bound for any unbiased estimator relying on the Cramér-Rao inequality. In what follows, in Appendix B.2, we will show a general bound in Lemma B.4 valid for any estimator (possibly biased) using a *hypothesis testing* technique (see, e.g. (Lattimore & Szepesvári, 2018, Chap. 13)). Finally, in Lemma B.5 we prove a high-probability lower bound on Eq. (10).

B.1. Lower Bound for Unbiased Estimator $\hat{\alpha}$

Theorem B.1 (Cramér-Rao inequality). *Suppose that $\alpha \in \mathbb{R}^d$ is an unknown deterministic parameter with a probability density function $f(x | \alpha)$ and that $\hat{\alpha}$ is an unbiased estimator of α . Moreover assume that for all $i, j \in [d]$, $x : f(x | \alpha) > 0$, $\frac{\partial^2}{\partial \alpha_i \partial \alpha_j} \ln f(x | \alpha)$ exists and is finite, and $\frac{\partial^2}{\partial \alpha_i \partial \alpha_j} \int \hat{\alpha} f(x | \alpha) dx = \int \hat{\alpha} \left(\frac{\partial^2}{\partial \alpha_i \partial \alpha_j} f(x | \alpha) \right) dx$.*

Then, for the Fisher information matrix defined as

$$\mathbf{F} = -\mathbb{E} \left[\nabla_{\alpha} \ln f(X | \alpha) \nabla_{\alpha} \ln f(X | \alpha)^{\top} \right]$$

we have

$$\mathbb{E} \left[(\hat{\alpha} - \mathbb{E}[\hat{\alpha}])(\hat{\alpha} - \mathbb{E}[\hat{\alpha}])^{\top} \right] \succeq \mathbf{F}^{-1}.$$

Lemma B.2. *For any unbiased estimator $\hat{\alpha}$ of α in Eq. (2) we have*

$$\mathbb{E} \left[(\mathbf{x}^{\top} \mathcal{T} \Sigma^{-1} (\alpha - \hat{\alpha}))^2 \right] \geq \mathbf{x}^{\top} \mathcal{T} \Sigma^{-1} (\Psi^{\top} \mathbf{K} \Psi)^{-1} \Sigma^{-1} \mathcal{T} \mathbf{x}. \quad (11)$$

Proof. Recall that according to the equivalence (2) $\mathbf{Y} \sim \mathcal{N}(\Psi \alpha, \mathbf{K})$ and the unknown parameter is α . To compute the Fisher information matrix we first observe that

$$\nabla_{\alpha} \ln p^{\mathbf{G}}(\mathbf{Y}; \Psi \alpha, \mathbf{K}) = \Psi^{\top} \mathbf{K}^{-1} (\mathbf{Y} - \Psi \alpha)$$

and so

$$\begin{aligned} \mathbf{F} &= \mathbb{E} \left[\nabla_{\alpha} \ln p^{\mathbf{G}}(\mathbf{Y}; \Psi \alpha, \mathbf{K}) \nabla_{\alpha} \ln p^{\mathbf{G}}(\mathbf{Y}; \Psi \alpha, \mathbf{K})^{\top} \right] \\ &= \Psi^{\top} \mathbf{K}^{-1} \mathbb{E} \left[(\mathbf{Y} - \Psi \alpha)(\mathbf{Y} - \Psi \alpha)^{\top} \right] \mathbf{K}^{-1} \Psi \\ &= \Psi^{\top} \mathbf{K}^{-1} \Psi. \end{aligned}$$

Thus, by Theorem B.1 we have

$$\mathbb{E} \left[(\alpha - \hat{\alpha})(\alpha - \hat{\alpha})^{\top} \right] \succeq (\Psi^{\top} \mathbf{K}^{-1} \Psi)^{-1}.$$

Finally, left-multiplying by $\mathbf{x}^{\top} \mathcal{T} \Sigma^{-1}$ and right-multiplying the above by $\Sigma^{-1} \mathcal{T} \mathbf{x}$ gives us the statement. \square

B.2. Lower Bound for Any Estimator $\hat{\alpha}$

The proof of is based on the following lemma.

Lemma B.3 (Bretagnolle & Huber 1979). *Let P and Q be probability measures on the same measurable space (Ω, \mathcal{F}) , and let $A \in \mathcal{F}$ be an arbitrary event. Then,*

$$P(A) + Q(A^c) \geq \frac{1}{2} \exp(-D_{\text{KL}}(P, Q)), \quad (12)$$

where $D_{\text{KL}}(P, Q) = \int_{\Omega} \ln(P(\omega)/Q(\omega)) dP(\omega)$ denotes Kullback-Leibler divergence between P and Q and $A^c = \Omega \setminus A$ is the complement of A .

Lemma B.4. *For any estimator $\hat{\alpha}$ of α in Eq. (2) we have*

$$\mathbb{E} \left[(\mathbf{x}^{\top} \mathcal{T} \Sigma^{-1} (\hat{\alpha} - \alpha))^2 \right] \geq \frac{\mathbf{x}^{\top} \mathbf{M} \mathbf{x}}{16\sqrt{e}}.$$

Proof. Throughout the proof let $\mathbf{q} = \Sigma^{-1} \mathcal{T} \mathbf{x}$. Consider two meta-learning problems with target distributions \mathbb{P} and \mathbb{Q} characterized by two means: $\alpha_{\mathbb{P}} = \mathbf{0}$ and $\alpha_{\mathbb{Q}} = \Delta (\Psi^{\top} \mathbf{K}^{-1} \Psi)^{-1} \mathbf{q}$ where $\Delta > 0$ is a free parameter to be tuned later on. Thus, according to our established equivalence (2), in these two cases targets are generated by respective models $\mathbb{P} = \mathcal{N}(\mathbf{0}, \mathbf{K})$ and $\mathbb{Q} = \mathcal{N}(\Delta \Psi (\Psi^{\top} \mathbf{K}^{-1} \Psi)^{-1} \mathbf{q}, \mathbf{K})$.

