Towards Sample-efficient Overparameterized Meta-learning

Abstract

An overarching goal in machine learning is to build a generalizable model with a small number of samples. To this end, overparameterization has been the subject of immense interest to explain the generalization ability of deep nets even when the size of the dataset is smaller than that of the model. While prior literature focuses on the classical supervised setting, this paper aims to demystify overparameterization for meta-learning. Here we have a sequence of linear-regression tasks and we ask: (1) Given earlier tasks, what is the optimal linear representation of features for a new downstream task? and (2) How many samples do we need to build this representation? This work shows that surprisingly, overparameterization arises as a natural answer to these fundamental meta-learning questions. Specifically, for (1), we first show that learning the optimal representation coincides with the problem of designing a task-aware regularization to promote inductive bias. This inductive bias explains how the downstream task actually benefits from overparameterization, in contrast to prior works on few-shot learning. For (2), we develop a theory to explain how feature covariance can implicitly help reduce the sample complexity well below the degrees of freedom and lead to small estimation error. We then integrate these findings to obtain an overall performance guarantee for our meta-learning algorithm. Numerical experiments on real and synthetic data verify our insights on overparameterized 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. Constructing a simple yet informative representation of features is a critical component of learning a model that generalizes well to an unseen test set. The field of meta-learning dates back to [8, 4] and addresses this challenge by transferring insights across distinct but related tasks. Usually, the meta-learner first (1) learns a feature-representation from previously seen tasks and then (2) uses this representation to succeed at an unseen task. The first phase is called representation learning and the second is called few-shot learning. Such information transfer between tasks is the backbone of modern transfer and multitask learning and finds ubiquitous applications in image classification [14], machine translation [6] and reinforcement learning [17].

Recent literature in ML theory has posited that overparameterization can be beneficial to generalization in traditional single-task setups for both regression [27, 37, 6, 31, 28] and classification [30, 29] problems. Empirical literature in deep learning suggests that overparameterization is of interest for both phases of meta-learning as well. Deep networks are stellar representation learners despite containing many more parameters than the sample size. Additionally, overparameterization
We highlight that obtaining the representation consists of two steps: first the learner projects \( x \) with \( \Lambda \) from earlier tasks to choose the model that minimizes few-shot risk. This goal enables us to engage with important questions regarding overparameterization: 

**Q1:** What should the size \( R \) and the representation \( \Lambda \) be to minimize risk at the few-shot phase? 
**Q2:** Can we learn the \( Rd \) dimensional representation \( \Lambda \) with \( N \ll Rd \) samples?

However, existing analysis of subspace-based meta-learning algorithms for both the representation learning and few-shot phases of linear models have typically focused on the classical underparameterized regime. These works (see Paragraphs 2-3 of Sec. 1.2) consider the case where representation learning involves projection onto a lower-dimensional subspace. On the other hand, recent works on double descent shows that an overparameterized interpolator beats PCA-based method, to build upon these results to develop a theoretical understanding of overparameterized meta-learning.

1.1 Our contributions

This paper studies meta-learning when each task is a linear regression problem, similar in spirit to [35][22]. In the representation learning phase, the learner is provided with training data from \( T \) distinct tasks, with \( n_1 \) training samples per task: using this data, it selects a matrix \( \Lambda \in \mathbb{R}^{d \times R} \) with arbitrary \( R \) to obtain a linear representation of features via the map \( x \rightarrow \Lambda^\top x \). In the few-shot learning phase, the learner faces a new task with \( n_2 \) training samples and aims to use the representation \( \Lambda^\top x \) to aid prediction performance.

We highlight that obtaining the representation consists of two steps: first the learner projects \( x \) onto \( R \) basis directions, and then performs eigen-weighting of each of these directions, as shown in Figure 2(b). The overarching goal of this paper is to propose a scheme to use the knowledge gained from earlier tasks to choose \( \Lambda \) that minimizes few-shot risk. This goal enables us to engage with important questions regarding overparameterization:

We study meta-learning when each task is a linear regression problem, similar in spirit to [35][22]. In the representation learning phase, the learner is provided with training data from \( T \) distinct tasks, with \( n_1 \) training samples per task: using this data, it selects a matrix \( \Lambda \in \mathbb{R}^{d \times R} \) with arbitrary \( R \) to obtain a linear representation of features via the map \( x \rightarrow \Lambda^\top x \). In the few-shot learning phase, the learner faces a new task with \( n_2 \) training samples and aims to use the representation \( \Lambda^\top x \) to aid prediction performance.

We highlight that obtaining the representation consists of two steps: first the learner projects \( x \) onto \( R \) basis directions, and then performs eigen-weighting of each of these directions, as shown in Figure 2(b). The overarching goal of this paper is to propose a scheme to use the knowledge gained from earlier tasks to choose \( \Lambda \) that minimizes few-shot risk. This goal enables us to engage with important questions regarding overparameterization:

**Q1:** What should the size \( R \) and the representation \( \Lambda \) be to minimize risk at the few-shot phase? 
**Q2:** Can we learn the \( Rd \) dimensional representation \( \Lambda \) with \( N \ll Rd \) samples?

