Towards sample-efficient overparameterized meta-learning

Abstract

Meta-learning typically involves two phases. First, one learns a suitable representation from the previously seen tasks. Secondly, this representation is used for learning a new task using only a few samples (i.e., few-shot learning). For both phases, ensuring sample efficient learning is critical. We first observe that existing works do not capture empirical phenomena, that is, typically, both representation learning as well as few-shot learning operate in the overparameterized regime where sample size is less than the degrees of freedom of the problem. To this aim, we study meta-learning with general task and feature covariances. For learning the representation, we investigate multiple spectral approaches showing that, spectral alignment of the feature and tasks covariances can help explain how linear representations can be learned with much fewer samples. Our analysis also leads to refined bounds for achieving optimal sample complexity for subspace-based linear representations. For few-shot learning, we derive the optimal linear representation minimizing the few-shot sample complexity. In contrast to subspace-based representations, this optimal representation can provably be high-dimensional (i.e., downstream task is overparameterized) explaining the empirical observations on few-shot learning in related literature. We then establish favorable robustness properties of this optimal representation to achieve end-to-end learning guarantees for two-phase meta-learning.

1. Introduction

In a multitude of machine learning (ML) tasks with limited data, it is crucial to build accurate models in a sample-efficient way; this goal proves to be especially challenging when the features are high-dimensional, as is often the case in modern ML.

Constructing a simple yet informative representation of features can be very helpful with learning a model that generalizes well to an unseen test set. Representation learning, which dates back to (Caruana, 1997; Baxter, 2000), aims to jointly utilize information from the training data of multiple related tasks. Such information transfer is the backbone of modern transfer and multitask learning and finds ubiquitous applications in image classification (Deng et al., 2009), machine translation (Bojar et al., 2014) and reinforcement learning (Finn et al., 2017), all of which may involve numerous tasks to be learned[1] with limited data per task.

In this paper, we focus on the fundamental problem of meta-learning with linear features, which consists of two steps. In the meta-training phase, we learn a linear representation from a collection of $n_{\rm task}$ earlier tasks, with $n_{\rm spt}$ training samples per task. Let $(\boldsymbol{x}_{ij}, y_{ij})$ be the jth training sample from task i and we collect training samples from all tasks in the set $\mathcal{S} = \{(\boldsymbol{x}_{ij}, y_{ij})\} \subset \mathbb{R}^d \times \mathbb{R}$. Denote $n_{\rm tot} = n_{\rm task} \times n_{\rm spt} = |\mathcal{S}|$. The goal of this stage is to output the matrix

$$\boldsymbol{\Lambda} := \boldsymbol{\Lambda}(\mathcal{S}) \in \mathbb{R}^{R \times d}, \tag{1.1}$$

that can be used to map raw input features to a linear feature map of dimension R.

In the second step (few-shot learning), we are given a new task with data $\mathcal{F} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_{\rm fs}}$, with possibly very small $n_{\rm fs}$. We use the representation $\boldsymbol{\Lambda}$ to transform the input features to obtain a generalizing model $\hat{\boldsymbol{\alpha}}_{\boldsymbol{\Lambda}}$ by minimizing the empirical risk

$$\hat{\mathcal{L}}_{\mathcal{F}}(\boldsymbol{\alpha}) = \frac{1}{n_{\rm fs}} \sum_{i=1}^{n_{\rm fs}} \ell(y_i, \boldsymbol{\alpha}^T \boldsymbol{\Lambda} \boldsymbol{x}_i). \tag{1.2}$$

Past analysis of algorithms for both the meta-training and few-shot phases have typically focused on the classical *underparameterized regime*: when the sample size is larger than the degrees of freedom (DoF) which are $\mathcal{O}(Rd)$ for (1.1) and $\mathcal{O}(R)$ for (1.2). These works are also often focused on learning simplistic representations, where $\boldsymbol{\Lambda}$ is simply a subspace (i.e. orthonormal rows).

However, recent literature in deep learning makes it clear that the high-dimensional (or overparameterized) regime is of significant interest for both problems. Deep networks are stellar representation learners despite containing many more parameters than the sample size (Finn et al., 2017). Additionally, the few-shot learning literature demonstrates that very few samples –often much fewer than the representation dimension– can be sufficient for successful adaptation to

[1] Each task corresponds to a vector in $\mathbb{R}^d$ and we learn the task vector given task data (features and labels). See Definitions 1 and 2 in Section 2 for details.

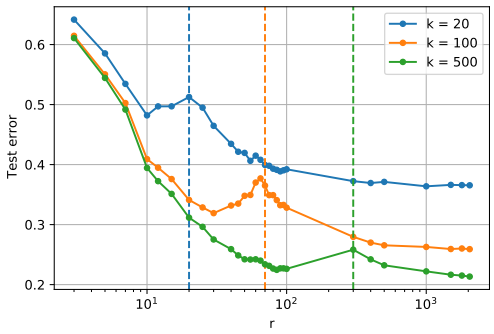


Figure 1. **An illustration of the benefit of overparameterization for few-shot learning:** A pretrained ResNet50 network on Imagenet (representation learning phase) was utilized to obtain a feature-representation for classification on CIFAR-10. After reducing the dimensionality to r via PCA, the resultant features are to train a logistic regression classifier on CIFAR-10 with k examples from each class (few-shot phase). The figure depicts the test error obtained using projected test features as a function of r. The interpolation threshold (i.e., when accurate fit for data happens) is indicated by the dashed line, and can be seen to coincide exactly with the location of the double-descent peak ($k = r$).

an unseen task. For example, (Finn et al., 2017) proposes a transfer learning method that adapts a pre-trained model to downstream tasks with only 1-5 new training samples. In Figure 1, the benefits of overparameterization and double-descent (also mentioned in (Mei & Montanari, 2019)) in the few-shot phase are observed in a transfer learning task for image classification; for all choices of k the accuracies are observed to be markedly higher in the overparameterized regime. This motivates the following fundamental questions.

Q1: What is the optimal representation $\mathbf{\Lambda} \in \mathbb{R}^{R\times d}$?

Q2: How to succeed with $n_{\rm tot} \ll Rd$ and $n_{\rm fs} \ll R$?

Contributions: Towards answering these, we make several key contributions to the finite-sample understanding of *linear* meta-learning, under assumptions discussed in Section 2. As described below, our results are enabled by studying a general data/task model with *arbitrary task covariance* $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ *and feature covariance* $\mathbf{\Sigma}_{\boldsymbol{X}}$ which allows for a rich set of observations.

• **Representation learning:** We study representation learning for linear regression tasks using established moment-based algorithms (precisely defined in Section 3.1), and formalize *how feature-task interaction governs learning*. Specializing our bounds to the case of a rank-r_t task covariance $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ and a "spiked" (approximately low-rank) feature covariance $\mathbf{\Sigma}_{\boldsymbol{X}}$ reveals that sample complexity for learning the task subspace is governed by the *spectral alignment of the task & feature covariances* and can go well below the DoF $\mathcal{O}(r_t d)$, answering **Q2**. Surprisingly, if this alignment is sufficiently strong, simple PCA estimator on the empirical feature covariance can be preferable to more sophisticated methods (see Fig. 2(a)). When specialized to the case with uninformative features ($\mathbf{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$), we show that $\mathcal{O}(r_t d)$ samples are sufficient when $n_{\rm spt} \gtrsim \mathcal{O}(r_t)$ *achieving optimal sample size* and improving over prior bounds of $\mathcal{O}(r_t^2 d)$ (Tripuraneni et al., 2020). Importantly, this leads to the critical finding that, for fixed total sample size $n_{\rm tot}$, *both* $n_{\rm spt}$ *and* $n_{\rm task}$ *should be reasonably large* for this near-optimal performance (see Fig. 2(b) for details).