Recall our abbreviation $\mathbf{M} = \mathcal{T} \Sigma^{-1} (\Psi^{\top} \mathbf{K}^{-1} \Psi)^{-1} \Sigma^{-1} \mathcal{T}$. Markov's inequality gives

$$\begin{aligned} \mathbb{E}_{\mathbb{P}} \left[(\hat{\alpha}^{\top} \mathbf{q} - \alpha_{\mathbb{P}}^{\top} \mathbf{q})^2 \right] &= \mathbb{E}_{\mathbb{P}} \left[(\hat{\alpha}^{\top} \mathbf{q})^2 \right] \geq \frac{\Delta^2}{4} (\mathbf{x}^{\top} \mathbf{M} \mathbf{x})^2 \mathbb{P} \left(|\hat{\alpha}^{\top} \mathbf{q}| \geq \frac{\Delta}{2} \mathbf{x}^{\top} \mathbf{M} \mathbf{x} \right), \quad \text{while} \\ \mathbb{E}_{\mathbb{Q}} \left[(\hat{\alpha}^{\top} \mathbf{q} - \alpha_{\mathbb{Q}}^{\top} \mathbf{q})^2 \right] &\geq \frac{\Delta^2}{4} (\mathbf{x}^{\top} \mathbf{M} \mathbf{x})^2 \mathbb{Q} \left(|\alpha_{\mathbb{Q}}^{\top} \mathbf{q} - \hat{\alpha}^{\top} \mathbf{q}| \geq \frac{\Delta}{2} \mathbf{x}^{\top} \mathbf{M} \mathbf{x} \right) \\ &\geq \frac{\Delta^2}{4} (\mathbf{x}^{\top} \mathbf{M} \mathbf{x})^2 \mathbb{Q} \left(|\hat{\alpha}^{\top} \mathbf{q}| < \frac{\Delta}{2} \mathbf{x}^{\top} \mathbf{M} \mathbf{x} \right), \end{aligned}$$

where the last inequality comes using the fact that $|a - b| \geq |a| - |b|$ for $a, b \in \mathbb{R}$ and observing that $\alpha_{\mathbb{Q}}^\top \mathbf{q} = \mathbf{x}^\top \mathbf{M} \mathbf{x}$. Summing both inequalities above and applying Lemma B.3 we get

$$\begin{aligned} \mathbb{E}_{\mathbb{P}} [(\hat{\alpha}^\top \mathbf{q} - \alpha_{\mathbb{P}}^\top \mathbf{q})^2] + \mathbb{E}_{\mathbb{Q}} [(\hat{\alpha}^\top \mathbf{q} - \alpha_{\mathbb{Q}}^\top \mathbf{q})^2] &\geq \frac{\Delta^2}{8} (\mathbf{x}^\top \mathbf{M} \mathbf{x})^2 \cdot \exp(-D_{\text{KL}}(\mathbb{P}, \mathbb{Q})) \\ &\stackrel{(a)}{=} \frac{\Delta^2}{8} (\mathbf{x}^\top \mathbf{M} \mathbf{x})^2 \cdot \exp\left(-\frac{\Delta^2}{2} \mathbf{x}^\top \mathbf{M} \mathbf{x}\right), \end{aligned}$$

where step (a) follows from KL-divergence between multivariate Gaussians with the same covariance matrix. Now, using a basic fact that $2 \max\{a, b\} \geq a + b$, we get that for any measure \mathbb{P} given by parameter α we have

$$\mathbb{E} [(\hat{\alpha}^\top \mathbf{q} - \alpha^\top \mathbf{q})^2] \geq \frac{\Delta^2}{16} (\mathbf{x}^\top \mathbf{M} \mathbf{x})^2 \cdot \exp\left(-\frac{\Delta^2}{2} \mathbf{x}^\top \mathbf{M} \mathbf{x}\right).$$

The statement then follows by choosing $\Delta^2 = (\mathbf{x}^\top \mathbf{M} \mathbf{x})^{-1}$. □

Now we prove a high-probability version of the just given inequality.

Lemma B.5. *For any estimator $\hat{\alpha}$ of α in Eq. (2) and any $\delta \in (0, 1)$ we have*

$$\mathbb{P}\left((\mathbf{x}^\top \mathcal{T} \Sigma^{-1} (\hat{\alpha} - \alpha))^2 \geq \ln\left(\frac{1}{4} \cdot \frac{1}{1 - \delta}\right) \mathbf{x}^\top \mathbf{M} \mathbf{x}\right) \geq 1 - \delta.$$

Proof. The proof is very similar to the proof of Lemma B.4 except we will not apply Markov's inequality and focus directly on giving a lower bound the deviation probabilities rather than expectations. Thus, similarly as before introduce mean parameters $\alpha_{\mathbb{P}} = \mathbf{0}$ and $\alpha_{\mathbb{Q}} = \Delta(\Psi^\top \mathbf{K}^{-1} \Psi)^{-1} \mathbf{q} / (\mathbf{x}^\top \mathbf{M} \mathbf{x})$ and their associated probability measures $\mathbb{P} = \mathcal{N}(\mathbf{0}, \mathbf{K})$ and $\mathbb{Q} = \mathcal{N}\left(\frac{\Delta \Psi (\Psi^\top \mathbf{K}^{-1} \Psi)^{-1} \mathbf{q}}{\mathbf{x}^\top \mathbf{M} \mathbf{x}}, \mathbf{K}\right)$.