Figure 1: **Illustration of the benefit of overparameterization in the few-shot phase.** (a) Double-descent in transfer learning: dashed lines indicate the location where the number of features \( R \) exceed the number of training points; i.e., the transition from under to over-parameterization. The experimental details are contained in the supplement. (b) Illustration of the benefit of using Weighted minL2-interpolation in Definition 3 (blue). See Remark 1 for details and discussion.
The answers to the questions above will shed light on whether overparameterization is beneficial in few-shot learning and representation learning respectively. Towards this goal, we make several contributions to the finite-sample understanding of linear meta-learning, under assumptions discussed in Section 2. Our results are obtained for a general data/task model with arbitrary task covariance $\Sigma_T$ and feature covariance $\Sigma_F$, which allows for a rich set of observations.

**Optimal representation for few-shot learning.** As a stepping stone towards the goal of characterizing few-shot risk for different $\Lambda$, in Section 3 we first consider learning with known covariances $\Sigma_T$ and $\Sigma_F$ respectively (Algorithm 1). Compared to projection-only representations in previous works (see Paragraphs 2-3 of Sec. 1.2), our scheme applies eigen-weighting matrix $\Lambda^*$ to incentivize the optimizer to place higher weight on promising eigen-directions. This eigen-weighting procedure has been shown in the single-task case to be extremely crucial to avail the benefits of overparameterization [5, 28, 31]: it captures an inductive bias that promotes certain features and demotes others. We show that the importance of eigen-weighting extends to the multi-task case as well.

**Canonical task covariance.** Our analysis in Section 3 also reveals that, the optimal subspace and representation matrix are closed-form functions of the canonical task covariance $\tilde{\Sigma}_T = \Sigma_F^{1/2} \Sigma_T \Sigma_F^{1/2}$, which captures the feature saliency by summarizing the feature and task distributions.

**Representation learning.** In practice, task and feature covariances (and hence the canonical covariance) are rarely known apriori. However, we can estimate the principal subspace of the canonical task covariance $\Sigma_T$ (which has a degree of freedom (DoF) of $\Omega(\text{Rd})$) from data. In Section 4, we first present empirical evidence that feature covariance $\Sigma_F$ is “positively correlated” with $\Sigma_T$. Then we propose an efficient algorithm based on Method-of-Moments (MoM), and show that the sample complexity of representation learning is well below $O(\text{Rd})$ due to the inductive bias. Our sample complexity bound depends on interpretable quantities such as effective ranks $\Sigma_F, \Sigma_T$ and improves over prior art (e.g., [22, 35]), even though the prior works were specialized to low-rank $\Sigma_T$ and identity $\Sigma_F$ (see Table 2).

**End to end meta-learning guarantee.** In Section 5, we consider the generalization of Section 3 where we have only estimates of the covariances instead of perfect knowledge. This leads to an overall meta-learning guarantee in terms of $\Lambda^*$, $N$ and $n_2$ and uncovers a bias-variance tradeoff: As $N$ decreases, it becomes more preferable to use a smaller $R$ (more bias, less variance) due to inaccurate estimate of the weak eigen-directions of $\Sigma_T$. In other words, we find that overparameterization is only beneficial for few-shot learning if the quality of representation learning is sufficiently good. This explains why, in practice, increasing the representation dimension may not help reduce few-shot risk beyond a certain point (see Fig. 5).

### 1.2 Related work

**Overparameterized ML and double-descent** The phenomenon of double-descent was first discovered by [5]. This paper and subsequent works on this topic [3, 31, 30, 28, 10] emphasize the importance of the right prior (sometimes referred to as inductive bias or regularization) to avail the benefits of overparameterization. However, an important question that arises is: where does this prior come from? Our work shows that the prior can come from the insights learned from related previously-seen tasks. Section 3 extends the ideas in [32, 37] to depict how the optimal representation described can be learned from imperfect covariance estimates as well.

**Theory for representation learning** Recent papers [22, 21, 35, 15] propose the theoretical bounds of representation learning when the tasks lie in an exactly $r$ dimensional subspace. [22, 21, 35] discuss method of moment estimators and [35, 15] discuss matrix factorized formulations. [35] shows that the number of samples that enable meaningful representation learning is $O(\text{dr}^2)$. [22, 21, 35] assume the features follow a standard normal distribution. We define a canonical covariance which
Figure 2: (a) Steps of the meta-learning algorithm. (b) Our representation-learning algorithm has two steps: projection and eigen-weighting. We focus on the use of overparameterization+weighting matrix (Def. 3), and compare this with overparameterization with simple projection (no eigen-weighting), and underparameterization (for which eigen-weighting has no impact and is equivalent to projection). \[35, 22, 21, 15\] study underparameterized projections only. To distinguish from eigen-weighting, we will refer to simple projections as subspace-based representations.

handles arbitrary feature and task covariances. We also show that our estimator succeeds with $O(dr)$ samples when $n_1 \sim r$, and extend the bound to general covariances with effective rank defined.

**Subspace-based meta learning** With tasks being low rank, \[22, 21, 35, 18, 15\] do few-shot learning in a low dimensional space. \[38, 39\] study meta-learning for linear bandits. \[25\] gives information theoretic lower and upper bounds. \[7\] proposes subspace-based methods for nonlinear problems such as classification. We investigate a representation with arbitrary dimension, specifically interested in overparameterized case and show it yields a smaller error with general task/feature covariances. Related work \[15\] provides results on overparameterized representation learning, but \[15\] requires number of samples per pre-training task to obey $n_1 \gtrsim d$, whereas our results apply as soon as $n_1 \gtrsim 1$.