• **Few-shot learning:** With access to infinite representation learning samples, what is the *optimal representation* $\mathbf{\Lambda}_{\rm opt}$? For least-squares regression, we answer this question by establishing an equivalence between the optimal linear representation and generalized ridge regression over the raw features (see Sec 3.2). This reveals that unlike simplistic subspace projections, the right representation for optimized few-shot performance can in fact be a high-dimensional map with $R \gg n_{\rm fs}$ addressing **Q1** and **Q2**. In contrast to subspace-based schemes which discard certain features, $\mathbf{\Lambda}_{\rm opt}$ shapes the feature covariance to emphasize informative features over less informative ones. In practice, $\mathbf{\Lambda}_{\rm opt}$ is unattainable due to finite sample ($n_{\rm tot} < \infty$) during the representation learning phase. We establish robustness guarantees under finite $n_{\rm tot}$ to provide an end-to-end guarantee for two phases in terms of $\mathbf{\Lambda}_{\rm opt}$. This also uncovers a sweet spot between subspace-based approaches and optimal-shaping schemes: As $n_{\rm tot}$ decreases, it becomes more preferable to use a smaller R due to inaccurate estimate of to $\mathbf{\Lambda}_{\rm opt}$. This explains why, in practice, smaller dimensional representations may be preferable (see Fig. 3(b) for details).

Notation: We describe each representation learning task using vector $\boldsymbol{\beta}_i \in \mathbb{R}^d$, data samples are $(\boldsymbol{x}_{ij}, y_{ij}) \in \mathbb{R}^d \times \mathbb{R}$, and noise $\varepsilon_{ij} \in \mathbb{R}$. The distribution of $\boldsymbol{\beta}_i$ and $\boldsymbol{x}_{ij}$ are $\mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{\beta}})$ and $\mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{X}})$. The rank of $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ and $\mathbf{\Sigma}_{\boldsymbol{X}}$ are r_t and r_f. The few-shot learning (downstream) task is $\boldsymbol{\beta}$ and samples are $(\boldsymbol{x}_i, y_i)$. There are $n_{\rm task}$ tasks with $n_{\rm spt}$ samples per task in the representation learning phase, and a few-shot learning task with $n_{\rm fs}$ samples. Denote $n_{\rm tot} = n_{\rm task} n_{\rm spt}$. Let the representation matrix be $\mathbf{\Lambda} \in \mathbb{R}^{R\times d}$. We use $a \lesssim b$ or $\tilde{\mathcal{O}}(b)$ that means there exists K that depends logarithmically on all parameters such that $a \leq Kb$.

1.1. Prior Art

A commonly studied setup is when tasks are generated from a Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{\beta}})$, and $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ is exactly low rank or contains a large portion of small eigenvalues. This is studied as mixed linear regression (Zhong et al., 2016; Li & Liang, 2018; Chen et al., 2020), which consists of multiple linear regression problems. The knowledge of the distribution of the tasks, or the subspace where the tasks lie in, enables robust and sample efficient learning for new tasks. (Lounici et al., 2011; Cavallanti et al., 2010; Maurer et al.,

2016) propose sample complexity bounds of representation learning for mixed linear regression. There are study of mixed linear regression combined with other structuressuch as binary task vectors (Balcan et al.) and sparse task vectors (Argyriou et al., 2008).

The recent papers (Kong et al., 2020b;a; Tripuraneni et al., 2020; Du et al., 2020) propose the theoretical bounds on sample complexity and estimation error of representation learning when the tasks lie in an exactly low dimensional subspace, where the first three discuss method of moment estimators and the last two discuss matrix factorized formulations. (Tripuraneni et al., 2020) shows that when features are d dimensional and the task covariance matrix is exactly rank r_t where $r_t \ll d$, the number of samples that enable meaningful representation learning is $\mathcal{O}(dr_t^2)$. However, (Kong et al., 2020b;a; Tripuraneni et al., 2020) all assume that the features follow a standard normal distribution. We study the more general setting of arbitrary feature and task variances, and we show better sample complexity is achieved when the features and tasks are aligned. We also propose another estimator and it succeeds with $\mathcal{O}(dr_t)$ samples when $n_{\text{spt}} \sim r_t$.

After the task covariance is estimated, we will use it to learn a new task generated from the same task distribution. (Kong et al., 2020b;a; Tripuraneni et al., 2020; Du et al., 2020) assume that task covariance is exactly low-rank, so they search for $\mathbf{\Lambda}$ in the low dimensional subspace and are able to learn with $\mathcal{O}(r_f)$ samples. We ask whether it is possible to search in the whole d dimensional space with $\mathcal{O}(r_f)$ samples (in which case we have to train in an overparameterized regime). If so, does it yield a smaller error compared to a low dimensional representation? This is meaningful especially when the task covariance $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ is approximately but not exactly low rank. (Bartlett et al., 2020) analyzes the generalization guarantee of overparameterized linear regression, (Chang et al., 2020) extends it to non-linear models, and (Nakkiran et al., 2020; Wu & Xu, 2020) analyze ridge regression with weighted regularizer, which also covers overparameterized linear regression. By contrast, we propose learning an (overparameterized) optimally-shaped representation that we then use in the few-shot learning phase, providing an end-to-end performance study of meta-learning.

2. Problem Setup

The meta-learning setup we consider consists of two phases: (i) representation learning, where prior tasks are used to learn a suitable representation, and (ii) few-shot learning, where a new task is learned with only a few samples. To aid in the theoretical analysis and understanding the generic behavior of the sample efficiency of general meta learning algorithms (see Fig. 1), we study linear regression tasks, and focus on a setup where the tasks and the features per task are generated randomly.

We assume that the two phases share the same distributions for features and tasks. In the first phase, there are multiple tasks, each with a batch of available data. The underlying task vectors are generated from the same distribution. We make this setup more precise using the following definitions.

Definition 1 ***Representation learning: tasks.*** *Suppose the tasks are i.i.d. drawn from the distribution $\mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{\beta}})$ where $\mathbf{\Sigma}_{\boldsymbol{\beta}} \in \mathbf{S}_+^d$. We denote the ith task by $\boldsymbol{\beta}_i \in \mathbb{R}^d$ and the number of tasks by n_{task}.*

After we randomly draw tasks from the distribution, we generate data as defined below.

Definition 2 ***Representation learning: data per task.*** *Fixing $\boldsymbol{\beta}_i$, we generate features $\boldsymbol{x}_{ij} \in \mathbb{R}^d$, labels $y_{ij} \in \mathbb{R}$ for $i = 1, ..., n_{\text{spt}}$, which follows $y_{ij} = \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i + \varepsilon_{ij}$. $\boldsymbol{x}_{ij}$ across all tasks are generated from $\mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{X}})$. Without loss of generality, we assume $\mathbf{\Sigma}_{\boldsymbol{X}}$ is diagonal (the task covariance $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ can be any positive semidefinite matrix). ε_{ij} denotes the additive noise with distribution $\mathcal{N}(0, \sigma_\varepsilon)$. For simplicity we assume $\mathbf{\Sigma}_{\boldsymbol{X}}$ and $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ are scaled so that $\|\mathbf{\Sigma}_{\boldsymbol{X}}\|, \|\mathbf{\Sigma}_{\boldsymbol{\beta}}\| \geq 1$.*

In Section 3 we study different estimators for the task covariance (or its low-rank approximation). We assume $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ is rank r_t and $\mathbf{\Sigma}_{\boldsymbol{X}}$ is rank r_f. In general r_t, r_f can be as large as d, but we will see the benefit when $\mathbf{\Sigma}_{\boldsymbol{\beta}}, \mathbf{\Sigma}_{\boldsymbol{X}}$ are approximately low rank.

Next we define the few-shot learning phase.

Definition 3 ***Few shot learning.*** *In the few shot learning phase, suppose the task vector $\boldsymbol{\beta}$ is generated from $\mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{\beta}})$, the features $\boldsymbol{x}_i$, $i = 1, ..., n_{\text{fs}}$ are independently generated from $\mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{X}})$, and $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i$ where noise ε_i are independently generated from $\mathcal{N}(0, \sigma_\varepsilon)$.*

We investigate the training error and the sample efficiency for the few-shot learning phase with respect to the representation of data. In (Kong et al., 2020b;a; Tripuraneni et al., 2020; Du et al., 2020), the task covariance is assumed to be rank r_t (i.e., representation of the features is r_t dimensional). We propose a weighted representation of features as below, which is in general R dimensional and the new task can be learned with $\mathcal{O}(r_t)$ samples even if $R \approx d$.

