Towards Sample-efficient Overparameterized Meta-learning

Yue Sun
University of Washington
yuesun@uw.edu

Adhyyan Narang
University of Washington
adhyyan@uw.edu

Halil Ibrahim Gulluk
Bogazici University
hibrahimgulluk@gmail.com

Samet Oymak
University of California, Riverside
oymak@ece.ucr.edu

Maryam Fazel
University of Washington
mfazel@uw.edu

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 [28, 38, 3, 32, 29] and classification [31, 30] 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

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

is observed to be beneficial in the few-shot phase for transfer-learning in Figure 1(a). A ResNet-50 network pretrained on Imagenet was utilized to obtain a representation of R features for classification on CIFAR-10. All layers except the final (softmax) layer are frozen and are treated as a fixed feature-map. We then train the final layer of the network for the downstream task which yields a linear classifier on pretrained features. The figure plots the effect of increasing R on the test error on CIFAR-10, for different choices of training size n_2. For each choice of n_2, increasing R beyond n_2 is seen to reduce the test-error. These findings are corroborated by [17] (MAML) and [37], who successfully use a transfer learning method that adapts a pre-trained model, with 112980 parameters, to downstream tasks with only 1-5 new training samples.

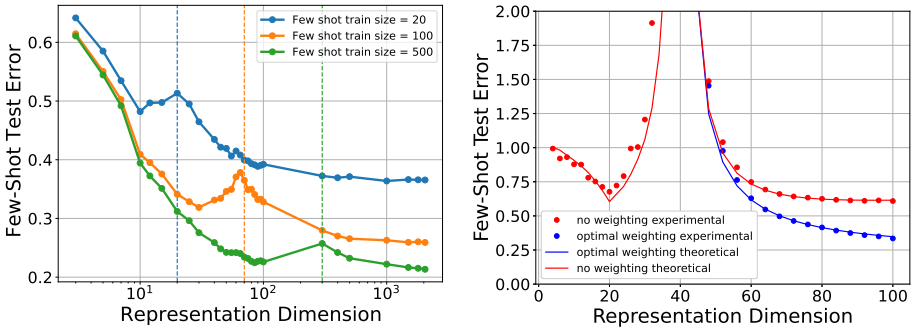

Figure 1: **Illustration of the benefit of overparameterization in the few-shot phase.** (a) Double-descent in transfer learning: dashed lines indicate the location where the number of features R exceed the number of training points; i.e., the transition from under to over-parameterization. The experimental 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.

In Figure 1(b), we consider a sequence of *linear* regression tasks and plot the few-shot error of our proposed projection and eigen-weighting based meta-learning algorithm for a fixed few-shot training size, but varying dimensionality of features. The resulting curve looks similar to Figure 1(a) and suggests that the observations regarding overparameterization for meta-learning in neural networks can, to a good extent, be captured by linear models, thus motivating their detailed study. This aligns with trends in recent literature: while deep nets are nonlinear, recent advances show that linearized problems such as kernel regression (e.g., via neural tangent kernel [20, 16, 23, 34, 12]) provide a good proxy to understand some of the theoretical properties of practical overparameterized deep nets.

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 [36, 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 $\mathbf{\Lambda} \in \mathbb{R}^{d \times R}$ with arbitrary R to obtain a linear *representation* of features via the map $\boldsymbol{x} \to \mathbf{\Lambda}^\top \boldsymbol{x}$. In the few-shot learning phase, the learner faces a new task with n_2 training samples and aims to use the representation $\mathbf{\Lambda}^\top \boldsymbol{x}$ to aid prediction performance.

We highlight that obtaining the representation consists of two steps: first the learner projects $\boldsymbol{x}$ 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 $\mathbf{\Lambda}$ 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 $\mathbf{\Lambda}$ be to minimize risk at the few-shot phase?

Q2: Can we learn the Rd dimensional representation $\mathbf{\Lambda}$ with $N \ll Rd$ samples?