**Mixed Linear Regression (MLR)** In MLR \[40, 23, 11\], multiple linear regression are executed, similar to representation learning. The difference is that, the tasks are drawn from a finite set, and number of tasks can be larger than $d$ and not necessarily low rank. \[24, 9, 26\] propose sample complexity bounds of representation learning for mixed linear regression. They can be combined with other structures such as binary task vectors \[2\] and sparse task vectors \[1\].

## 2 Problem Setup

The problem we consider consists of two phases:

1. Representation learning: Prior tasks are used to learn a suitable representation to process features.
2. Few-shot learning: A new task is learned with a few samples by using the suitable representation.

This section defines the key notations and describes the data generation procedure for the two phases. In summary, we study linear regression tasks, the features and tasks are generated randomly, i.i.d. from their associated distributions $D_T$ and $D_F$, and the two phases share the same feature and task distributions. The setup is summarized in Figure 2(a).

### 2.1 Data generation

**Definition 1 (Task and feature distributions)** Throughout, $D_T$ and $D_F$ denote the distributions of tasks $\beta_i$ and features $x_{ij}$ respectively. These distributions are subGaussian, zero-mean with corresponding covariance matrices $\Sigma_T$ and $\Sigma_F$.

**Definition 2 (Data distribution for a single task)** Given a specific realization of task vector $\beta \sim D_T$, the corresponding label/input distribution $(y, x) \sim D_\beta$ is obtained via $y = x^T \beta + \varepsilon$ where $x \sim D_F$ and $\varepsilon$ is zero-mean subgaussian noise with variance $\sigma^2$. 
Data for Representation Learning (Phase 1). We have $T$ tasks, each with $n_1$ training examples. The task vectors $(\beta_i)_{i=1}^T \subset \mathbb{R}^d$ are drawn i.i.d. from the distribution $D_T$. The data for $i$th task is given by $(y_{ij}, x_{ij})_{n_{ij} \sim D_{\beta_i}}$. In total, there are $N = T \times n_1$ examples.

Data for Few-Shot Learning (Phase 2). Sample task $\beta_i \sim D_T$. Few-shot dataset has $n_2$ examples $(y_i, x_i)_{n_2 \sim D_{\beta_i}}$.

We use representation learning data to learn a representation of feature-task distribution, called eigen-weighting matrix $\Lambda$ in Def. 3 below. The matrix $\Lambda$ is passed to few-shot learning stage, helping learn $\beta_i$ with few data.

2.2 Training in Phase 2

We will define a weighted representation, called eigen-weighting matrix, and show how it is applied for few-shot learning. The matrix is learned during representation learning using the data from the $T$ tasks. Denote $X \in \mathbb{R}^{n_2 \times d}$ whose $i$th row is $x_i$, and $y = [y_1, ..., y_m]^\top$. We are interested in studying the weighted 2-norm interpolator defined below for overparameterization regime $R \geq n_2$.

Definition 3 (Eigen-weighting matrix and Weighted $\ell_2$-norm interpolator) Let the representation dimension be $R$, where $R$ is any integer between 1 and $d$. We define an eigen-weighting matrix $\Lambda \in \mathbb{R}^{d \times R}$ and the associated weighted $\ell_2$-norm interpolator

$$\hat{\beta}_\Lambda = \arg \min_{\beta} \|\Lambda^\top \beta\|_2 \text{ s.t. } y = X\beta \text{ and } \beta \in \text{range space}(\Lambda).$$

The solution is equivalent to defining $\hat{\alpha}_\Lambda = \Lambda^\top \hat{\beta}_\Lambda$ and solving an unweighted minimum 2-norm regression with features $X\Lambda$. This corresponds to our few-shot learning problem

$$\hat{\alpha}_\Lambda = \arg \min_{\alpha} \|\alpha\|_2 \text{ s.t. } y = X\Lambda \alpha$$

from which we obtain $\hat{\beta}_\Lambda = \Lambda \hat{\alpha}_\Lambda$. When there is no confusion, we can replace $\hat{\beta}_\Lambda$ with $\hat{\beta}$. One can easily see that $\hat{\beta} = \Lambda^\top (X\Lambda)^\top y$. We note that Definition 3 is a special case of the weighted ridge regression discussed in [37], as stated in Observation 1. An alternative equivalence between min-norm interpolation and ridge regression can be found in [31].

Observation 1 Let $X \in \mathbb{R}^{n_2 \times d}$ and $y \in \mathbb{R}^{n_2}$, define

$$\hat{\beta}_1 = \lim_{t \to 0} \arg \min_{\beta} \|X\beta - y\|_2^2 + t\beta^2 (\Lambda\Lambda^\top)^\top \beta, \beta \in \text{column space of } \Lambda.$$  (2.1)

We have that $\hat{\beta}_1 = \hat{\beta}$.