Definition 4 ***Shaped representation.*** *We define the weighted representation of features $\boldsymbol{x}$ with a positive semidefinite shaping matrix $\mathbf{\Lambda} \in \mathbb{R}^{R \times d}$ as $\mathbf{\Lambda}\boldsymbol{x}$.*

In the next section, we will learn the downstream task vector by finding the least norm solution with the shaped representation. The optimal shaping depends on the covariance of task and feature. Since we do not know their covariance, we use the covariance estimators in representation learning phase.

3. Main Results

Our meta-learning scheme involves two phases: representation learning and few-shot learning. The Algorithm 1 frames our linear meta-learning approach. We first compute a moment estimator to obtain a representation $\mathbf{\Lambda}$ in the representation learning phase. This representation can apply a simple subspace-based dimension reduction (PCA option), as in (Kong et al., 2020b;a; Tripuraneni et al., 2020), or can be high-dimensional which allows for a *softer* refinement of raw features by *shaping* their covariance (Shaping option). We then use this feature representation in the few-shot learning phase on a new task to obtain a model $\hat{\boldsymbol{\beta}}_{\mathbf{\Lambda}}$ with small risk.

Algorithm 1 Meta-learning

Require: *Representation learning data:* As in Def. 1, 2
Few-shot learning data: As in Def. 3
Value of R if using PCA
Representation learning: Calculate any moment estimator from Section 3.1.
if PCA **then**
Let $\mathbf{\Lambda} \in \mathbb{R}^{R\times d}$ be the projection to the span of top R singular values of moment estimator.
else if Shaping **then**
Calculate $\mathbf{\Lambda}$ as defined in Def. 5 and (3.12).
end if
Few-shot learning:
Calculate $\hat{\boldsymbol{\beta}}_{\mathbf{\Lambda}}$ from (3.4).
Return: $\hat{\boldsymbol{\beta}}_{\mathbf{\Lambda}}$.

In the this section, we will present the analyzes of Algorithm 1 in two aspects: In Section 3.1, we discuss the method of moments (MoM), which learns the task distribution from the dataset collected from multiple tasks. In Section 3.2, we propose the optimal overparameterized representation of features, and show how to use the representation for few-shot learning. Finally we integrate the two parts and give an analysis of the overall meta-learning algorithm.

3.1. Representation learning

This section investigates the representation learning via different types of moment-based spectral estimators. In general, recovery guarantees for arbitrary covariance matrices $\mathbf{\Sigma}_{\boldsymbol{X}}$ and $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ suppose we have sufficient samples. We are particularly interested in the case when $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ is approximately $r_t \ll d$ rank, so that we can estimate the moments with fewer samples. Estimating the moments is our primary goal, which can give full information about task covariance (if $\mathbf{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$, as we will show later). We can also obtain other key variables such as the principal eigenspace. We are further interested in the case when the feature covariance $\mathbf{\Sigma}_{\boldsymbol{X}}$ is approximately low rank and its principal eigenspace covers the top eigenvectors of $\mathbf{\Sigma}_{\boldsymbol{\beta}}$. Based on this, we consider the following estimators. Suppose the rank of $\mathbf{\Sigma}_{\boldsymbol{X}}$ is approximately r_f; we are interested in the $r_t < r_f \ll d$ case but hope that the estimators work well with general r_f, r_t.

1. **Method of moment (MoM).**

$$\hat{\boldsymbol{M}} = \frac{1}{n_{\text{task}}}\sum_{i=1}^{n_{\text{task}}}\frac{2}{n_{\text{spt}}^2}\left[\sum_{j=1}^{n_{\text{spt}}/2} y_{ij}y_{(i+n_{\text{spt}}/2)j} \cdot(\boldsymbol{x}_{ij}\boldsymbol{x}_{(i+n_{\text{spt}}/2)j}^\top + \boldsymbol{x}_{(i+n_{\text{spt}}/2)j}\boldsymbol{x}_{ij}^\top)\right]. \tag{3.1}$$

The mean of this estimator is $\boldsymbol{M} = \mathbf{\Sigma}_{\boldsymbol{X}}\mathbf{\Sigma}_{\boldsymbol{\beta}}\mathbf{\Sigma}_{\boldsymbol{X}}$ and we can estimate the top r_t eigenvector with $\tilde{\mathcal{O}}(r_f r_t^2)$ samples.

Remark: While this paper will focus on (3.1), one can also consider the symmetric version of the estimator (as in (Tripuraneni et al., 2020)) above (for which results are in similar spirit).

$$\hat{\boldsymbol{Q}} = \frac{1}{n_{\text{task}}}\sum_{i=1}^{n_{\text{task}}}\frac{1}{n_{\text{spt}}^2}\sum_{j=1}^{n_{\text{spt}}} y_{ij}^2\boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top. \tag{3.2}$$

The mean of this estimator is $\boldsymbol{Q} = 2\mathbf{\Sigma}_{\boldsymbol{X}}\mathbf{\Sigma}_{\boldsymbol{\beta}}\mathbf{\Sigma}_{\boldsymbol{X}} + \mathbf{tr}(\mathbf{\Sigma}_{\boldsymbol{\beta}}\mathbf{\Sigma}_{\boldsymbol{X}})\mathbf{\Sigma}_{\boldsymbol{X}}$ and its error is similar to $\hat{\boldsymbol{M}}$[2].

2. **MoM after Task-Averaging (MoM-TA).** If we further assume each task contains at least $n_{\text{spt}} = \Omega(r_t)$ samples, then we can estimate each single task by $\hat{\boldsymbol{\beta}}_i = \sum_{j=1}^{n_{\text{spt}}} y_{ij}\boldsymbol{x}_{ij}$. Let $\hat{\boldsymbol{B}} = [\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_k]$. When $\mathbf{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$, we can estimate the top r_t eigenvectors of $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ with $\tilde{\mathcal{O}}(dr_f)$ samples. This is better than (Kong et al., 2020b; Tripuraneni et al., 2020) which require $\tilde{\mathcal{O}}(dr_t^2)$ samples. The MoM-TA estimate obeys the result of MoM estimator as we remark that $\frac{1}{n_{\text{task}}}\mathbb{E}[\hat{\boldsymbol{B}}\hat{\boldsymbol{B}}^\top] = \boldsymbol{Q}$.

3. **PCA on Feature Covariance (MoM-F).** If $\mathbf{\Sigma}_{\boldsymbol{X}}$ and $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ are approximately rank r_t and the span of top r_t eigenvectors of $\mathbf{\Sigma}_{\boldsymbol{X}}$ and $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ are closely aligned, we can directly estimate the covariance of features by

$$\hat{\mathbf{\Sigma}}_{\boldsymbol{X}} = \sum_{j=1}^{n_{\text{spt}}}\sum_{i=1}^{n_{\text{task}}}\boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top \tag{3.3}$$

The mean of this estimator is $\mathbf{\Sigma}_{\boldsymbol{X}}$ and we can estimate the top r_t eigenvector of $\mathbf{\Sigma}_{\boldsymbol{X}}$ with $\tilde{\mathcal{O}}(r_t)$ samples.

In the following subsections, let $n_{\text{tot}} = n_{\text{spt}}n_{\text{task}}$, and let $\mathcal{S} = \max\{\|\mathbf{\Sigma}_{\boldsymbol{X}}\|, \|\mathbf{\Sigma}_{\boldsymbol{\beta}}\|\}$, $\mathcal{T}_{\boldsymbol{\beta},\boldsymbol{X}} = \mathbf{tr}(\mathbf{\Sigma}_{\boldsymbol{\beta}}\mathbf{\Sigma}_{\boldsymbol{X}})$, $\mathcal{T}_{\boldsymbol{X}} = \mathbf{tr}(\mathbf{\Sigma}_{\boldsymbol{X}})$, $\mathcal{T}_{\boldsymbol{\beta}} = \mathbf{tr}(\mathbf{\Sigma}_{\boldsymbol{\beta}})$.