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* $\mathbf{\Sigma}_\beta$ *and feature covariance* $\mathbf{\Sigma}_F$ 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 $\mathbf{\Lambda}$, in Section 3 we first consider learning with **known covariances** $\mathbf{\Sigma}_T$ and $\mathbf{\Sigma}_F$ 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 $\mathbf{\Lambda}^*$ 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 benefit of overparameterization [5, 29, 32]: 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* $\tilde{\mathbf{\Sigma}}_T = \mathbf{\Sigma}_F^{1/2}\mathbf{\Sigma}_T\mathbf{\Sigma}_F^{1/2}$, 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 $\tilde{\mathbf{\Sigma}}_T$ (which has a degree of freedom (DoF) of $\Omega(Rd)$) from data. In Section 4 we first present empirical evidence that feature covariance $\mathbf{\Sigma}_F$ is "positively correlated" with $\tilde{\mathbf{\Sigma}}_T$. Then we propose an efficient algorithm based on Method-of-Moments (MoM), and show that the sample complexity of representation learning is well below $\mathcal{O}(Rd)$ due to the inductive bias. Our sample complexity bound depends on interpretable quantities such as *effective ranks* $\mathbf{\Sigma}_F, \tilde{\mathbf{\Sigma}}_T$ and improves over prior art (e.g., [22, 36]), even though the prior works were specialized to low-rank $\tilde{\mathbf{\Sigma}}_T$ and identity $\mathbf{\Sigma}_F$ (see Table 2).

$\mathbf{\Sigma}_F$	Feature covariance
$\mathbf{\Sigma}_T$	Task covariance
$\tilde{\mathbf{\Sigma}}_T$	Canonical task covariance
n_1	Samples per each earlier task
T	Number of earlier tasks
N	Total sample size $T \times n_1$
n_2	Samples for new task
$\mathbf{\Lambda}$	Eigen-weighting matrix

Table 1: Main notation

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 $\mathbf{\Lambda}^*$, 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 $\tilde{\mathbf{\Sigma}}_T$. 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, 32, 31, 29, 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 [33, 38] to depict how the optimal representation described can be learned from imperfect covariance estimates as well.

Theory for representation learning Recent papers [22, 21, 36, 15] propose the theoretical bounds of representation learning when the tasks lie in an exactly r dimensional subspace. [22, 21, 36] discuss method of moment estimators and [36, 15] discuss matrix factorized formulations. [36] shows that the number of samples that enable meaningful representation learning is $\mathcal{O}(dr^2)$. [22, 21, 36] assume the features follow a standard normal distribution. We define a canonical covariance which

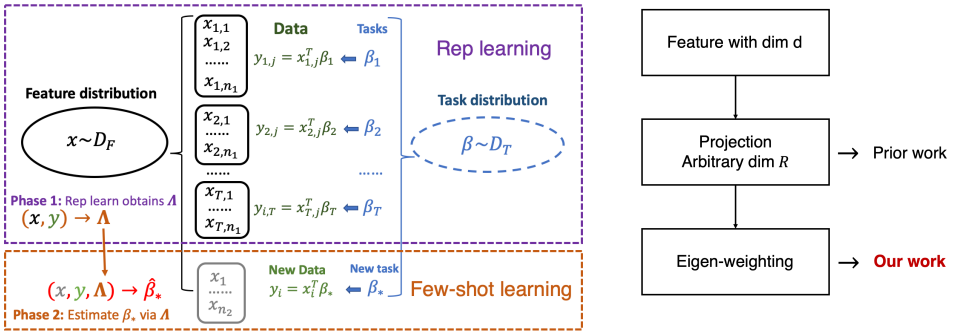

Figure 2: (a) Steps of the meta-learning algorithm. (b) Our representation-learning algorithm has two steps: projection and eigen-weighting. We focus on the use of overparameterization+weighting matrix (Def. 3), and compare this with overparameterization with simple projection (no eigen-weighting), and underparameterization (for which eigen-weighting has no impact and is equivalent to projection). [36, 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 $\mathcal{O}(dr)$ 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, 36, 18, 15] do few-shot learning in a low dimensional space. [39, 40] study meta-learning for linear bandits. [26] 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 [41, 24, 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. [25, 9, 27] 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 $\mathcal{D}_T$ and $\mathcal{D}_F$, 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, $\mathcal{D}_T$ and $\mathcal{D}_F$ denote the distributions of tasks $\boldsymbol{\beta}_i$ and features $\boldsymbol{x}_{ij}$ respectively. These distributions are subGaussian, zero-mean with corresponding covariance matrices $\boldsymbol{\Sigma}_T$ and $\boldsymbol{\Sigma}_F$.*