3 Canonical Covariance and Optimal Representation

In this section, we ask the simpler question: if the covariances $\Sigma_T$ and $\Sigma_F$ are known, what is the best choice of $\Lambda$ to minimize the risk of the interpolator from Definition 3? In general, the covariances are not known; however, the insights from this section help us study the more general case in Section 5. Define the risk as the expected error of inferring the label on the few-shot dataset,

$$\text{risk}(\Lambda, \Sigma_T, \Sigma_F) = E_{x,y,\beta} (y - x^\top \hat{\beta}_\Lambda)^2 = E_{\beta} (\hat{\beta}_\Lambda - \beta)^\top \Sigma_F (\hat{\beta}_\Lambda - \beta) + \sigma^2.$$  (3.1)

The natural choice of optimization for choosing $\Lambda$ would be to choose the weighting that minimizes the eventual risk of the learned interpolator,

$$\Lambda^* = \arg \min_{\Lambda \in \mathbb{R}^{d \times R}} \text{risk}(\Lambda^*, \Sigma_T, \Sigma_F)$$  (3.2)

Since the label $y$ is bilinear in $x$ and $\beta$, we introduce whitened features $\check{x} = \Sigma_p^{-1/2} x$ and associated task vector $\check{\beta} = \Sigma_p^{-1/2} \beta$. This change of variables ensures $x^T \beta = \check{x}^T \check{\beta}$; now, the task covariance in the transformed coordinates takes the form

$$\Sigma_T = \Sigma_p^{1/2} \Sigma_T \Sigma_p^{-1/2},$$

which we call the canonical task covariance; it captures the joint behavior of feature and task covariances $\Sigma_F, \Sigma_T$. Below, we observe that the risk in Equation (3.1) is invariant to the change of co-ordinates that we have described above i.e it does not change when $\Sigma_p^{1/2} \Sigma_T \Sigma_p^{-1/2}$ is fixed and we vary $\Sigma_F$ and $\Sigma_T$. 

5
Algorithm 1 Constructing the optimal representation

Require: Projection dimension $R$, noise level $\sigma$, canonical covariance $\Sigma_T$, task covariance $\Sigma_F$.

1: function COMPUTE_OPTIMAL_REP($R$, $\Sigma_F$, $\Sigma_T$, $\sigma$, $n_2$)
2: \hspace{1em} $U_1, \Sigma_P, \Sigma_T, \sigma_P = \text{COMPUTE_REDUCTION}(R, \Sigma_F, \Sigma_T, \sigma)$
3: \hspace{1em} Optimization: Get $\theta^*$ from $\text{OPTIMAL_REP}$.
4: \hspace{1em} Map to eigenvalues: Set diagonal $\Lambda_R \in \mathbb{R}^{R \times R}$ with entries $\Lambda_R^{*,i} = (1/\theta_i^* - 1)^{-2}$.
5: \hspace{1em} Lifting and feature whitening: $\Lambda^* \leftarrow U_1 (\Sigma_P^{-1/2} R)^{-1/2} \Lambda_R^*$.
6: \hspace{1em} return $\Lambda^*$
7: function COMPUTE_REDUCTION($R$, $\Sigma_F$, $\Sigma_T$, $\sigma$)
8: \hspace{1em} Get eigen-decomposition $\Sigma_T = U \Sigma_T U^\top$.
9: \hspace{1em} Principal eigenspace $U_1 \in \mathbb{R}^{d \times R}$ is the first $R$ columns of $U$.
10: \hspace{1em} Equivalent noise level: $\sigma_2^* \leftarrow \sigma^2 + \text{tr}(\Sigma_T) - \text{tr}(\Sigma_P^{1/2})$.
11: \hspace{1em} return $U_1, \Sigma_P^{1/2}, \Sigma_T^{1/2}, \sigma_R$.

Observation 2 (Equivalence to problem with whitened features) Let data be generated as in Phase 1. Denote $\Sigma_T = \Sigma_F^{1/2} \Sigma_T \Sigma_F^{-1/2}$. Then $\text{risk}(\Sigma_F^{-1/2} \Lambda, \Sigma_T, \Sigma_F) = \text{risk}(\Lambda, \Sigma_T, I)$.

This observation can be easily verified by substituting the change-of-coordinates into Equation (3.1) and evaluating the risk.

The risk in (3.1) quantifies the quality of representation $\Lambda$; however it is not a manageable function of $\Lambda$ that can be straightforwardly optimized. In this subsection, we show that it is asymptotically equivalent to a different optimization problem, which can be easily solved by analyzing KKT optimality conditions. Theorem [1] characterizes this equivalence; the COMPUTE_REDUCTION subroutine of Algorithm [1] calculates key quantities that are used in specifying the reduction, and the COMPUTE_OPTIMAL_REP subroutine of Algorithm [1] uses the solution of the simpler problem to obtain a solution for the original.

Assumption 1 (Bounded feature covariance) There exist positive constants $\Sigma_{\min}, \Sigma_{\max}$ such that $0 < \Sigma_{\min} I \preceq \Sigma_F \preceq \Sigma_{\max} I$.

Assumption 2 (Joint diagonalizability) $\Sigma_F$ and $\Sigma_T$ are diagonal matrices.

Assumption 3 (Double asymptotic regime) We let the dimensions and the sample size grow as $d, R, n_2 \to \infty$ at fixed ratios $\bar{d} := d/n_2$ and $\bar{R} := R/n_2$.

Assumption 4 The joint empirical distribution of the eigenvalues of $\Lambda_R$ and $\tilde{\Sigma}_F^{1/2}$ is given by the average of Dirac $\delta$‘s: $\frac{1}{R} \sum_{i=1}^R \delta_{\Lambda_R^{*,i}} = \sqrt{\pi} \tilde{\Sigma}_F^{1/2}$. It converges to a fixed distribution as $d \to \infty$.