3.1.1. Method of moments

We first study $\hat{\boldsymbol{M}}$ in (3.1). Suppose $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ is approximately rank r_t and $\mathbf{\Sigma}_{\boldsymbol{X}}$ is approximately rank r_f, and $r_f \geq r_t$. If

[2] In the appendix we provide the analysis for both $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{Q}}$.

the span of top r_t eigenvectors of $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ is covered by the span of top r_f eigenvectors of $\boldsymbol{\Sigma}_{\boldsymbol{X}}$, then the r_t-PCA of $\boldsymbol{M}$ gives an estimate of the span of top r_t eigenvector of $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$, and for this we need only an accurate estimate of $\boldsymbol{M}$, given by $\hat{\boldsymbol{M}}$.

Theorem 1 *Assume the representation learning dataset is generated as in Def 1, 2. Let* $N_0 = (\log(n_{\text{task}})\mathcal{T}_{\boldsymbol{\beta},\boldsymbol{X}} + \sigma_\varepsilon^2)\mathcal{T}_{\boldsymbol{\beta}}$, *with probability at least* $1 - (n_{\text{tot}}\mathcal{T}_{\boldsymbol{\beta}})^{-10} - (n_{\text{tot}}N_0)^{-10}$, *the moment estimator* $\hat{\boldsymbol{M}}$ *above satisfies*

$$\|\hat{\boldsymbol{M}} - \boldsymbol{M}\| \lesssim \underbrace{\sqrt{\frac{\mathcal{S}\mathcal{T}_{\boldsymbol{\beta},\boldsymbol{X}}^2\mathcal{T}_{\boldsymbol{X}}}{n_{\text{tot}}}}}_{\substack{\mathcal{E}_{\text{mb}}(\boldsymbol{\Sigma}_{\boldsymbol{X}},\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \\ \text{MoM bias}}} + \sigma_\varepsilon \underbrace{\sqrt{\frac{\mathcal{S}(\mathcal{T}_{\boldsymbol{\beta},\boldsymbol{X}} + \sigma_\varepsilon^2\mathcal{T}_{\boldsymbol{\beta}})}{n_{\text{tot}}}}}_{\substack{\mathcal{E}_{\text{mv}}(\boldsymbol{\Sigma}_{\boldsymbol{X}},\boldsymbol{\Sigma}_{\boldsymbol{\beta}},\sigma_\varepsilon) \\ \text{MoM variance}}} + \underbrace{\mathcal{S}^2\sqrt{\frac{\mathcal{S}\mathcal{T}_{\boldsymbol{\beta}}}{n_{\text{task}}}}}_{\substack{\mathcal{E}_{\text{tb}}(\boldsymbol{\Sigma}_{\boldsymbol{X}},\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \\ \text{task bias}}}$$

We define the three parts of the error as $\mathcal{E}_{\text{mb}}$, $\mathcal{E}_{\text{mv}}$, $\mathcal{E}_{\text{tb}}$, and define $\mathcal{E}(\boldsymbol{\Sigma}_{\boldsymbol{X}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}, \sigma_\varepsilon)$ as the sum of $\mathcal{E}_{\text{mb}}$, $\mathcal{E}_{\text{mv}}$, $\mathcal{E}_{\text{tb}}$. We make clear their dependence on the covariance of tasks and features, which will be used in Theorem 4. Suppose $\boldsymbol{M}$ is approximately rank r_t, which means $\lambda_{r_t} - \lambda_{r_t+1}$ is large, then if we do r_t PCA on $\hat{\boldsymbol{M}}$ and denote the span of top r_t eigenvectors as $\hat{\boldsymbol{W}}$, its angle with the span of top r_t eigenvectors of $\boldsymbol{M}$, named $\boldsymbol{W}$, is bounded by the well known result in (Davis & Kahan, 1970). We will give a concrete example below showing the recovery error of $\hat{\boldsymbol{M}}$ and its top eigenvectors.

Example 1 *Suppose* $\boldsymbol{\Sigma}_{\boldsymbol{X}} = diag(\boldsymbol{I}_{r_f}, \iota\boldsymbol{I}_{d-r_f})$, *and* $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = diag(\boldsymbol{I}_{r_t}, 0)$, $\sigma_\varepsilon = 0$, *then*[3]

$$\|\hat{\boldsymbol{M}} - \boldsymbol{M}\| \lesssim \sqrt{\frac{r_t^2(r_f + \iota(d - r_f))}{n_{\text{tot}}}} + \sqrt{\frac{r_t}{n_{\text{task}}}}.$$

Suppose $n_{\text{tot}} \gtrsim r_t^2(r_f + \iota(d - r_f))$, *then we have*

$$\sin(\angle\boldsymbol{W}, \hat{\boldsymbol{W}}) \lesssim \sqrt{\frac{r_t^2(r_f + \iota(d - r_f))}{n_{\text{tot}}}}$$

When $\iota = 1$, i.e., $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$, we need $\tilde{\mathcal{O}}(dr_t^2)$ samples to estimate $\boldsymbol{M}$, which is also addressed in the analysis of related literature (Kong et al., 2020b; Tripuraneni et al., 2020). However, when the feature is approximately low rank and aligns with task, we need $\tilde{\mathcal{O}}(r_f r_t^2)$ samples to estimate $\boldsymbol{M}$ and its top eigenvectors cover the range of tasks. This alignment of tasks and features is discovered in the related work that study the spectrum of the kernel matrices of Hessians of deep neural networks, such as (Sagun et al., 2017; Arora et al., 2019; Oymak et al., 2019).

[3]The second term can also be written as $\sqrt{r_t n_{\text{spt}}/n_{\text{tot}}}$. Typically $r_t \gtrsim n_{\text{spt}}$ so the first term is larger.

3.1.2. Task average estimators

In the previous MoM estimators, if the features are standard normal vectors, and the task covariance is rank r_t, then we need to estimate the subspace spanned by tasks with dr_t^2 samples. In this section, we will show that, if there are enough samples in each task, we are able to retrieve the subspace with dr_f samples. Let $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I})$, and each task contains $n_{\text{spt}} \gtrsim r_t$ samples, then we can get the span of $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ with $\tilde{\mathcal{O}}(dr_t)$ samples.

Theorem 2 *(Standard normal feature, noiseless) Let data be generated as in Def 1, 2. Suppose* $\sigma_\varepsilon = 0$, $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$, *and suppose the rank of* $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ *is* r_t. *Define* $\hat{\boldsymbol{\beta}}_j = \sum_{j=1}^{n_{\text{spt}}} y_{ij}\boldsymbol{x}_{ij}$, $\boldsymbol{B} = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_k]$, *and* $\hat{\boldsymbol{B}} = [\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_k]$. *Let* $n_{\text{spt}} > c_1\mathcal{T}_{\boldsymbol{\beta}}\lambda_{r_t}^{-1}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, $n_{\text{task}} > c_2\max\{d, \frac{\mathcal{S}\mathcal{T}_{\boldsymbol{\beta}}}{\lambda_{r_t}^2(\boldsymbol{\Sigma}_{\boldsymbol{\beta}})}\}$, *with probabilty* $1 - (n_{\text{task}}^{-c_3} + (n_{\text{task}}\mathcal{T}_{\boldsymbol{\beta}})^{-c_4} + \exp(-c_5 n_{\text{task}}^2))$, *where* c_i *are constants,* $c_{3,4} > 10$,

$$\sigma_{\max}(\hat{\boldsymbol{B}} - \boldsymbol{B}) \lesssim \sqrt{\frac{n_{\text{task}}\mathcal{T}_{\boldsymbol{\beta}}}{n_{\text{spt}}}}.$$

Denote the span of top r_t *singular column vectors of* $\hat{\boldsymbol{B}}$ *and* $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ *as* $\hat{\boldsymbol{W}}, \boldsymbol{W}$, *then*

$$\sin(\angle\hat{\boldsymbol{W}}, \boldsymbol{W}) \lesssim \sqrt{\frac{\mathcal{T}_{\boldsymbol{\beta}}}{n_{\text{spt}}\lambda_{r_t}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}})}}.$$

If $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = diag(\boldsymbol{I}_{r_t}, 0)$, *then* $\sin(\angle\hat{\boldsymbol{W}}, \boldsymbol{W}) \leq \sqrt{\log(n_{\text{task}})r_t/n_{\text{spt}}}$.