Definition 2 (Data distribution for a single task) *Given a specific realization of task vector $\boldsymbol{\beta} \sim \mathcal{D}_T$, the corresponding label/input distribution $(y, \boldsymbol{x}) \sim \mathcal{D}_{\boldsymbol{\beta}}$ is obtained via $y = \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon$ where $\boldsymbol{x} \sim \mathcal{D}_F$ and ε is zero-mean subgaussian noise with variance σ^2.*

Data for Representation Learning (Phase 1). We have T tasks, each with n_1 training examples. The task vectors $(\boldsymbol{\beta}_i)_{i=1}^T \subset \mathbb{R}^d$ are drawn i.i.d. from the distribution $\mathcal{D}_T$. The data for ith task is given by $(y_{ij}, \boldsymbol{x}_{ij})_{j=1}^{n_1} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\boldsymbol{\beta}_i}$. In total, there are $N = T \times n_1$ examples.

Data for Few-Shot Learning (Phase 2). Sample task $\boldsymbol{\beta}_\star \sim \mathcal{D}_T$. Few-shot dataset has n_2 examples $(y_i, \boldsymbol{x}_i)_{j=1}^{n_2} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\boldsymbol{\beta}_\star}$.

We use representation learning data to learn a representation of feature-task distribution, called eigen-weighting matrix $\boldsymbol{\Lambda}$ in Def. 3 below. The matrix $\boldsymbol{\Lambda}$ is passed to few-shot learning stage, helping learn $\boldsymbol{\beta}_\star$ 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 $\boldsymbol{X} \in \mathbb{R}^{n_2 \times d}$ whose i^{th} row is $\boldsymbol{x}_i$, and $\boldsymbol{y} = [y_1, ..., y_m]^\top$. 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 ℓ_2-norm interpolator) *Let the representation dimension be R, where R is any integer between 1 and d. We define an eigen-weighting matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times R}$ and the associated weighted ℓ_2-norm interpolator*

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\Lambda}^\dagger \boldsymbol{\beta}\|_2 \quad \textit{s.t.} \quad \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} \quad \textit{and} \quad \boldsymbol{\beta} \in \text{range_space}(\boldsymbol{\Lambda}).$$

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

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

from which we obtain $\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}\hat{\boldsymbol{\alpha}}_{\boldsymbol{\Lambda}}$. When there is no confusion, we can replace $\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}}$ with $\hat{\boldsymbol{\beta}}$. One can easily see that $\hat{\boldsymbol{\beta}} = \boldsymbol{\Lambda}(\boldsymbol{X}\boldsymbol{\Lambda})^\dagger \boldsymbol{y}$. We note that Definition 3 is a special case of the weighted ridge regression discussed in [38], as stated in Observation 1. An alternative equivalence between min-norm interpolation and ridge regression can be found in [32].

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

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

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

3 Canonical Covariance and Optimal Representation

In this section, we ask the simpler question: if the covariances $\boldsymbol{\Sigma}_T$ and $\boldsymbol{\Sigma}_F$ are known, what is the best choice of $\boldsymbol{\Lambda}$ 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,

$$\text{risk}(\boldsymbol{\Lambda}, \boldsymbol{\Sigma}_T, \boldsymbol{\Sigma}_F) = \boldsymbol{E}_{\boldsymbol{x}, y, \boldsymbol{\beta}} (y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}})^2 = \boldsymbol{E}_{\boldsymbol{\beta}} (\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}} - \boldsymbol{\beta})^\top \boldsymbol{\Sigma}_F (\hat{\boldsymbol{\beta}}_{\boldsymbol{\Lambda}} - \boldsymbol{\beta}) + \sigma^2. \tag{3.1}$$

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

$$\boldsymbol{\Lambda}^* = \arg\min_{\boldsymbol{\Lambda}' \in \mathbb{R}^{d \times R}} \text{risk}(\boldsymbol{\Lambda}', \boldsymbol{\Sigma}_T, \boldsymbol{\Sigma}_F) \tag{3.2}$$

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

$$\tilde{\boldsymbol{\Sigma}}_T = \boldsymbol{\Sigma}_F^{1/2} \boldsymbol{\Sigma}_T \boldsymbol{\Sigma}_F^{1/2},$$

which we call the **canonical task covariance**; it captures the joint behavior of feature and task covariances $\boldsymbol{\Sigma}_F, \boldsymbol{\Sigma}_T$. 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 $\boldsymbol{\Sigma}_F^{1/2} \boldsymbol{\Sigma}_T \boldsymbol{\Sigma}_F^{1/2}$ is fixed and we vary $\boldsymbol{\Sigma}_F$ and $\boldsymbol{\Sigma}_T$.