With these assumptions, we can derive an analytical expression to quantify the risk of a representation $\Lambda$. We will then optimize this analytic expression to obtain a formula for the optimal representation.

Theorem 1 (Asymptotic risk equivalence) Suppose Assumptions [1][2][3][4] hold. Let $\xi > 0$ be the unique number obeying $n_2 = \sum_{i=1}^R (1 + (\xi \Lambda_i^2)^{-1})^{-1}$. Define $\theta \in \mathbb{R}^R$ with entries $\theta_i = \frac{\xi \Lambda_i^2}{1 + \xi \Lambda_i^2}$ and calculate $\tilde{\Sigma}_F^{1/2}, \sigma_F$ using the COMPUTE_REDUCTION procedure of Algorithm [1]. Then, define the asymptotic risk formula

$$f(\theta, \tilde{\Sigma}_F^{1/2}, n_2) = \frac{1}{n_2 - \|\theta\|_2^2} \left( n_2 \sum_{i=1}^R (1 - \theta_i)^2 \tilde{\Sigma}_F^{1/2} + \|\theta\|_2^2 + 1 \sigma_F^2 \right).$$

(3.3)

We have that

$$\lim_{n_2 \to \infty} f(\theta, \tilde{\Sigma}_F^{1/2}, n_2) = \lim_{n_2 \to \infty} \text{risk}(\Sigma_F^{-1/2} \Lambda, \Sigma_T, \Sigma_F)$$

(3.4)

The proof of Theorem [1] applies the convex Gaussian Min-max Theorem (CGMT) in [34] and can be found in the Appendix B.2. We show that as dimension grows, the distribution of the estimator $\beta$ converges to a Gaussian distribution and we can calculate the expectation of risk.

This is equivalent to the more general scenario where $\Sigma_F$ and $\Sigma_T$ are jointly diagonalizable.
Theorem 1 provides us with a closed-form risk for any linear representation. Now, one can solve for the optimal representation by computing (OPT-REP) below. In order to do this, we propose an algorithm for the optimization problem in Appendix B.5 via a study of the KKT conditions for the problem.  

\[
\theta^* = \arg \min_{\theta} f(\theta, \Sigma_T, \Sigma_F), \text{ s.t. } 0 < \theta < 1, \sum_{i=1}^{R} \theta_i = n_2
\]  

\text{(OPT-REP)}

The optimal representation is \( \Lambda^*_{R,i} = (\frac{1}{\theta^*_i} - 1)\xi^{-2} \). The subroutine \text{COMPUTEOPTIMALREP} in Algorithm 1 summarizes this procedure.

**Remark** Thm. 1 states that \( \text{risk}(\Sigma_F^{1/2} \Lambda, \Sigma_T, \Sigma_F) \) can be arbitrarily well-approximated by \( f(\theta, \Sigma_T^{1/2}, n_2) \) if \( n_2 \) is sufficiently large. In Fig. 2(b), we set \( \Sigma_F = I_{100}, \Sigma_T = \text{diag}(I_{20}, I_{80}) \), \( n_2 = 40 \). The curves in Fig 2(b) are the finite dimensional approximation of \( f \) (LHS of (3.4)); the dots are empirical approximations of the risk (RHS of (3.4)). We tested two cases when \( \Lambda \) is the optimal eigen-weighting or projection matrix with no weighting. Our theorem is corroborated by the observation that the dots and curves are visibly very close. The approximation is already accurate for the finite dimensional problem with just \( n_2 = 40 \).

**The benefit of overparameterization**. Theorem 1 leads to an optimal eigen-weighting strategy via asymptotic analysis. In Figure 3, we plot the effect on the risk of increasing \( R \) for different shapes of task covariance; the parameter \( \iota \) controls how spiked \( \Sigma_T \) is, with a smaller value for \( \iota \) indicating increased spiked-ness. For the underparameterized problem, the weighting does not have any impact on the risk. In the overparameterized regime, the eigen-weighted learner achieves lower few-shot error than its unweighted (\( \Lambda = I \)) counterpart, showing that eigen-weighting becomes critical.

The eigen-weighting procedure can introduce inductive bias during few-shot learning, and helps explain how optimal representation minimizing the few-shot error can be overparameterized with \( R \gg n_2 \). We note that, an \( R \) dimensional representation can be recovered by a \( d \) dimensional representation matrix of rank \( R \), thus the underparameterized case can never beat \( d \) dimensional case in theory. The error with optimal eigen-weighting in overparameterized regime is smaller than the respective underparameterized counterpart. The error is lower with smaller \( \iota \). It implies that, while \( \Sigma_T \) gets closer to low-rank, the excess error caused by choosing small dimension \( R \) (equal to the gap \( \sigma_R^2 - \sigma^2 \) in Algo 1) is not as significant.

Low dimensional representations zero out features and cause bias. By contrast, when \( \Sigma_T \in \mathbb{R}^{d \times d} \) is not low rank, every feature contributes to learning with the importance of the features reflected by the weights. This viewpoint is in similar spirit to that of [19] where the authors devise a misspecified linear regression to demonstrate the benefits of overparameterization. Our algorithm allows arbitrary representation dimension \( R \) and eigen-weighting.