Note that

$$\begin{aligned} \mathbb{P}\left(|\hat{\alpha}^\top \mathbf{q}| \geq \frac{\Delta}{2}\right) &= \mathbb{P}\left(|\alpha_{\mathbb{P}}^\top \mathbf{q} - \hat{\alpha}^\top \mathbf{q}| \geq \frac{\Delta}{2}\right), \\ \mathbb{Q}\left(|\alpha_{\mathbb{Q}}^\top \mathbf{q} - \hat{\alpha}^\top \mathbf{q}| \geq \frac{\Delta}{2}\right) &\geq \mathbb{Q}\left(|\hat{\alpha}^\top \mathbf{q}| < \frac{\Delta}{2}\right) \end{aligned}$$

and so by using Lemma B.3 we obtain an exponential tail bound

$$\mathbb{P}\left(|\alpha_{\mathbb{P}}^\top \mathbf{q} - \hat{\alpha}^\top \mathbf{q}| \geq \frac{\Delta}{2}\right) + \mathbb{Q}\left(|\alpha_{\mathbb{Q}}^\top \mathbf{q} - \hat{\alpha}^\top \mathbf{q}| \geq \frac{\Delta}{2}\right) \geq \exp(-D_{\text{KL}}(\mathbb{P} \parallel \mathbb{Q})) = \frac{1}{2} \exp\left(-\frac{\Delta^2}{\mathbf{x}^\top \mathbf{M} \mathbf{x}}\right).$$

Setting the r.h.s. in the above to $2(1 - \delta)$ where δ is an error probability, and solving for Δ gives us tuning

$$\Delta^2 = 2 \ln\left(\frac{1}{4} \cdot \frac{1}{1 - \delta}\right) \mathbf{x}^\top \mathbf{M} \mathbf{x}.$$

Thus, we get

$$\mathbb{P}\left((\alpha_{\mathbb{P}}^\top \mathbf{q} - \hat{\alpha}^\top \mathbf{q})^2 \geq \frac{1}{2} \ln\left(\frac{1}{4} \cdot \frac{1}{1 - \delta}\right) \mathbf{x}^\top \mathbf{M} \mathbf{x}\right) + \mathbb{Q}\left((\alpha_{\mathbb{Q}}^\top \mathbf{q} - \hat{\alpha}^\top \mathbf{q})^2 \geq \frac{1}{2} \ln\left(\frac{1}{4} \cdot \frac{1}{1 - \delta}\right) \mathbf{x}^\top \mathbf{M} \mathbf{x}\right) \geq 2(1 - \delta)$$

and using the fact that $2 \max(a, b) \geq a + b$ we get that for any probability measure \mathbb{P} given by parameter α we have

$$\mathbb{P}\left((\alpha_{\mathbb{P}}^\top \mathbf{q} - \hat{\alpha}^\top \mathbf{q})^2 \geq \frac{1}{2} \ln\left(\frac{1}{4} \cdot \frac{1}{1 - \delta}\right) \mathbf{x}^\top \mathbf{M} \mathbf{x}\right) \geq 1 - \delta.$$

□

C. Proof of the Upper Bounds

Theorem 4.2 (restated). For the estimator $\hat{\boldsymbol{\theta}}_n(\hat{\boldsymbol{\alpha}}^{\text{MLE}})$ and for any $\mathbf{x} \in \mathbb{R}^d$ we have

$$\mathbb{E}[\mathcal{L}(\mathbf{x})] = \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2.$$

Moreover for the same estimator, with probability at least $1 - \delta$, $\delta \in (0, 1)$ we have

$$\mathcal{L}(\mathbf{x}) \leq 2 \ln \left(\frac{2}{\delta} \right) \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2.$$

Proof. Recall that

$$\hat{\boldsymbol{\alpha}}^{\text{MLE}} = (\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \mathbf{K}^{-1} \mathbf{Y}.$$

The first result follows from Lemma 4.5 where we have to give an identity for

$$\mathbb{E} \left[(\mathbf{x}^\top \mathcal{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}^{\text{MLE}}))^2 \right] \quad (13)$$

and the missing piece is a covariance of the estimator $\hat{\boldsymbol{\alpha}}^{\text{MLE}}$

$$\begin{aligned} & \mathbb{E} [(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}^{\text{MLE}})(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}^{\text{MLE}})^\top] \\ &= (\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \mathbf{K}^{-1} \text{Cov}(\mathbf{Y}, \mathbf{Y}) \mathbf{K}^{-1} \boldsymbol{\Psi} (\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi})^{-1} \\ &= (\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi})^{-1}. \end{aligned} \quad (14)$$

To prove the second result we have to give a high probability upper bound on Eq. (13).