Algorithm 1 Constructing the optimal representation

Require: Projection dimension R, noise level σ, canonical covariance $\tilde{\boldsymbol{\Sigma}}_T$, task covariance $\boldsymbol{\Sigma}_F$.

function COMPUTEOPTIMALREP($R, \boldsymbol{\Sigma}_F, \tilde{\boldsymbol{\Sigma}}_T, \sigma, n_2$)
  $\boldsymbol{U}_1, \boldsymbol{\Sigma}_F^R, \tilde{\boldsymbol{\Sigma}}_T^R, \sigma_R$ = COMPUTEREDUCTION($R, \boldsymbol{\Sigma}_F, \tilde{\boldsymbol{\Sigma}}_T, \sigma$)
  Optimization: Get $\boldsymbol{\theta}^*$ from (OPT-REP).
  Map to eigenvalues: Set diagonal $\boldsymbol{\Lambda}_R^* \in \mathbb{R}^{R\times R}$ with entries $\boldsymbol{\Lambda}_{R,i}^* = (1/\boldsymbol{\theta}_i^* - 1)^{-2}$.
  Lifting and feature whitening: $\boldsymbol{\Lambda}^* \leftarrow \boldsymbol{U}_1(\boldsymbol{\Sigma}_F^R)^{-1/2}\boldsymbol{\Lambda}_R^*$.
  return $\boldsymbol{\Lambda}^*$

function COMPUTEREDUCTION($R, \boldsymbol{\Sigma}_F, \tilde{\boldsymbol{\Sigma}}_T, \sigma$)
  Get eigen-decomposition $\tilde{\boldsymbol{\Sigma}}_T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top$.
  Principal eigenspace $\boldsymbol{U}_1 \in \mathbb{R}^{d\times R}$ = the first R columns of $\boldsymbol{U}$.
  Top eigenvalues: Set $\tilde{\boldsymbol{\Sigma}}_T^R = \boldsymbol{U}_1^\top\tilde{\boldsymbol{\Sigma}}_T\boldsymbol{U}_1, \boldsymbol{\Sigma}_F^R = \boldsymbol{U}_1^\top\boldsymbol{\Sigma}_F\boldsymbol{U}_1$
  Equivalent noise level: $\sigma_R^2 \leftarrow \sigma^2 + \mathbf{tr}(\tilde{\boldsymbol{\Sigma}}_T) - \mathbf{tr}(\tilde{\boldsymbol{\Sigma}}_T^R)$.
  return $\boldsymbol{U}_1, \boldsymbol{\Sigma}_F^R, \tilde{\boldsymbol{\Sigma}}_T^R, \sigma_R$

Observation 2 (Equivalence to problem with whitened features) *Let data be generated as in Phase 1. Denote $\tilde{\boldsymbol{\Sigma}}_T = \boldsymbol{\Sigma}_F^{1/2}\boldsymbol{\Sigma}_T\boldsymbol{\Sigma}_F^{1/2}$. Then risk$(\boldsymbol{\Sigma}_F^{-1/2}\boldsymbol{\Lambda}, \boldsymbol{\Sigma}_T, \boldsymbol{\Sigma}_F)$ = risk$(\boldsymbol{\Lambda}, \tilde{\boldsymbol{\Sigma}}_T, \boldsymbol{I})$.*

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 $\boldsymbol{\Lambda}$; however it is not a manageable function of $\boldsymbol{\Lambda}$ 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 COMPUTEREDUCTION subroutine of Algorithm 1 calculates key quantities that are used in specifying the reduction, and the COMPUTEOPTIMALREP 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 $\boldsymbol{\Sigma}_F$ is lower/upper bounded as follows: $\mathbf{0} \prec \Sigma_{\min}\boldsymbol{I} \preceq \boldsymbol{\Sigma}_F \preceq \Sigma_{\max}\boldsymbol{I}$.*

Assumption 2 (Joint diagonalizability) *$\boldsymbol{\Sigma}_F$ and $\boldsymbol{\Sigma}_T$ are diagonal matrices.*[1]