**4 Representation Learning**

In this section, we will show how to estimate the useful distribution in representation learning phase that enables us to calculate eigen-weighting matrix \( \Lambda^* \). Note that \( \Lambda^* \) depends on the canonical covariance \( \Sigma_T = \Sigma_F^{1/2} \Sigma_T \Sigma_F^{1/2} \). Learning the \( R \)-dimensional principal subspace of \( \Sigma_T \) enables us to calculate \( \Lambda^* \). Denote this subspace by \( \mathcal{S}_T \).

\footnote{In Sec. 5, the constraint is \( 0 < \theta < 1 - \frac{d-n_2}{n_2} \theta \) for robustness concerns.}

\footnote{In the algorithm, \( \xi = 1 \) and \( \Lambda_{R,i} = (1/\theta^*_i - 1)^{-2} \), because \( e \Lambda^* \) for any constant \( e \) gives the same \( \hat{\theta} \).}

\footnote{We also need to estimate \( \Sigma_F \) for whitening. Estimating \( \Sigma_F \) is rather easy and incurs smaller error compared to \( \Sigma_T \). The analysis is provided in the first part of Appendix B.}
Subspace estimation vs. inductive bias. The subspace-based representation \( \hat{\Sigma}_T \) has degrees of freedom \( = Rd \). When \( \Sigma_T \) is exactly rank \( R \) and features are whitened, \([35]\) provides a sample-complexity lower bound of \( \Omega(Rd) \) examples and gives an algorithm achieving \( O(R^2d) \) samples. However, in practice, deep nets learn good representations despite overparameterization. In this section, recalling our Q2, we argue that the inductive bias of the feature distribution can implicitly accelerate learning the canonical covariance. This differentiates our results from most prior works such as \([22, 21, 35]\) in two aspects:

1. Rather than focusing on a low dimensional subspace and assuming \( N \gtrsim Rd \), we can estimate \( \hat{\Sigma}_T \) or \( \hat{S}_T \) in the overparameterized regime \( N \lesssim Rd \).
2. Rather than assuming whitened features \( \Sigma_F = I \) and achieving a sample complexity of \( R^2d \), our learning guarantee holds for arbitrary covariance matrices \( \Sigma_F, \Sigma_T \). The sample complexity depends on effective rank and can be arbitrarily smaller than DoF. We showcase our bounds via a spiked covariance setting in Example 1 below.

For learning \( \Sigma_T \) or its subspace \( \hat{S}_T \), we investigate the method-of-moments (MoM) estimator.

Definition 4 (MoM Estimator) For \( 1 \leq i \leq T \), define \( b_{i,1} = 2n_1^{-1} \sum_{j=1}^{n_1^2} y_{ij}x_{ij} \), \( b_{i,2} = 2n_1^{-1} \sum_{j=n_1^2+1}^{n_1^2+2} y_{ij}x_{ij} \). Set
\[
\hat{M} = n_1^{-1} \sum_{i=1}^{T} (b_{i,1}b_{i,2}^T + b_{i,2}b_{i,1}^T),
\]
The expectation of \( \hat{M} \) is equal to \( M = \Sigma_F \Sigma_T \Sigma_F \).

Inductive bias in representation learning: Recall that canonical covariance \( \hat{\Sigma}_T = \Sigma_F^{1/2} \Sigma_T \Sigma_F^{1/2} \) is the attribute of interest. However, feature covariance \( \Sigma_F^{1/2} \) term implicitly modulates the estimation procedure because the population MoM is not \( \hat{\Sigma}_T \) but \( M = \Sigma_F^{1/2} \Sigma_T \Sigma_F^{1/2} \). For instance, when estimating the principle canonical subspace \( \hat{S}_T \), the degree of alignment between \( \Sigma_F \) and \( \Sigma_T \) can make or break the estimation procedure: if \( \Sigma_F \) and \( \Sigma_T \) have well-aligned principal subspaces, \( \hat{S}_T \) will be easier to estimate since \( \Sigma_F \) will amplify the \( \hat{S}_T \) direction within \( M \).

We verify the inductive bias on practical image dataset, reported in Appendix A. We assessed correlation coefficient between covariances \( \Sigma_T, \Sigma_F \) via the canonical-feature alignment score defined as the correlation coefficient
\[
\rho(\Sigma_F, \hat{\Sigma}_T) := \frac{\langle \Sigma_F, \hat{\Sigma}_T \rangle}{\|\Sigma_F\|_F \|\hat{\Sigma}_T\|_F} = \frac{\text{trace}(M)}{\|\Sigma_F\|_F \|\hat{\Sigma}_T\|_F}.
\]
Observe that, the MoM estimator \( \hat{M} \) naturally shows up in the alignment definition because the inner product of \( \Sigma_T, \Sigma_F \) is equal to \( \text{trace}(M) \). This further supports our inductive bias intuition. As reference, we compared it to canonical-identity alignment defined as \( \frac{\text{trace}(\Sigma_T)}{\sqrt{\|\Sigma_F\|_F}} \) (replacing \( \Sigma_F \) with \( I \)). The canonical-feature alignment score is higher than the canonical-identity alignment score. This significant score difference exemplifies how \( \Sigma_F \) and \( \Sigma_T \) can synergistically align with each other (inductive bias). This alignment helps our MoM estimator defined below, illustrated by Example 1 (spiked covariance).