Let $\mathbf{q} = \boldsymbol{\Sigma}^{-1} \mathcal{T} \mathbf{x}$ and observe that $\mathbf{q}^\top \hat{\boldsymbol{\alpha}}^{\text{MLE}}$ is Gaussian (since \mathbf{Y} is composed of Gaussian entries) with mean $\mathbf{q}^\top \boldsymbol{\alpha}$ by equivalence (2), and covariance $(\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi})^{-1}$ by Eq. (14). Then, by Gaussian concentration for any error probability $\delta \in (0, 1)$ we have

$$\mathbb{P} \left(\left| \mathbf{q}^\top \boldsymbol{\alpha} - \mathbf{q}^\top \hat{\boldsymbol{\alpha}}^{\text{MLE}} \right|^2 \geq 2 \mathbf{q}^\top (\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi})^{-1} \mathbf{q} \ln \left(\frac{2}{\delta} \right) \right) \leq \delta$$

which completes the proof. \square

D. Derivation of EM Steps

Recall that our goal is to solve

$$\max_{\boldsymbol{\theta}'} \int \ln(p(\boldsymbol{\theta}, \mathcal{D} | \mathcal{E}')) dp(\boldsymbol{\theta} | \mathcal{D}, \hat{\mathcal{E}}_t).$$

First, we will focus on the integral. The chain rule readily gives

$$\ln p(\boldsymbol{\Theta}, \mathcal{D} | \mathcal{E}') = \ln p(\boldsymbol{\Theta} | \mathcal{D}, \mathcal{E}') + \ln p(\boldsymbol{\Theta} | \mathcal{E}').$$

Using the same reasoning and notation as in the proof of Proposition A.1 we get

$$\begin{aligned} \int \ln p(\boldsymbol{\theta} | \mathcal{D}, \mathcal{E}') dp(\boldsymbol{\theta} | \mathcal{D}, \hat{\mathcal{E}}_t) &= \sum_{i=1}^n \sum_{j=1}^{m_i} \left(\frac{1}{2} \ln \left(\frac{1}{\sigma^2} \right) - \frac{1}{2\sigma^2} \int (Y_{i,j} - \mathbf{x}_{i,j}^\top \boldsymbol{\theta}_i)^2 dp(\boldsymbol{\theta}_i | \mathcal{D}, \hat{\mathcal{E}}_t) \right) + \text{const}(\mathcal{E}') \\ &= \sum_{i=1}^n \sum_{j=1}^{m_i} \left(\frac{1}{2} \ln \left(\frac{1}{\sigma^2} \right) - \frac{1}{2\sigma^2} (Y_{i,j} - \mathbf{x}_{i,j}^\top \boldsymbol{\mu}_i)^2 - \mathbf{x}_{i,j}^\top \mathcal{T}_i \mathbf{x}_{i,j} \right) + \text{const}(\mathcal{E}'), \end{aligned}$$

using the fact that $\int (Y_{i,j} - \mathbf{x}_{i,j}^\top \boldsymbol{\theta}_i)^2 dp(\boldsymbol{\theta}_i | \mathcal{D}, \hat{\mathcal{E}}_t) = (Y_{i,j} - \mathbf{x}_{i,j}^\top \boldsymbol{\mu}_i)^2 + \mathbf{x}_{i,j}^\top \mathcal{T}_i \mathbf{x}_{i,j}$ where we took $\boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathcal{T}_i)$ according to Proposition A.1.

Now we compute the expected log-likelihood of the vector of task parameters:

$$\int \ln p(\boldsymbol{\theta} | \mathcal{E}') dp(\boldsymbol{\theta} | \mathcal{D}, \hat{\mathcal{E}}_t) = \frac{n}{2} \ln \det \boldsymbol{\Sigma}^{-1} - \frac{1}{2} \sum_{i=1}^n \int (\boldsymbol{\theta}_i - \boldsymbol{\alpha})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta}_i - \boldsymbol{\alpha}) dp(\boldsymbol{\theta}_i | \mathcal{D}, \hat{\mathcal{E}}_t) + \text{const}(\mathcal{E}').$$

M-step for σ^2 . Now, note that since the likelihood of the vector of task variables Θ does not depend on the parameter σ^2 we can solve for σ^2 based on the first order condition of the problem above. Differentiating the above equation with respect to σ^{-2} (and ignoring the constant) gives

$$\sum_{i=1}^n \sum_{j=1}^{m_i} (\sigma^2 - ((Y_{i,j} - \mathbf{x}_{i,j}^\top \boldsymbol{\mu}_i)^2 + \mathbf{x}_{i,j}^\top \mathcal{T}_i \mathbf{x}_{i,j})). \quad (15)$$

while setting the derivative to zero gives

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n \frac{1}{m_i} \sum_{j=1}^{m_i} ((Y_{i,j} - \mathbf{x}_{i,j}^\top \boldsymbol{\mu}_i)^2 + \mathbf{x}_{i,j}^\top \mathcal{T}_i \mathbf{x}_{i,j}). \quad (16)$$