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

Assumption 4 *The joint empirical distribution of the eigenvalues of $\boldsymbol{\Lambda}_R$ and $\tilde{\boldsymbol{\Sigma}}_T^R$ is given by the average of Dirac δ's: $\frac{1}{R}\sum_{i=1}^R \delta_{\boldsymbol{\Lambda}_{R,i}, \sqrt{R}\tilde{\boldsymbol{\Sigma}}_{T,i}^R}$. 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 $\boldsymbol{\Lambda}$. 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 $n_2 = \sum_{i=1}^R \left(1 + (\xi\boldsymbol{\Lambda}_i^2)^{-1}\right)^{-1}$. Define $\boldsymbol{\theta} \in \mathbb{R}^R$ with entries $\boldsymbol{\theta}_i = \frac{\xi\boldsymbol{\Lambda}_i^2}{1+\xi\boldsymbol{\Lambda}_i^2}$ and calculate $\tilde{\boldsymbol{\Sigma}}_T^R, \sigma_R$ using the* COMPUTEREDUCTION *procedure of Algorithm 1. Then, define the analytic risk formula*

$$f(\boldsymbol{\theta}, \tilde{\boldsymbol{\Sigma}}_T^R, n_2) = \frac{1}{n_2 - \|\boldsymbol{\theta}\|_2^2}\left(n_2\sum_{i=1}^R(1-\boldsymbol{\theta}_i)^2\tilde{\boldsymbol{\Sigma}}_{T,i}^R + (\|\boldsymbol{\theta}\|_2^2 + 1)\sigma_R^2\right). \tag{3.3}$$

We have that

$$\lim_{n_2\to\infty} f(\boldsymbol{\theta}, \tilde{\boldsymbol{\Sigma}}_T^R, n_2) = \lim_{n_2\to\infty} \textit{risk}(\boldsymbol{\Sigma}_F^{-1/2}\boldsymbol{\Lambda}, \boldsymbol{\Sigma}_T, \boldsymbol{\Sigma}_F) \tag{3.4}$$

The proof of Theorem 1 applies the convex Gaussian Min-max Theorem (CGMT) in [35] and can be found in the Appendix B.2.We show that as dimension grows, the distribution of the estimator $\hat{\beta}$ converges to a Gaussian distribution and we can calculate the expectation of risk.

[1]This is equivalent to the more general scenario where $\boldsymbol{\Sigma}_F$ and $\boldsymbol{\Sigma}_T$ 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 [2].

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; f(\boldsymbol{\theta}, \boldsymbol{\Sigma}_T, \boldsymbol{\Sigma}_F), \text{ s.t. } 0 \leq \boldsymbol{\theta} < 1, \sum_{i=1}^{R} \boldsymbol{\theta}_i = n_2 \tag{OPT-REP}$$

The optimal representation is[3] $\boldsymbol{\Lambda}^*_{R,i} = ((1/\boldsymbol{\theta}^*_i - 1)\xi)^{-2}$. The subroutine COMPUTEOPTIMALREP in Algorithm 1 summarizes this procedure.

Remark 1 *Thm. 1 states that risk*$(\boldsymbol{\Sigma}_F^{-1/2}\boldsymbol{\Lambda}, \boldsymbol{\Sigma}_T, \boldsymbol{\Sigma}_F)$ *can be arbitrarily well-approximated by* $f(\boldsymbol{\theta}, \tilde{\boldsymbol{\Sigma}}_T^R, n_2)$ *if* n_2 *is sufficiently large. In Fig. 1(b), we set* $\boldsymbol{\Sigma}_F = \boldsymbol{I}_{100}$, $\boldsymbol{\Sigma}_T = diag(\boldsymbol{I}_{20}, 0.1\boldsymbol{I}_{80})$, $n_2 = 40$. *The curves in Fig1(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* $\boldsymbol{\Lambda}$ *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* $n_2 = 40$.

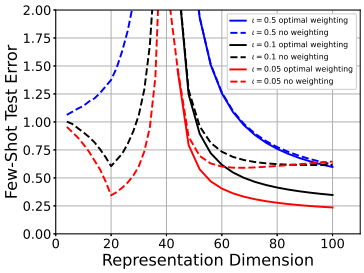

Figure 3: Theoretical risk of optimal representation. $\boldsymbol{\Sigma}_F = \boldsymbol{I}_{100}$, $\boldsymbol{\Sigma}_T = \text{diag}(\boldsymbol{I}_{20}, \iota\boldsymbol{I}_{80})$, $n_2 = 40$.