In the following subsections, let \( N = n_1 T \) refer to the total tasks in representation-learning phase. Let \( r_F = \text{tr}(\Sigma_F), \tilde{r}_T = \text{tr}(\Sigma_T), \) and \( \tilde{r}_T = \text{tr}(\Sigma_T) \). Define the approximate low-rankness measure of feature covariance by
\[ s_F = \min_{s_F'} s_{F'}, \text{ s.t. } s_{F'} \in \{1, \ldots, d\}, s_{F'}/d \geq \lambda_{s_F'+1}(\Sigma_F) \]
We have two results for this estimator.

1. Generally, we can estimate \( \hat{M} \) with \( O(r_F \tilde{r}_T^2) \) samples.
2. Let \( n_1 \geq s_T \), we can estimate \( \hat{M} \) with \( O(s_F \tilde{r}_T) \) samples.

Paper \([35]\) has sample complexity \( O(dr^2) \) (\( r \) is exact rank). Our sample complexity is \( O(r_F \tilde{r}_T^2) \). \( r_F, \tilde{r}_T \) can be seen as effective ranks and our bounds are always smaller than \([35]\). We will discuss later in Example 1. Our second result says when \( n_1 \geq s_T \), our sample complexity achieves the \( O(dr) \) which is proven a lower bound in \([35]\).

\(^{5}\) The \((s_F + 1)\)-th eigenvalue is smaller than \( s_F/d \). Note the top eigenvalue is 1.
Right side: Sample complexity and error of MoM estimators. $s_F, s_T$ is the dimension of the principal eigenspace of the feature (task) covariance. $r_F = s_F + t_F(d - s_F), r_T = s_T + t_T(d - s_T)$ are the effective ranks. Left side: This is the well-studied setting of identity feature covariance and low-rank task covariance. Our bound in the second row is the first result to achieve optimal sample complexity of $O(ds_T)$ (cf. [35, 22]).

**Theorem 2** Let data be generated as in Phase 1. Assume $\|\Sigma_F\|, \|\Sigma_T\|$ = 1 for normalization.\footnote{This is simply equivalent to scaling $y_{ij}$, which does not affect the normalized error $\|\tilde{M} - M\|/\|M\|$. In the appendix we define $S = \max(\{\|\Sigma_F\|, \|\Sigma_T\|\})$ and prove the theorem for general $S$.}

1. Let $n_1$ be an even number. Then with probability at least $1 - N^{-100}$,

$$\|\hat{M} - M\| \lesssim (\hat{r}_T + \sigma^2) \sqrt{r_F \over N} + \sqrt{r_T \over T}.$$  

2. Assume $T \geq s_F$. If $n_1 \geq \hat{r}_T + \sigma^2$, then with probability at least

$$\|\hat{M} - M\| \lesssim (\hat{r}_T + \sigma^2) / n_1^{1/2}.$$  

Denote the top-$R$ principal subspaces of $M, \hat{M}$ by $M_{top}, \hat{M}_{top}$ and assume the eigen-gap condition $\lambda_R(M) - \lambda_{R+1}(M) > 2\|M - \hat{M}\|$. Then a direct application of Davis-Kahan Theorem [12] bounds the subspace angle as follows

$$\text{angle}(M_{top}, \hat{M}_{top}) \lesssim \|\hat{M} - M\| / (\lambda_R(M) - \lambda_{R+1}(M)).$$

**Example 1 (Spiked $\hat{\Sigma}_T$, Aligned principal subspaces)**

Suppose the spectra of $\Sigma_F$ and $\Sigma_T$ are bimodal as follows

$$\Sigma_F = \text{diag}(I_{s_F}, t_F I_{d-s_F}), \Sigma_T = \text{diag}(I_{s_T}, t_T I_{d-s_T}).$$

Set statistical error $\text{Err}_{T,N} := \sqrt{r_T^2 t_F / N} + \sqrt{r_F t_T / T}.$ When $t_T, t_F < 1$, $s_F \geq s_T$, the recovery error of $\hat{\Sigma}_T$ and its principal subspace $\hat{S}_T$ are bounded as

$$\text{angle}(M_{top}, \hat{S}_T) \lesssim \text{Err}_{T,N} + t_T t_F$$  

and  

$$\|\hat{M} - \hat{\Sigma}_T\| \lesssim \text{Err}_{T,N} + t_T t_F.$$  

The estimation errors for $\hat{\Sigma}_T, \hat{S}_T$ are controlled in terms of the effective ranks and the spectrum tails $t_F, t_T$. Typically $s_F s_T \approx n_1$ so $\sqrt{r_T^2 t_F / N}$ term dominates the statistical error in practice. In Fig. 4 we plot the error of estimating $\hat{M}$ (whose principal subspace coincides with $\hat{\Sigma}_T$). $\Sigma_F = \text{diag}(I_{s_0}, t_F I_{d-s_0}), \Sigma_T = \text{diag}(I_{s_0}, t_T I_{d-s_T}). T = N = 100$. We can see that the error increase with $t_F$.