M-step for α . Differentiating the objective w.r.t. α (and ignoring the constant) gives $\sum_{i=1}^n \Sigma^{-1} (\mathbb{E}[\theta_i] - \alpha)$ from which we get

$$\alpha = \sum_{i=1}^n \boldsymbol{\mu}_i. \quad (17)$$

M-step for Σ . Differentiating the expected log-likelihood of the vector of task parameters with respect to $\mathbf{A} = \Sigma^{-1}$ gives

$$\sum_{i=1}^n \text{tr}(\Sigma d\mathbf{A}) - \text{tr} \int ((\boldsymbol{\vartheta}_i - \alpha)(\boldsymbol{\vartheta}_i - \alpha)^\top d\mathbf{A}) dp(\boldsymbol{\vartheta}_i | \mathcal{D}, \hat{\boldsymbol{\epsilon}}_t) \quad (18)$$

from which we get

$$\Sigma = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[(\boldsymbol{\theta}_i - \alpha)(\boldsymbol{\theta}_i - \alpha)^\top]. \quad (19)$$

Finally, computing the expectation

$$\sum_{i=1}^n (\mathbb{E}[\boldsymbol{\theta}_i \boldsymbol{\theta}_i^\top] - 2\boldsymbol{\mu}_i \alpha^\top + \alpha \alpha^\top) = \sum_{i=1}^n ((\boldsymbol{\mu}_i - \alpha)(\boldsymbol{\mu}_i - \alpha)^\top + \mathcal{T}_i) \quad (20)$$

shows the update for Σ .

E. Selecting λ in Biased Regression

The parameter λ is selected via random search in the following way. For each of the 50 samples of λ from log-uniform distribution on interval $[0; 100]$ we perform the following procedure to estimate the risk \hat{L} . Firstly, we split the training tasks into $K = 10$ groups $\mathcal{S}_1, \dots, \mathcal{S}_K$ of (approximately) equal size and compute the estimates $\hat{\alpha}_k$ using the data $\mathcal{S}^{\setminus k}$ from all of the groups excluding the group k : $\mathcal{S}^{\setminus k} := \cup_{i \neq k} \mathcal{S}_i$. For each of the estimated values $\hat{\alpha}_k$ we perform adaptation to and testing on the tasks in the group \mathcal{S}_k using the given value of λ . We split the samples of each task data $D_i \in \mathcal{S}_k$ randomly into adaptation and test sets 10 times each time such that the size of adaptation set is close to the size of adaptation sets used with the actual test data. For each of the splits we compute an estimate of the parameter vector $\hat{\boldsymbol{\theta}}_{k,i,l}$ where k is the index of the group which was not used to estimate $\hat{\alpha}_k$, i is the index of a task data $D_i \in \mathcal{S}_k$, l is the index of a random split of the samples in that task into adaptation and test sets. With this parameter vector and using the test set of the task $D_i \in \mathcal{S}_k$ we can also estimate the loss $\hat{L}_{k,i,l}$ after which all the loss values are averaged:

$$\hat{L} = \frac{1}{K} \sum_{k=1}^K \frac{1}{|\mathcal{S}^{\setminus k}|} \sum_{i: D_i \in \mathcal{S}^{\setminus k}} \frac{1}{10} \sum_{l=1}^{10} \hat{L}_{k,i,l}.$$

At the end we select the value of λ which lead to the smallest value of \hat{L} using this cross-validation procedure.

F. Supplementary Statements

Proposition F.1. For M (see Eq. (5)) we have

$$M = \sigma^4 \cdot (\Sigma \mathbf{X}_n^\top \mathbf{X}_n + \sigma^2 \mathbf{I})^{-1} \mathbf{A}^{-1} (\Sigma \mathbf{X}_n^\top \mathbf{X}_n + \sigma^2 \mathbf{I})^{-1},$$

where we denote

$$\mathbf{A} = \sum_{i=1}^n \mathbf{X}_i^\top (\mathbf{X}_i \boldsymbol{\Sigma} \mathbf{X}_i^\top + \sigma^2 \mathbf{I})^{-1} \mathbf{X}_i.$$

Proof. Recall that

$$\mathbf{M} = \mathcal{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi})^{-1} \boldsymbol{\Sigma}^{-1} \mathcal{T}$$

and observe that

$$\mathbf{K}^{-1} = \begin{bmatrix} (\mathbf{X}_1 \boldsymbol{\Sigma} \mathbf{X}_1^\top + \sigma^2 \mathbf{I})^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & (\mathbf{X}_2 \boldsymbol{\Sigma} \mathbf{X}_2^\top + \sigma^2 \mathbf{I})^{-1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & (\mathbf{X}_n \boldsymbol{\Sigma} \mathbf{X}_n^\top + \sigma^2 \mathbf{I})^{-1} \end{bmatrix},$$

which in turn implies

$$\boldsymbol{\Psi}^\top \mathbf{K}^{-1} \boldsymbol{\Psi} = \sum_{i=1}^n \mathbf{X}_i^\top (\mathbf{X}_i \boldsymbol{\Sigma} \mathbf{X}_i^\top + \sigma^2 \mathbf{I})^{-1} \mathbf{X}_i.$$

On the other hand,

$$\begin{aligned} \mathcal{T} \boldsymbol{\Sigma}^{-1} &= \left(\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{X}_n \right)^{-1} \boldsymbol{\Sigma}^{-1} \\ &= \sigma^2 (\sigma^2 \mathbf{I} + \boldsymbol{\Sigma} \mathbf{X}_n^\top \mathbf{X}_n)^{-1}. \end{aligned}$$