In the appendix, we will propose a theorem with general feature covariance $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and noisy data, as a generalization of Theorem 2.

Remark 1 *Theorem 2 requires* $n_{\text{spt}} > c_1\mathcal{T}_{\boldsymbol{\beta}}\lambda_{r_t}^{-1}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, *so* n_{spt} *is lower bounded by* $\mathcal{O}(r_t)$ *in Theorem 2. If* $n_{\text{spt}} = 1$ *and we estimate* $\hat{\boldsymbol{B}}$, *then* $n_{\text{task}}\hat{\boldsymbol{Q}} = \hat{\boldsymbol{B}}\hat{\boldsymbol{B}}^\top$. *This means Theorem 1 can be applied to this estimator when* $n_{\text{spt}} = 1$.

3.1.3. Estimating the covariance of features

As we have defined in Def 2, features $\boldsymbol{x}_{ij}$ are generated from $\mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$. We aim to estimate the covariance $\boldsymbol{\Sigma}_{\boldsymbol{X}}$. Although there are different kinds of algorithms, such as maximum likelihood estimator (Anderson et al., 1970), to be consistent with the algorithms in the latter sections, we study the sample covariance matrix defined by (3.3).

Lemma 1 *Suppose* $\boldsymbol{x}_{ij}$ *are generated independently from* $\mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$. *We estimate* (3.3), *then when* $n_{\text{tot}} \gtrsim \mathcal{T}_{\boldsymbol{X}}$, *with probability at least* $1 - (n_{\text{tot}}\mathbf{tr}(\boldsymbol{\Sigma}_{\boldsymbol{X}}))^{-10}$,

$$\|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}} - \boldsymbol{\Sigma}_{\boldsymbol{X}}\| \lesssim \sqrt{\frac{\mathcal{S}\mathcal{T}_{\boldsymbol{X}}}{n_{\text{tot}}}}$$

Denote the span of top r_f eigenvectors of $\boldsymbol{\Sigma_X}$ as $\boldsymbol{W}$ and the span of top r_f eigenvectors of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}}$ as $\hat{\boldsymbol{W}}$. Let $\delta_\lambda = \lambda_{r_f}(\boldsymbol{\Sigma_X}) - \lambda_{r_f+1}(\boldsymbol{\Sigma_X})$. Then if $n_{\text{tot}} \gtrsim \frac{\mathcal{ST}_{\boldsymbol{X}}}{\delta_\lambda^2}$, we have

$$\sin(\angle \boldsymbol{W}, \hat{\boldsymbol{W}}) \lesssim \sqrt{\frac{\mathcal{ST}_{\boldsymbol{X}}}{n_{\text{tot}}\delta_\lambda^2}}$$

Example 2 *When $\boldsymbol{\Sigma_X} = diag(\boldsymbol{I}_{r_f}, 0)$, we have* $\sin(\angle \boldsymbol{W}, \hat{\boldsymbol{W}}) \lesssim \sqrt{\frac{r_f}{n_{\text{tot}}}}$.

Lemma 1 gives the quality of the estimation of the covariance of features $\boldsymbol{x}$. When the condition number of the matrix $\boldsymbol{\Sigma_X}$ is close to 1, we need $n_{\text{tot}} \gtrsim d$ to get an estimation with error $\mathcal{O}(1)$. However, when the matrix $\boldsymbol{\Sigma_X}$ is close to rank r_f, the amount of samples to achieve the same error is smaller, and we can use $n_{\text{tot}} \gtrsim r_f$ samples to get $\mathcal{O}(1)$ estimation error.

3.2. Few-shot learning

In few-shot learning phase, we hope to learn the feature $\boldsymbol{\beta} \in \mathbb{R}^d$ which is generated from $\mathcal{N}(0, \boldsymbol{\Sigma_\beta})$. The data is $(\boldsymbol{x}_i, y_i)_{i=1,\dots,n_{\text{fs}}}$ where $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma_X})$, and $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i$ where $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon)$. We assume $n_{\text{fs}} < d$. Denote $\boldsymbol{X} \in \mathbb{R}^{n_{\text{fs}} \times d}$ whose ith row is $\boldsymbol{x}_i$, and $\boldsymbol{y} = [y_1, ..., y_m]^\top$. Suppose we select a shaping matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d\times d}$, we are interested in the least norm solution defined as

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{\Lambda}} = \arg\min_{\boldsymbol{\alpha}'} \|\boldsymbol{\alpha}'\|_{\ell_2} \text{ s.t. } \boldsymbol{y} = \boldsymbol{X\Lambda\alpha}' \tag{3.4a}$$

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}\hat{\boldsymbol{\alpha}}_{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}(\boldsymbol{X\Lambda})^\dagger \boldsymbol{y}. \tag{3.4b}$$

The (excess) risk of $\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}}$ is given by

$$\text{risk}(\boldsymbol{\Lambda}, \boldsymbol{\Sigma_\beta}) = \boldsymbol{E}_{\boldsymbol{x},y,\boldsymbol{\beta}}(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}})^2 \tag{3.5}$$

$$= \boldsymbol{E}_{\boldsymbol{\beta}}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}} - \boldsymbol{\beta})^\top \boldsymbol{\Sigma_X}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}} - \boldsymbol{\beta}) + \sigma_\varepsilon^2. \tag{3.6}$$

We want to solve for the optimal representation with $\boldsymbol{\Sigma_\beta}$ as

$$\boldsymbol{\Lambda}^* = \arg\min_{\boldsymbol{\Lambda}' \in \boldsymbol{S}_{++}^d} \text{risk}(\boldsymbol{\Lambda}', \boldsymbol{\Sigma_\beta}) \tag{3.7}$$

Define $\boldsymbol{\Lambda} = \arg\min_{\boldsymbol{\Lambda}' \in \boldsymbol{S}_{++}^d} \text{risk}(\boldsymbol{\Lambda}', \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}})$.

First we observe that, the shaping in Def 4 is a special case of the weighted ridge regression discussed in (Wu & Xu, 2020). We observe the following equivalence of these two descriptions.

Observation 1 *Let $\boldsymbol{X} \in \mathbb{R}^{n_{\text{fs}}\times d}$ and $\boldsymbol{y} \in \mathbb{R}^{n_{\text{fs}}}$, and define*

$$\hat{\boldsymbol{\beta}}_1 = \boldsymbol{\Lambda}(\boldsymbol{X\Lambda})^\dagger \boldsymbol{y}, \tag{3.8}$$

$$\hat{\boldsymbol{\beta}}_2 = \lim_{t\to 0} \text{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{X}^\top \boldsymbol{\beta} - \boldsymbol{y}\|^2 + t\boldsymbol{\beta}^\top \boldsymbol{\Lambda}^{-2}\boldsymbol{\beta}, \tag{3.9}$$

then $\hat{\boldsymbol{\beta}}_1 = \hat{\boldsymbol{\beta}}_2$.

In Section 3.2.1, we will derive an expression for the risk with an arbitrary representation $\boldsymbol{\Lambda}$. Note we can only use $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$, not $\boldsymbol{\Sigma_\beta}$ to obtain $\boldsymbol{\Lambda}$. In Section 3.2.2, we bound the sensitivity of risk in $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} - \boldsymbol{\Sigma_\beta}$. Finally we obtain an end to end guarantee of the risk of the whole meta-learning algorithm, including representation learning and few-shot learning phases.