5 Robustness of Optimal Representation and Overall Meta-Learning Bound

In Section 3 we described the algorithm for computing the optimal representation with known distributions of features and tasks. In Section 4 we proposed the MoM estimator in representation learning phase to estimate the unknown covariance matrices. In this section, we study the algorithm’s behaviors when we calculate $\hat{X}$ using the estimated canonical covariance, rather than the full-information setting of Section 3.
Armed with the provably reliable estimators of Section 4, we can replace \( \tilde{\Sigma}_T \) and \( \Sigma_F \) in Algorithm 1 with our estimators. In this section, we inquire: how does the estimation error in covariance-estimation in representation learning stage affect the downstream few-shot learning risk? That says, we are interested in:

\[
\text{risk}(\Lambda, \Sigma_T, \Sigma_F) - \text{risk}(\Lambda^*, \Sigma_T, \Sigma_F).
\]

Let us replace the constraint in \( \text{OPT-REP} \) by \( \theta \leq \hat{\theta} \leq 1 - \frac{d-n_2}{n_2^2} \). This changes the “optimization” step in Algorithm 1. Theorem 3 does not require an explicit computation of the optimal representation by enforcing \( \hat{\theta} \). Instead, we use the robustness of such a representation (due to its well-conditioned nature) to deduce its stability. That said, for practical computation of optimal representation, we simply use Algorithm 1. We can then evaluate \( \hat{\theta} \) after-the-fact as the minimum singular value of this representation to apply Theorem 3 without assuming an explicit \( \hat{\theta} \).

Let \( \Lambda_\theta(R) = \text{COMPUTE-OPTIMAL-REP}(R, \Sigma_F, M, \sigma, n_2) \) denote the estimated optimal representation and \( \Lambda_\theta^*(R) = \text{COMPUTE-OPTIMAL-REP}(R, \Sigma_F, \tilde{\Sigma}_T, \sigma, n_2) \) denote the true optimal representation, which cannot be accessed in practice. Below we present the bound of the whole meta-learning algorithm. It shows that a bounded error in representation learning leads to a bounded increase on the downstream few-shot learning risk, thus quantifying the robustness of few-shot learning to errors in covariance estimates.

**Theorem 3** Let \( \Lambda_\theta(R), \Lambda_\theta^*(R) \) be as defined above, and \( r_F = \text{tr}(\Sigma_F), r_T = \text{tr}(\tilde{\Sigma}_T), \hat{r}_T = \text{tr}(\Sigma_T) \). The risk of meta-learning algorithm satisfies:

\[
\text{risk}(\Lambda_\theta(R), \Sigma_T, \Sigma_F) - \text{risk}(\Lambda_\theta^*(R), \Sigma_T, \Sigma_F) \leq \frac{n_2^2}{d(R-n_2)(2n_2-R\hat{\theta})} \left( (\hat{r}_T + \sigma^2) \sqrt{\frac{F_T}{N}} + \sqrt{\frac{r_F}{T}} \right).
\]

Notice that as the number of previous tasks \( T \) and total representation-learning samples \( N \) observed increases, the risk of the estimated \( \Lambda_\theta(R) \) approaches that of the optimal \( \Lambda_\theta^*(R) \) as we expect. The result only applies to the overparameterized regime of interest \( R > n_2 \). The expression of risk in the underparameterized case is different, and covered by the second case of Equation (4.4) in [37]. We plot it in Fig. 1(b) on the left side of the peak as a comparison.

**Risk with respect to PCA level \( R \).** In Fig. 5, we plot the error of the whole meta-learning algorithm. We simulate representation learning and get \( M \), use it to compute \( \Lambda \) and plot the theoretical downstream risk (experiments match, see Fig. 1(b)). Mainly, we compare the behavior of Theorem 3 with different \( R \). When \( R \) grows, we search \( \Lambda \) in a larger space. The optimal \( \Lambda \) in a feasible subset is always no better than searching in a larger space, thus the risk decreases with \( R \) increasing. At the same time, representation learning error increases with \( R \) since we need to fit a matrix in a larger space. In essence, this result provides a theoretical justification on a sweet-spot for the optimal representation. \( d = R \) is optimal when \( N = \infty \), i.e., representation learning error is 0. As \( N \) decreases, there is a tradeoff between learning error and truncating small eigenvalues. Thus choosing \( R \) adaptively with \( N \) can strike the right bias-variance tradeoff between the excess risk (variance) and the risk due to suboptimal representation.

![Figure 5: End to end learning guarantees.](image)

6 Conclusion

In this paper, we study the sample efficiency of meta-learning with linear representations. We show that the optimal representation is typically overparameterized and outperforms subspace-based representations for general data distributions. We refine the sample complexity analysis for learning arbitrary distributions and show the importance of inductive bias of feature and task. Finally we provide an end-to-end bound for the meta-learning algorithm showing the tradeoff of choosing larger representation dimension v.s. robustness against representation learning error.

---

Note that Sec. 6 of [37] gives the exact value of \( \text{risk}(\Lambda^*, \Sigma_T, \Sigma_F) \) so we have an end to end error guarantee.

The bracketed expression applies first conclusion of Theorem 3. One can plug in the second as well.
Acknowledgements

This work is supported in part by the NSF TRIPODS II grant DMS 2023166, NSF TRIPODS CCF 1740551, NSF CCF-2046816, NSF CCF 2007036, Army Research Office grant W911NF-21-1-0312, and the Moorthy Professorship at UW ECE.

References


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In supplementary file

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]