Combining the above gives the statement. \square

Lemma F.2. In the following assume that $\mathbf{X}_i^\top \mathbf{X}_i = \frac{m_i}{d} \mathbf{I}$ for all i . Let $\lambda_j(\boldsymbol{\Sigma})$ be the j th eigenvalue of $\boldsymbol{\Sigma}$. Then,

$$\lambda_j(\mathbf{M}) = \sigma^4 \cdot \frac{d^2}{(m_n \lambda_j(\boldsymbol{\Sigma}) + d\sigma^2)^2} \cdot \frac{HM\left(\lambda_j(\boldsymbol{\Sigma}) + \frac{d\sigma^2}{m_i}\right)_{i=1}^n}{n},$$

where $HM(z_i)_{i=1}^n$ denotes the harmonic mean of sequence $(z_i)_{i=1}^n$. Moreover,

$$\lambda_j(\mathcal{T}) = \frac{d\sigma^2 \lambda_j(\boldsymbol{\Sigma})}{d\sigma^2 + m_n \lambda_j(\boldsymbol{\Sigma})}.$$

Finally, the eigenvectors of \mathbf{M} and \mathcal{T} coincide with the eigenvectors of $\boldsymbol{\Sigma}$.

Proof. We first characterize eigenvalues of matrix \mathbf{M} . By Proposition F.1,

$$\mathbf{M} = \sigma^4 \cdot (\boldsymbol{\Sigma} \mathbf{X}_n^\top \mathbf{X}_n + \sigma^2 \mathbf{I})^{-1} \mathbf{A}^{-1} (\boldsymbol{\Sigma} \mathbf{X}_n^\top \mathbf{X}_n + \sigma^2 \mathbf{I})^{-1}.$$

We start with \mathbf{A}^{-1} , and by the spectral theorem, $\boldsymbol{\Sigma} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top$ for some unitary \mathbf{U} and diagonal $\boldsymbol{\Lambda}$:

$$\begin{aligned} \mathbf{A}^{-1} &= \left(\sum_{i=1}^n \mathbf{X}_i^\top (\mathbf{X}_i \boldsymbol{\Sigma} \mathbf{X}_i^\top + \sigma^2 \mathbf{I})^{-1} \mathbf{X}_i \right)^{-1} = \left(\sum_{i=1}^n (\boldsymbol{\Sigma} \mathbf{X}_i^\top \mathbf{X}_i + \sigma^2 \mathbf{I})^{-1} \mathbf{X}_i^\top \mathbf{X}_i \right)^{-1} \\ &= \left(\sum_{i=1}^n \left(\boldsymbol{\Sigma} \cdot \frac{m_i}{d} + \sigma^2 \mathbf{I} \right)^{-1} \frac{m_i}{d} \right)^{-1} \\ &= \left(\sum_{i=1}^n \left(\mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top \cdot \frac{m_i}{d} + \sigma^2 \mathbf{I} \right)^{-1} \frac{m_i}{d} \right)^{-1} \\ &= \mathbf{U} \left(\sum_{i=1}^n \left(\boldsymbol{\Lambda} + \frac{d\sigma^2}{m_i} \cdot \mathbf{I} \right)^{-1} \right)^{-1} \mathbf{U}^\top. \end{aligned}$$

Now,

$$\begin{aligned} (\boldsymbol{\Sigma} \mathbf{X}_n^\top \mathbf{X}_n + \sigma^2 \mathbf{I})^{-1} &= \left(\boldsymbol{\Sigma} \cdot \frac{m_n}{d} + \sigma^2 \mathbf{I} \right)^{-1} \\ &= \left(\mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top \cdot \frac{m_n}{d} + \sigma^2 \mathbf{I} \right)^{-1} \\ &= \mathbf{U} \left(\boldsymbol{\Lambda} \cdot \frac{m_n}{d} + \sigma^2 \mathbf{I} \right)^{-1} \mathbf{U}^\top. \end{aligned}$$

Thus,

$$\mathbf{M} = \mathbf{U} \left(\left(\boldsymbol{\Lambda} \cdot \frac{m_n}{d} + \sigma^2 \mathbf{I} \right)^2 \sum_{i=1}^n \left(\boldsymbol{\Lambda} + \frac{d\sigma^2}{m_i} \right)^{-1} \right)^{-1} \mathbf{U}^\top.$$

and moreover the j th eigenvalue of \mathbf{M} is

$$\begin{aligned} \lambda_j(\mathbf{M}) &= \frac{1}{\left(\frac{m_n}{d} \lambda_j(\boldsymbol{\Sigma}) + \sigma^2 \right)^2} \cdot \frac{1}{\sum_{i=1}^n \frac{1}{\lambda_j(\boldsymbol{\Sigma}) + \frac{d\sigma^2}{m_i}}} \\ &= \frac{1}{\left(\frac{m_n}{d} \lambda_j(\boldsymbol{\Sigma}) + \sigma^2 \right)^2} \cdot \frac{\text{HM} \left(\lambda_j(\boldsymbol{\Sigma}) + \frac{d\sigma^2}{m_i} \right)_{i=1}^n}{n}. \end{aligned}$$

where recall that $\text{HM}(z_i)_{i=1}^n$ denotes the harmonic mean of sequence $(z_i)_{i=1}^n$.