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 ι controls how spiked $\boldsymbol{\Sigma}_T$ is, with a smaller value for ι indicating increased spikedness. 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 ($\boldsymbol{\Lambda} = \boldsymbol{I}$) 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 risk 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 ι. It implies that, while $\tilde{\boldsymbol{\Sigma}}_T$ 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 $\tilde{\boldsymbol{\Sigma}}_T \in \mathbb{R}^{d\times d}$ 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 $\boldsymbol{\Lambda}^*$. Note that $\boldsymbol{\Lambda}^*$ depends on the canonical covariance $\tilde{\boldsymbol{\Sigma}}_T = \boldsymbol{\Sigma}_F^{1/2}\boldsymbol{\Sigma}_T\boldsymbol{\Sigma}_F^{1/2}$. Learning the R-dimensional principal subspace of $\tilde{\boldsymbol{\Sigma}}_T$ enables us[4] to calculate $\boldsymbol{\Lambda}^*$. Denote this subspace by $\tilde{\boldsymbol{S}}_T$.

[2] In Sec. 5 the constraint is $\underline{\theta} \leq \boldsymbol{\theta} \leq 1 - \frac{d-n_2}{n_2}\underline{\theta}$ for robustness concerns.

[3] In the algorithm, $\xi = 1$ and $\boldsymbol{\Lambda}_{R,i} = (1/\boldsymbol{\theta}^*_i - 1)^{-2}$, because $c\boldsymbol{\Lambda}^*$ for any constant c gives the same $\hat{\boldsymbol{\beta}}$.

[4] We also need to estimate $\boldsymbol{\Sigma}_F$ for whitening. Estimating $\boldsymbol{\Sigma}_F$ is rather easy and incurs smaller error compared to $\tilde{\boldsymbol{\Sigma}}_T$. The analysis is provided in the first part of Appendix B.

Subspace estimation vs. inductive bias. The subspace-based representation $\tilde{\boldsymbol{S}}_T$ has degrees of freedom$= Rd$. When $\tilde{\boldsymbol{\Sigma}}_T$ is exactly rank R and features are whitened, [36] provides a sample-complexity lower bound of $\Omega(Rd)$ examples and gives an algorithm achieving $\mathcal{O}(R^2 d)$ 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, 36] in two aspects:

1. Rather than focusing on a *low dimensional* subspace and assuming $N \gtrsim Rd$, we can estimate $\tilde{\boldsymbol{\Sigma}}_T$ or $\tilde{\boldsymbol{S}}_T$ in the overparameterized regime $N \lesssim Rd$.
2. Rather than assuming whitened features $\boldsymbol{\Sigma}_F = \boldsymbol{I}$ and achieving a sample complexity of $R^2 d$, our learning guarantee holds for arbitrary covariance matrices $\boldsymbol{\Sigma}_F, \boldsymbol{\Sigma}_T$. 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 $\tilde{\boldsymbol{\Sigma}}_T$ or its subspace $\tilde{\boldsymbol{S}}_T$, we investigate the method-of-moments (MoM) estimator.
Definition 4 (MoM Estimator) *For $1 \le i \le T$, define $\hat{\boldsymbol{b}}_{i,1} = 2n_1^{-1} \sum_{j=1}^{n_1/2} y_{ij} \boldsymbol{x}_{ij}$, $\hat{\boldsymbol{b}}_{i,2} = 2n_1^{-1} \sum_{j=n_1/2+1}^{n_1} y_{ij} \boldsymbol{x}_{ij}$. Set*

$$\hat{\boldsymbol{M}} = n_1^{-1} \sum_{i=1}^{T} (\boldsymbol{b}_{i,1} \boldsymbol{b}_{i,2}^\top + \boldsymbol{b}_{i,2} \boldsymbol{b}_{i,1}^\top),$$

The expectation of $\hat{\boldsymbol{M}}$ is equal to $\boldsymbol{M} = \boldsymbol{\Sigma}_F \boldsymbol{\Sigma}_T \boldsymbol{\Sigma}_F$.