3.2.1. Computing Optimal Representation

Prior work typically (Kong et al., 2020b;a; Tripuraneni et al., 2020) projects the features onto the subspace for few shot learning. In **Q1** we ask, what can be said about the performance of a general linear representation with arbitrary dimension? For a given $\boldsymbol{\Lambda}$, the following theorem characterizes the exact asymptotic risk that helps us differentiate the performance of different linear representations.

In the follwing discussion, we assume that $\boldsymbol{\Sigma_X}$, $\boldsymbol{\Sigma_\beta}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ share the same eigenspace, and suppose they are diagonal. In this case, we search over diagonal representation matrix $\boldsymbol{\Lambda}$. Next, we will propose an expression for computing the representation matrix $\boldsymbol{\Lambda}$ as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$.

Definition 5 *(Precise few-shot risk) When $d > n_{\text{fs}}$ there exists unique $\xi > 0$ such that*

$$n_{\text{fs}} = \sum_{i=1}^d \left(1 + (\xi \boldsymbol{\Sigma_X}_i)^{-1}\right)^{-1}, \tag{3.10}$$

Define $\boldsymbol{\theta} \in \mathbb{R}^d$ to be $\boldsymbol{\theta}_i = \frac{\xi\boldsymbol{\Lambda}_i}{1+\xi\boldsymbol{\Lambda}_i}$, and define the risk as

$$risk(\boldsymbol{\Lambda}, \boldsymbol{\Sigma_\beta}) = \frac{1}{n_{\text{fs}} - \|\boldsymbol{\theta}\|^2}\left(\frac{n_{\text{fs}}}{d}\sum_{i=1}^d (1-\boldsymbol{\theta}_i)^2 \boldsymbol{\Sigma_\beta}_i + \|\boldsymbol{\theta}\|^2\sigma_\varepsilon^2\right). \tag{3.11}$$

We denote the right hand side as $f(\boldsymbol{\theta}; \boldsymbol{\Sigma_\beta})$.

Recall that we defined the few-shot risk as in (3.5). In the appendix, we derive the asymptotical risk applying the CGMT in (Thrampoulidis et al., 2015), and show that (3.5) and (3.11) are asymptotically equivalent when n_{fs}/d is fixed and $n_{\text{fs}}, d \to \infty$.

Finding optimal representation. Definition 5 grants us access to a closed-form risk for any linear representation. Thus, one can solve for the optimization representation by minimizing this risk using the parameterization $\boldsymbol{\theta}$.[4]

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}; \boldsymbol{\Sigma_\beta}), \text{ s.t. } 0 \le \boldsymbol{\theta} < 1, \sum_{i=1}^d \boldsymbol{\theta}_i = n_{\text{fs}}$$

Recalling $\boldsymbol{\theta}_i = \frac{\xi\boldsymbol{\Lambda}_i}{1+\xi\boldsymbol{\Lambda}_i}$, we then find the *optimal representation* via the reverse map $\boldsymbol{\Lambda}_i^* = (1/\boldsymbol{\theta}_i^* - 1)^{-1}/\xi$.

Ensuring robustness. We use $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ instead of $\boldsymbol{\Sigma_\beta}$ for computing the optimal representation $\boldsymbol{\Lambda}$, thus we need the risk to be robust with respect to $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} - \boldsymbol{\Sigma_\beta}$. Let $0 < \underline{\theta} < n_{\text{fs}}/d$,

[4] The optimization problem is not convex. In appendix we provide an algorithm solving it in polynomial time following the proof of this theorem.

in implementation we require $\underline{\theta} \le \boldsymbol{\theta} \le 1 - \frac{d-n_{\rm fs}}{n_{\rm fs}}\underline{\theta}$ instead of $0 \le \boldsymbol{\theta} \le 1$ for robustness concerns. The sensitivity in $\underline{\theta}$ is shown in Theorem 3.

Generally, the optimal representation is a d dimensional matrix and it is not guaranteed to be low rank. When $n_{\rm fs} < d$, the downstream problem is overparameterized. The overmarameterization is often used for complicated models such as neural networks, and we justified it via the linear model. The overparameterized linear regression is also studied in (Wu & Xu, 2020; Bartlett et al., 2020), and we propose the algorithm for arbitrary feature and task covariance.

We give the algorithm computing the optimal shaping with arbitrary dimension R below.

R dimensional optimal representation: Let $\boldsymbol{\Sigma_\beta} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ be the eigendecomposition of $\boldsymbol{\Sigma_\beta}$. Let $\boldsymbol{U}_1 \in \mathbb{R}^{d\times R}$ be the first R columns block of $\boldsymbol{U}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{d\times(d-R)}$ be the remaining $d-R$ columns block. Let $\boldsymbol{X}_R = \boldsymbol{X}\boldsymbol{U}_1 \in \mathbb{R}^{n_{\rm fs}\times R}$. We again define a shaping matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{R\times R}$ as in 3.4. $\boldsymbol{\Lambda}_i$'s are diagonal values of $\boldsymbol{\Lambda}$. $\hat{\boldsymbol{\beta}}_R = \boldsymbol{U}_1^\top \hat{\boldsymbol{\beta}}$, where $\hat{\boldsymbol{\beta}}$ is square roots of diagonal values of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$, and $\hat{\boldsymbol{\beta}}_{Ri}$ denotes its ith entry. $\boldsymbol{\Sigma}_{Ri}$ is ith diagonal element of $\boldsymbol{\Sigma}_R$. We define the folloing quantities.

$$\boldsymbol{\Sigma}_R = \boldsymbol{U}_1^\top \boldsymbol{\Sigma_X} \boldsymbol{U}_1, \quad \boldsymbol{\Sigma}_{\boldsymbol{\beta}_R} = \boldsymbol{U}_1^\top \boldsymbol{\Sigma_\beta} \boldsymbol{U}_1 \tag{3.12a}$$

$$\boldsymbol{\Lambda}_R = \arg\min_{\boldsymbol{\Lambda}'\in\mathbb{R}^{R\times R}} \text{risk}(\boldsymbol{\Lambda}', \boldsymbol{\Sigma}_{\boldsymbol{\beta}_R}) \tag{3.12b}$$

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{\Lambda}R} = \arg\min_{\boldsymbol{\alpha}'\in\mathbb{R}^R} \|\boldsymbol{\alpha}'\|_{\ell_2} \text{ s.t. } \boldsymbol{y} = \boldsymbol{X}_R\boldsymbol{\Lambda}_R\boldsymbol{\alpha}' \tag{3.12c}$$

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}R} = \boldsymbol{\Lambda}_R\hat{\boldsymbol{\alpha}}_{\boldsymbol{\Lambda}R} = \boldsymbol{\Lambda}_R(\boldsymbol{X}_R\boldsymbol{\Lambda}_R)^\dagger \boldsymbol{y} \tag{3.12d}$$

Define $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^\perp, \boldsymbol{\Sigma}_{\boldsymbol{X}}^\perp$ as the projection of $\boldsymbol{\Sigma_\beta}, \boldsymbol{\Sigma_X}$ onto $\boldsymbol{U}_2$, the noise variance is equivalent to $\sigma_{\varepsilon R}^2 = \sigma_\varepsilon^2 + \mathbf{tr}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^\perp \boldsymbol{\Sigma}_{\boldsymbol{X}}^\perp)$. Note we define R dimensional optimal representation as if we know the covariance matrix $\boldsymbol{\Sigma_\beta}$, when we have only $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$, these definitions can be applied in the same way[5]. We will discuss how truncation level R affects risk in Remark 2.

3.2.2. Robustness of optimal representation

In this part we focus on the identity feature case $\boldsymbol{\Sigma_X} = \boldsymbol{I}$, and study the robustness of few-shot learning risk with respect to inaccuracy of representation learning.