Using the same arguments as above

$$\begin{aligned} \mathcal{T} &= \left(\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2} \mathbf{X}_n^\top \mathbf{X}_n \right)^{-1} \\ &= \left(\mathbf{U} \boldsymbol{\Lambda}^{-1} \mathbf{U}^\top + \frac{m_n}{d\sigma^2} \right)^{-1} \end{aligned}$$

and so

$$\lambda_j(\mathcal{T}) = \frac{1}{\frac{1}{\lambda_j(\boldsymbol{\Sigma})} + \frac{m_n}{d\sigma^2}} = \frac{d\sigma^2 \lambda_j(\boldsymbol{\Sigma})}{d\sigma^2 + m_n \lambda_j(\boldsymbol{\Sigma})}.$$

Finally, in both cases of \mathbf{M} and \mathcal{T} we observe that their eigenvectors are eigenvectors of $\boldsymbol{\Sigma}$. \square

Corollaries 4.3 and 4.4 (restated). *In the following assume that $\mathbf{X}_i^\top \mathbf{X}_i = \frac{m_i}{d} \mathbf{I}$ for all i . For $\boldsymbol{\Sigma} = \tau^2 \mathbf{I}$, any $\mathbf{x} \in \mathbb{R}^d$, and any $c > 0$,*

$$c \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2 = c \cdot \frac{H_{\tau^2}}{n} \cdot \frac{d^2 \sigma^4}{(\tau^2 m_n + d\sigma^2)^2} \cdot \|\mathbf{x}\|^2 + \frac{d\sigma^2 \tau^2}{\tau^2 m_n + d\sigma^2} \cdot \|\mathbf{x}\|^2 + \sigma^2,$$

where H_{τ^2} is a harmonic mean of the sequence $\left(\tau^2 + \frac{d\sigma^2}{m_i} \right)_{i=1}^n$.

Moreover, let $\boldsymbol{\Sigma}$ be a PSD matrix of rank $s \leq d$ with eigenvalues $\lambda_1 \geq \dots \geq \lambda_s > 0$. Then for any $\mathbf{x} \in \mathbb{R}^d$ and any $c > 0$,

$$c \mathbf{x}^\top \mathbf{M} \mathbf{x} + \mathbf{x}^\top \mathcal{T} \mathbf{x} + \sigma^2 \geq c \cdot \frac{H_{\lambda_s}}{n} \cdot \frac{d^2 \sigma^4}{(\lambda_1 m_n + d\sigma^2)^2} \cdot \|\mathbf{x}\|_{\mathbf{P}_s^\top \mathbf{P}_s}^2 + \frac{d\sigma^2 \lambda_s}{\lambda_s m_n + d\sigma^2} \cdot \|\mathbf{x}\|_{\mathbf{P}_s^\top \mathbf{P}_s}^2 + \sigma^2$$

where $\mathbf{P}_s = [\mathbf{u}_1, \dots, \mathbf{u}_s]^\top$ and $(\mathbf{u}_j)_{j=1}^s$ are eigenvectors of $\boldsymbol{\Sigma}$.

Proof. Recalling that by Proposition F.1,

$$\mathbf{M} = \sigma^4 \cdot (\boldsymbol{\Sigma} \mathbf{X}_n^\top \mathbf{X}_n + \sigma^2 \mathbf{I})^{-1} \mathbf{A}^{-1} (\boldsymbol{\Sigma} \mathbf{X}_n^\top \mathbf{X}_n + \sigma^2 \mathbf{I})^{-1}.$$

and using Lemma F.2 with $\boldsymbol{\Sigma} = \tau^2 \mathbf{I}$ we get the first result.

Now we turn to the low-rank case. We start by considering a PSD matrix Σ_ε with s eigenvalues $\lambda_1 \geq \dots \geq \lambda_s > 0$ and remaining $d - s$ are $\varepsilon > 0$. Denote also by $M_\varepsilon, \mathcal{T}_\varepsilon$ matrices w.r.t. Σ_ε . The idea is to lower bound $\mathbf{x}^\top M_\varepsilon \mathbf{x}$ and $\mathbf{x}^\top \mathcal{T}_\varepsilon \mathbf{x}$ and then analyze a limiting behavior as $\varepsilon \rightarrow 0$.

By Lemma F.2, $M_\varepsilon, \mathcal{T}_\varepsilon$, and Σ_ε share the same eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_s$, and so

$$\begin{aligned} c\mathbf{x}^\top M_\varepsilon \mathbf{x} + \mathbf{x}^\top \mathcal{T}_\varepsilon \mathbf{x} &= c \sum_{j=1}^d (\mathbf{u}_j^\top \mathbf{x})^2 \lambda_j(M_\varepsilon) + \sum_{j=1}^d (\mathbf{u}_j^\top \mathbf{x})^2 \lambda_j(\mathcal{T}_\varepsilon) \\ &= c \cdot \sum_{j=1}^s \frac{H\lambda_j}{n} \cdot \frac{\sigma^4}{(\lambda_j \frac{m_n}{d} + \sigma^2)^2} (\mathbf{u}_j^\top \mathbf{x})^2 + c \cdot \underbrace{\frac{H\varepsilon}{n} \cdot \frac{\sigma^4}{(\varepsilon \frac{m_n}{d} + \sigma^2)^2}}_{(a)} \left(\sum_{j=s+1}^d (\mathbf{u}_j^\top \mathbf{x})^2 \right) \\ &\quad + \sum_{j=1}^s \frac{\sigma^2 \lambda_j}{\lambda_j \frac{m_n}{d} + \sigma^2} (\mathbf{u}_j^\top \mathbf{x})^2 + \frac{\sigma^2 \varepsilon}{\varepsilon \frac{m_n}{d} + \sigma^2} \left(\sum_{j=s+1}^d (\mathbf{u}_j^\top \mathbf{x})^2 \right). \end{aligned}$$