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

We verify the inductive bias on practical image dataset, reported in Appendix A. We assessed correlation coefficient between covariances $\tilde{\boldsymbol{\Sigma}}_T, \boldsymbol{\Sigma}_F$ via the canonical-feature alignment score defined as the correlation coefficient

$$\rho(\boldsymbol{\Sigma}_F, \tilde{\boldsymbol{\Sigma}}_T) := \frac{\langle \boldsymbol{\Sigma}_F, \tilde{\boldsymbol{\Sigma}}_T \rangle}{\|\boldsymbol{\Sigma}_F\|_F \|\tilde{\boldsymbol{\Sigma}}_T\|_F} = \frac{\text{trace}(\boldsymbol{M})}{\|\boldsymbol{\Sigma}_F\|_F \|\tilde{\boldsymbol{\Sigma}}_T\|_F}.$$

Observe that, the MoM estimator $\boldsymbol{M}$ naturally shows up in the alignment definition because the inner product of $\tilde{\boldsymbol{\Sigma}}_T, \boldsymbol{\Sigma}_F$ is equal to trace$(\boldsymbol{M})$. This further supports our inductive bias intuition. As reference, we compared it to canonical-identity alignment defined as $\frac{\text{trace}(\tilde{\boldsymbol{\Sigma}}_T)}{\sqrt{d}\|\tilde{\boldsymbol{\Sigma}}_T\|_F}$ (replacing $\boldsymbol{\Sigma}_F$ with $\boldsymbol{I}$). The canonical-feature alignment score is higher than the canonical-identity alignment score. This significant score difference exemplifies how $\boldsymbol{\Sigma}_F$ and $\tilde{\boldsymbol{\Sigma}}_T$ 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 $r_F = \mathbf{tr}(\boldsymbol{\Sigma}_F)$, $r_T = \mathbf{tr}(\boldsymbol{\Sigma}_T)$, and $\tilde{r}_T = \mathbf{tr}(\tilde{\boldsymbol{\Sigma}}_T)$. Define the approximate low-rankness measure of feature covariance by[5]

$$s_F = \min\ s'_F,\ \text{s.t. } s'_F \in \{1, ..., d\},\ s'_F/d \ge \lambda_{s'_F+1}(\boldsymbol{\Sigma}_F)$$

We have two results for this estimator.

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

Paper [36] has sample complexity $\mathcal{O}(dr^2)$ (r is exact rank). Our sample complexity is $\mathcal{O}(r_F \tilde{r}_T^2)$. $r_F, \tilde{r}_T$ can be seen as effective ranks and our bounds are always smaller than [36]. We will discuss later in Example 1. Our second result says when $n_1 \ge s_T$, our sample complexity achieves the $\mathcal{O}(dr)$ which is proven a lower bound in [36].

[5]The $(s_F + 1)$-th eigenvalue is smaller than s_F/d. Note the top eigenvalue is 1.

feature cov	$\boldsymbol{\Sigma}_F = \boldsymbol{I}, \boldsymbol{\Sigma}_T = \text{diag}(\boldsymbol{I}_{s_T}, \boldsymbol{0})$			$\boldsymbol{\Sigma}_F = \text{diag}(\boldsymbol{I}_{s_F}, \iota_F \boldsymbol{I}_{d-s_F})$, $\boldsymbol{\Sigma}_T = \text{diag}(\boldsymbol{I}_{s_T}, \iota_T \boldsymbol{I}_{d-s_T})$		
estimator	sample N	sample n_1	error	sample N	sample n_1	error
MoM	ds_T^2	1	$(ds_T^2/N)^{1/2}$	$r_F r_T^2$	1	$(r_F r_T^2/N)^{1/2}$
MoM	ds_T	s_T	$(s_T/n_1)^{1/2}$	$r_F r_T$	r_T	$(r_T/n_1)^{1/2}$

Table 2: **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. $r_F = s_F + \iota_F(d - s_F)$, $r_T = s_T + \iota_T(d - s_T)$ 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 $\mathcal{O}(ds_T)$ (cf. [36, 22]).

Theorem 2 *Let data be generated as in Phase 1. Assume* $\|\boldsymbol{\Sigma}_F\|, \|\boldsymbol{\Sigma}_T\| = 1$ *for normalization*[6].