In Section 3.1, suppose one uses the estimator[6] $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \hat{\boldsymbol{M}}$ to estimate the task covariance $\boldsymbol{\Sigma_\beta}$, and suffers the error $\|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} - \boldsymbol{\Sigma_\beta}\| \le \delta_{\boldsymbol{\Sigma_\beta}}$. In few-shot learning phase, one uses $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ as the nominal task covariance and get $\boldsymbol{\Lambda}$. With the true task distribution, the risk is $\text{risk}(\boldsymbol{\Lambda}, \boldsymbol{\Sigma_\beta})$. We are interested in its difference from the optimal risk $\text{risk}(\boldsymbol{\Lambda}, \boldsymbol{\Sigma_\beta}) - \text{risk}(\boldsymbol{\Lambda}^*, \boldsymbol{\Sigma_\beta})$. Note that (Wu & Xu, 2020) Sec.6 gives the exact value of $\text{risk}(\boldsymbol{\Lambda}^*, \boldsymbol{\Sigma_\beta})$ so we have an end to end error guarantee.

Theorem 3 *Suppose the data is generated as Definition 3, $\boldsymbol{\Lambda}$ and $\underline{\theta}$ are defined in Def. 5 and the estimated task is obtained as (3.4). Suppose $\|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} - \boldsymbol{\Sigma_\beta}\| \le \delta_{\boldsymbol{\Sigma_\beta}}$. Then the risk of few-shot learning phase suffers at most*

$$\textit{risk}(\boldsymbol{\Lambda}, \boldsymbol{\Sigma_\beta}) - \textit{risk}(\boldsymbol{\Lambda}^*, \boldsymbol{\Sigma_\beta}) \le \frac{2n_{\rm fs}^2 \delta_{\boldsymbol{\Sigma_\beta}}}{(d-n_{\rm fs})(2n_{\rm fs} - d\underline{\theta})\underline{\theta}}.$$

Theorem 3 shows the robustness of few-shot learning algorithm with respect to the error in representation learning phase. Given Theorem 1 that bounds $\delta_{\boldsymbol{\Sigma_\beta}}$, we will propose the bound of Algorithm 1 in the following theorem.

Theorem 4 *Suppose $\boldsymbol{\Sigma_X} = \boldsymbol{I}$. We run Algorithm 1 and set $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \hat{\boldsymbol{M}}$. The optimal shaping matrix $\boldsymbol{\Lambda}_R$ depends on $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$. Let $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_R}$ be the projecion of $\boldsymbol{\Sigma_\beta}$ onto the top R eigenvector space, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_R^\perp} = \boldsymbol{\Sigma_\beta} - \boldsymbol{\Sigma}_{\boldsymbol{\beta}_R}$. The asymptotic risk of few-shot learning is upper bounded by*

$$\textit{risk}(\boldsymbol{\Lambda}_R, \boldsymbol{\Sigma_\beta}) - \textit{risk}(\boldsymbol{\Lambda}_R^*, \boldsymbol{\Sigma_\beta}) \tag{3.13}$$

$$\lesssim \frac{n_{\rm fs}^2 \cdot \mathcal{E}\left(\boldsymbol{\Sigma_X}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_R}, \sigma_\varepsilon^2 + \mathbf{tr}(\boldsymbol{\Sigma_X}\boldsymbol{\Sigma}_{\boldsymbol{\beta}_R^\perp})\right)}{(d-n_{\rm fs})(2n_{\rm fs} - d\underline{\theta})\underline{\theta}} \tag{3.14}$$

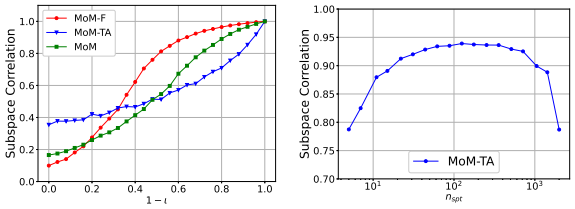
If we do not apply dimension reduction (i.e., $R = d$), then

$$\textit{risk}(\boldsymbol{\Lambda}, \boldsymbol{\Sigma_\beta}) \lesssim \textit{risk}(\boldsymbol{\Lambda}^*, \boldsymbol{\Sigma_\beta}) + \frac{n_{\rm fs}^2 \cdot \mathcal{E}(\boldsymbol{\Sigma_X}, \boldsymbol{\Sigma_\beta}, \sigma_\varepsilon)}{(d-n_{\rm fs})(2n_{\rm fs} - d\underline{\theta})\underline{\theta}}$$

Remark 2 ***(Risk with respect to PCA level** R**)** *We compare the behavior of (3.14) with different trunctation levels R. $\textit{risk}(\boldsymbol{\Lambda}_R^*, \boldsymbol{\Sigma_\beta})$ decrease with R. $\mathcal{E}$ is an increasing function with respect to R and noise. With R increasing, the third term $\sigma_\varepsilon^2 + \mathbf{tr}(\boldsymbol{\Sigma_X}\boldsymbol{\Sigma}_{\boldsymbol{\beta}_R^\perp})$ decreases. Thus generally the whole end to end error $\textit{risk}(\boldsymbol{\Lambda}_R, \boldsymbol{\Sigma_\beta})$ might not be monotone in R when $R > n_{\rm fs}$. This is depicted in Fig. 3(b). In essence, this result provides a theoretical justification on the existence of a sweet-spot for the optimal representation strategy. With infinite $n_{\rm tot}$ ($\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma_\beta}$), it is safe to learn a large representation. As $n_{\rm tot}$ decreases, it becomes difficult to estimate task covariance accurately, especially its small eigenvalues due to finite-sample estimation noise. Thus a large dimensional representation –that uses this noisy covariance– may result in noisier features leading to the excess risk term $\mathcal{E}$. Thus choosing R adaptively with $n_{\rm tot}$ can strike the right bias-variance tradeoff between the excess risk (variance) and the risk due to suboptimal representation $\boldsymbol{\Lambda}_R^*$ i.e. $\textit{risk}(\boldsymbol{\Lambda}_R^*, \boldsymbol{\Sigma_\beta}) - \textit{risk}(\boldsymbol{\Lambda}^*, \boldsymbol{\Sigma_\beta})$.*

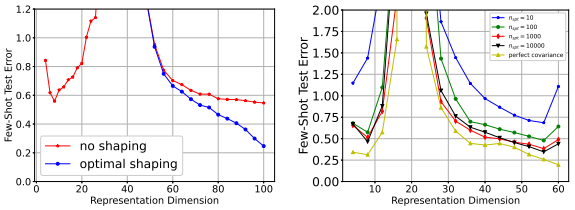
[5] In the definition below, $\boldsymbol{\Lambda}_R \in \mathbb{R}^{R\times R}$, the shaping matrix as defined in Algorithm 1 is $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_R\boldsymbol{U}_1^\top \in \mathbb{R}^{R\times d}$

[6] In essence we require $\hat{\boldsymbol{M}} - \boldsymbol{\Sigma_\beta}$ being small to compute $\boldsymbol{\Lambda}$ accurately. If $\boldsymbol{\Sigma_X} \ne \boldsymbol{I}$, logically $\boldsymbol{M} \ne \boldsymbol{\Sigma_\beta}$ and this estimation does not always work. In some special cases, such as $\boldsymbol{\Sigma_X} = \text{diag}(\boldsymbol{I}_{r_f}, \iota\boldsymbol{I}_{d-r_f})$ and $\boldsymbol{\Sigma_\beta} = \text{diag}(\boldsymbol{I}_{r_t}, 0)$ $(r_f > r_t)$, $\hat{\boldsymbol{M}}$ is still a good estimator of $\boldsymbol{\Sigma_\beta}$.



(a) (b)

Figure 2. (a) $\mathbf{\Sigma_\beta} = (\boldsymbol{I}_{10}, \mathbf{0}_{90})$. $\sigma_\varepsilon = 0.5$, $\mathbf{\Sigma_X} = (\boldsymbol{I}_{10}, \iota \cdot \boldsymbol{I}_{90})$, $n_{\text{task}} = 20$, $n_{\text{spt}} = 40$. Learning 10 dimensional top eigenspace of $\mathbf{\Sigma_\beta}$ with biased features. MoM-F, MoM-TA, MoM stands for $\hat{\mathbf{\Sigma}}_{\boldsymbol{X}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{Q}}$. Subspace correlation is defined as $\|\hat{\boldsymbol{U}}^\top \boldsymbol{U}\|^2 / \|\boldsymbol{U}\|^2$. When $\iota \to 0$ the feature and task are more aligned. (b) $\mathbf{\Sigma_\beta} = (\boldsymbol{I}_{10}, \mathbf{0}_{90})$, $\mathbf{\Sigma_X} = \boldsymbol{I}_{100}$, $\sigma_\varepsilon = 0.5$, learning with $n_{\text{tot}} = 20000$ samples with varying n_{task}. High correlation happens when n_{task} and n_{spt} are reasonably large.