Now,

$$\begin{aligned} &\lim_{\varepsilon \rightarrow 0} (c\mathbf{x}^\top M_\varepsilon \mathbf{x} + \mathbf{x}^\top \mathcal{T}_\varepsilon \mathbf{x}) \\ &= c \cdot \sum_{j=1}^s \frac{H\lambda_j}{n} \cdot \frac{\sigma^4}{(\lambda_j \frac{m_n}{d} + \sigma^2)^2} (\mathbf{u}_j^\top \mathbf{x})^2 + \frac{d\sigma^2}{M} \sum_{j=s+1}^d (\mathbf{u}_j^\top \mathbf{x})^2 + \sum_{j=1}^s \frac{\sigma^2 \lambda_j}{\lambda_j \frac{m_n}{d} + \sigma^2} (\mathbf{u}_j^\top \mathbf{x})^2 \\ &\geq c \cdot \frac{H\lambda_s}{n} \cdot \frac{\sigma^4}{(\lambda_1 \frac{m_n}{d} + \sigma^2)^2} \sum_{j=1}^s (\mathbf{u}_j^\top \mathbf{x})^2 + \frac{\sigma^2 \lambda_s}{\lambda_s \frac{m_n}{d} + \sigma^2} \sum_{j=1}^s (\mathbf{u}_j^\top \mathbf{x})^2, \end{aligned}$$

where we note that the limit of term (a) is handled as

$$\lim_{\varepsilon \rightarrow 0} \frac{1}{\sum_{i=1}^n \frac{1}{\varepsilon + \frac{d\sigma^2}{m_i}}} \cdot \frac{\sigma^4}{(\varepsilon \frac{m_n}{d} + \sigma^2)^2} = \frac{d\sigma^2}{M} \geq 0.$$

□

G. Further Experimental Details and Results

In this section we provide extra figures for our experimental results. Fig. 1 is complemented with Fig. 7, adding a second example in addition to the one shown in the previous figure.

For completeness, the pseudocode of MoM is given in Algorithm 2.

Algorithm 2 MoM Estimator for Learning Linear Features of (Tripuraneni et al., 2020)

Input: $((\mathbf{x}_{1,j}, y_{1,j}))_{j=1}^{m_1}, \dots, ((\mathbf{x}_{m_{n-1},j}, y_{m_{n-1},j}))_{j=1}^{m_{n-1}}$ — training examples from $n - 1$ past tasks, s — problem rank.

$$UDV^\top \leftarrow \text{SVD} \left(\frac{1}{M - m_n} \sum_{i=1}^{n-1} \sum_{j=1}^{m_n} y_{i,j}^2 \mathbf{x}_{i,j} \mathbf{x}_{i,j}^\top \right)$$

$$\hat{B} \leftarrow [D_{1,1} \mathbf{u}_1, \dots, D_{s,s} \mathbf{u}_s]$$

return \hat{B}

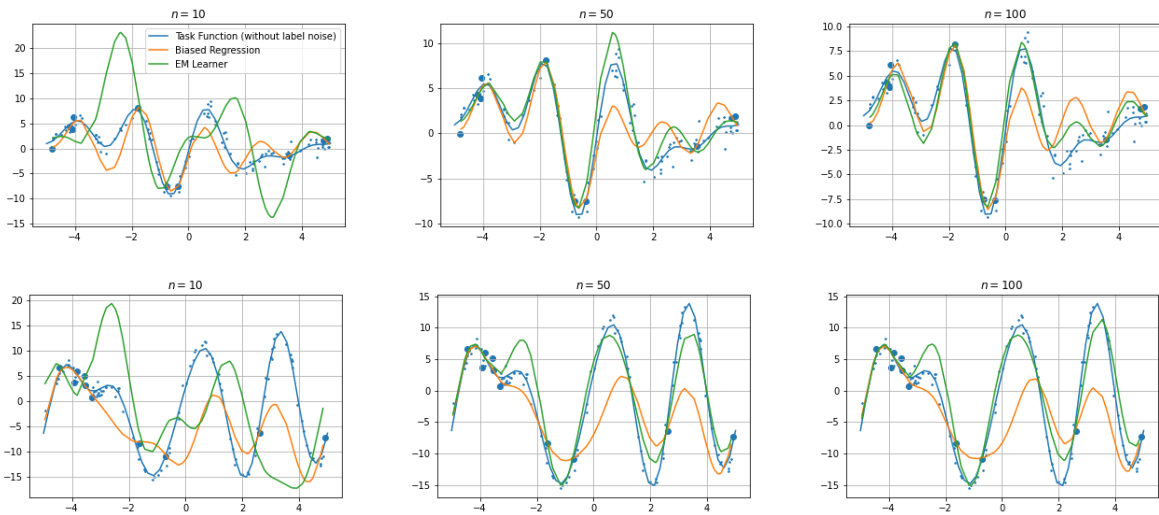


Figure 7. Two sets of examples of predictions on the synthetic, ‘Fourier’ meta-learning problem. Top and bottom rows correspond to different (random) instances; the top row in fact replicates Fig. 1. Training data is shown in bold, small dots show test data. We also show the predictions for two learners (at every input) and the target function. The column correspond to outputs obtained training on $n \in \{10, 50, 100\}$ tasks.