(a) (b)

Figure 3. (a) $\mathbf{\Sigma_\beta} = (25 \cdot \boldsymbol{I}_{10}, \boldsymbol{I}_{90})$, $\mathbf{\Sigma_X} = \boldsymbol{I}_{100}$ $\sigma_\varepsilon = 0.5$, $n_{\text{fs}} = 40$, few shot learning with knowledge of $\mathbf{\Sigma_\beta}$, with identity or optimal shaping matrix. For $R < n_{\text{fs}}$, we are not in overparameterized regime, so there is no optimal shaping. (b) Algorithm 1 with $d = 60, n_{\text{fs}} = 20, n_{\text{task}} = 60$, $\mathbf{\Sigma_\beta} = (25 \cdot \boldsymbol{I}_6, \boldsymbol{I}_{54})$, $\mathbf{\Sigma_X} = \boldsymbol{I}_{60}$ and varying n_{spt}, R. End to end risk with different representation learning samples and dimensions. Optimal shaping (with respect to $\hat{\mathbf{\Sigma}}_{\boldsymbol{\beta}}$) applied.

4. Numerical Experiments

In this section, we will verify by experiments the three main contributions proposed before: (1) We apply the method of moment estimator for retrieving the covariance of the task or the top eigenspace. We show that when feature space aligns with the task space, we only need $\mathcal{O}(r_f r_t^2)$ instead of $\mathcal{O}(dr_t^2)$ samples. (2) In Theorem 2, we introduce task average estimator and argue that when each task contains $\Omega(r_t)$ samples, we can retrieve the subspace of tasks with $\mathcal{O}(r_f r_t)$ samples. (3) In Def. 5, we present the optimal representation matrix that minimizes the excess risk and in Theorem 4 we show the overall risk of Algorithm 1.

• **Error of representation learning with respect to task feature alignment.** We depict this in Fig. 2(a). After applying method of moments estimators we do r_t-SVD truncation. We can see from Fig. 2(a) that, MoM's learn the subspaces accurately when $1 - \iota$ is large, (i.e., with feature-task alignment), whereas behave worse when $\mathbf{\Sigma_X} = \boldsymbol{I}$ Specifically, The MoM-F estimator works better only when ι is small, i.e., high task-feature alignment.

• **Task average helps learning with fewer samples.** We show the sample efficiency of the task average estimator in Fig. 2(b). We fix the total number of samples n_{tot}, and vary the number of samples per task n_{spt}. In Theorem 2, we argue that it estimates the task subspace with $\mathcal{O}(dr_t)$ samples when $n_{\text{spt}} \geq r_t$. We can see from the figure that, when n_{spt} is small, this estimator degenerate to $\hat{\boldsymbol{Q}}$ which asks for $\mathcal{O}(dr_t^2)$ samples, so the subspace estimation is bad. When n_{task} is small, the sampled task features cannot approximate the true task distribution, which also cause large error. We need n_{spt} and n_{task} both reasonably large.

• **Optimal representation.** In Fig. 3(a), we know $\mathbf{\Sigma_\beta}$ and do R-SVD truncation and apply optimal shaping for varying R. When the problem becomes overparameterized ($n_{\text{fs}} < R$), the estimation of downstream taskdepends on the shaping matrix. The risk corresponding to the optimal shaping matrix is smaller than using raw features for solving the downstream task.

• **End to end behavior of meta-learning algorithm.** We run the whole meta-learning process as in Algorithm 1.The normalized risk is plotted in Fig. 3(b). We use the optimal shaping matrix $\mathbf{\Lambda}$, calculated as a function of $\hat{\mathbf{\Sigma}}_{\boldsymbol{\beta}}$, for solving the downstream task. Both dimension reduction ($n_{\text{fs}} > R$) or optimal overparameterized problem ($n_{\text{fs}} < R$) end up with small risk, whereas it becomes hard to learn when $n_{\text{fs}} \approx R$. The risk of overparameterized case can be smaller than using low dimensional representation (typically applied in (Tripuraneni et al., 2020; Kong et al., 2020b;a; Du et al., 2020)). As discussed in Remark 2, because we cannot learn $\mathbf{\Sigma_\beta}$ perfectly in Algorithm 1, the smallest risk with finite data happens when the chosen representation dimension $R < d$, unlike the case with known $\mathbf{\Sigma_\beta}$.

5. Conclusion

In this paper, we study the sample efficiency of meta-learning with linear representations, motivated by the wide application of meta learning with overparameterized neural networks. On a theoretical level, we show that in representation learning, one can learn the representation with even fewer data if the features follow a spiked distribution and align with task parameters. We also propose an estimator that learns with optimal sample size, improving over prior works. Then we propose the optimal shaped representation which is typically overparameterized. We analyze its robustness while optimal representation is unattainable due to finite samples. Finally we propose an end to end learning guarantee of the overall meta-learning procedures.

References

Anderson, T. W. et al. Estimation of covariance matrices which are linear combinations or whose inverses are linear combinations of given matrices. *Essays in probability and statistics*, pp. 1–24, 1970.

Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332, 2019.

Balcan, M.-F., Blum, A., and Vempala, S. Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory*.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 2020.

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

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pp. 12–58, 2014.

Caruana, R. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.

Cavallanti, G., Cesa-Bianchi, N., and Gentile, C. Linear algorithms for online multitask classification. *The Journal of Machine Learning Research*, 11:2901–2934, 2010.

Chang, X., Li, Y., Oymak, S., and Thrampoulidis, C. Provable benefits of overparameterization in model compression: From double descent to pruning neural networks. *arXiv preprint arXiv:2012.08749*, 2020.

Chen, S., Li, J., and Song, Z. Learning mixtures of linear regressions in subexponential time via fourier moments. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 587–600, 2020.

Davis, C. and Kahan, W. M. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Kong, W., Somani, R., Kakade, S., and Oh, S. Robust meta-learning for mixed linear regression with small batches. *arXiv preprint arXiv:2006.09702*, 2020a.

Kong, W., Somani, R., Song, Z., Kakade, S., and Oh, S. Meta-learning for mixed linear regression. *arXiv preprint arXiv:2002.08936*, 2020b.

Li, Y. and Liang, Y. Learning mixtures of linear regressions with nearly optimal complexity. In *Conference On Learning Theory*, pp. 1125–1144, 2018.

Lounici, K., Pontil, M., Van De Geer, S., Tsybakov, A. B., et al. Oracle inequalities and optimal inference under group sparsity. *The annals of statistics*, 39(4):2164–2204, 2011.

Maurer, A., Pontil, M., and Romera-Paredes, B. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

Mei, S. and Montanari, A. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

Nakkiran, P., Venkat, P., Kakade, S., and Ma, T. Optimal regularization can mitigate double descent. *arXiv preprint arXiv:2003.01897*, 2020.

Oymak, S., Fabian, Z., Li, M., and Soltanolkotabi, M. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Thrampoulidis, C., Abbasi, E., and Hassibi, B. Lasso with non-linear measurements is equivalent to one with linear measurements. *Advances in Neural Information Processing Systems*, 28:3420–3428, 2015.

Tripuraneni, N., Jin, C., and Jordan, M. I. Provable meta-learning of linear representations. *arXiv preprint arXiv:2002.11684*, 2020.

Wu, D. and Xu, J. On the optimal weighted ℓ_2 regularization in overparameterized linear regression, 2020.

Zhong, K., Jain, P., and Dhillon, I. S. Mixed linear regression with multiple components. In *Advances in neural information processing systems*, pp. 2190–2198, 2016.