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

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

2. Assume $T \geq s_F$. *If* $n_1 \gtrsim \tilde{r}_T + \sigma^2$, *then with probability at least* $1 - CT^{-100}$

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

Denote the top-R *principal subspaces of* $\boldsymbol{M}, \hat{\boldsymbol{M}}$ *by* $\boldsymbol{M}_{top}, \hat{\boldsymbol{M}}_{top}$ *and assume the eigen-gap condition* $\lambda_R(\boldsymbol{M}) - \lambda_{R+1}(\boldsymbol{M}) > 2\|\hat{\boldsymbol{M}} - \boldsymbol{M}\|$. *Then a direct application of Davis-Kahan Theorem [13] bounds the subspace angle as follows*

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

Estimating eigenspace of canonical covariance. Note that if $\boldsymbol{\Sigma}_F$ and $\boldsymbol{\Sigma}_T$ are aligned, (e.g. Example 1 below with $s_F = s_T = R$), then $\boldsymbol{M}_{\text{top}} = \tilde{\boldsymbol{S}}_T$ is exactly the principal subspace of $\tilde{\boldsymbol{\Sigma}}_T$. Theorem 2 indeed gives estimation error for the principal subspace of $\tilde{\boldsymbol{\Sigma}}_T$. Note that, such alignment is and more general requirement compared to related works which require whitened features [36, 22].

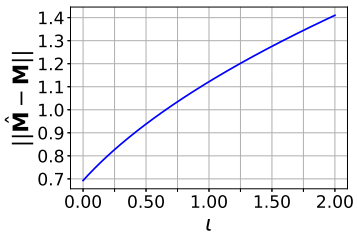

Figure 4: Error of MoM estimator

Example 1 (Spiked $\tilde{\boldsymbol{\Sigma}}_T$, Aligned principal subspaces) *Suppose the spectra of* $\boldsymbol{\Sigma}_F$ *and* $\tilde{\boldsymbol{\Sigma}}_T$ *are bimodal as follows* $\boldsymbol{\Sigma}_F = diag(\boldsymbol{I}_{s_F}, \iota_F \boldsymbol{I}_{d-s_F})$, $\boldsymbol{\Sigma}_T = diag(\boldsymbol{I}_{s_T}, \iota_T \boldsymbol{I}_{d-s_T})$. *Set statistical error* $Err_{T,N} := \sqrt{r_T^2 r_F/N} + \sqrt{r_T/T}$. *When* $\iota_T, \iota_F < 1$, $s_F \geq s_T$, *the recovery error of* $\tilde{\boldsymbol{\Sigma}}_T$ *and its principal subspace* $\tilde{\boldsymbol{S}}_T$ *are bounded as*

$$angle(\hat{\boldsymbol{M}}_{top}, \tilde{\boldsymbol{S}}_T) \lesssim Err_{T,N} + \iota_F^2 \iota_T \quad and \quad \|\hat{\boldsymbol{M}} - \tilde{\boldsymbol{\Sigma}}_T\| \lesssim Err_{T,N} + \iota_F \iota_T.$$

The estimation errors for $\tilde{\boldsymbol{\Sigma}}_T, \tilde{\boldsymbol{S}}_T$ are controlled in terms of the effective ranks and the spectrum tails ι_F, ι_T. Typically $s_F s_T \gtrsim n_1$ so $\sqrt{r_T^2 r_F/N}$ term dominates the statistical error in practice. In Fig. 4 we plot the error of estimating $\boldsymbol{M}$ (whose principal subspace coincides with $\tilde{\boldsymbol{\Sigma}}_T$). $\boldsymbol{\Sigma}_F = \text{diag}(\boldsymbol{I}_{30}, \iota \boldsymbol{I}_{70})$, $\boldsymbol{\Sigma}_T = \text{diag}(\boldsymbol{I}_{30}, \boldsymbol{0}_{70})$. $T = N = 100$. We can see that the error increase with ι .

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 $\boldsymbol{\Lambda}$ using the *estimated* canonical covariance, rather than the full-information setting of Section 3.

[6]This is simply equivalent to scaling y_{ij}, which does not affect the normalized error $\|\hat{\boldsymbol{M}} - \boldsymbol{M}\|/\|\boldsymbol{M}\|$. In the appendix we define $\mathcal{S} = \max\{\|\boldsymbol{\Sigma}_F\|, \|\boldsymbol{\Sigma}_T\|\}$ and prove the theorem for general $\mathcal{S}$.

Armed with the provably reliable estimators of Section 4, we can replace $\tilde{\mathbf{\Sigma}}_T$ and $\mathbf{\Sigma}_F$ 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[7] $\text{risk}(\mathbf{\Lambda}, \mathbf{\Sigma}_T, \mathbf{\Sigma}_F) - \text{risk}(\mathbf{\Lambda}^*, \mathbf{\Sigma}_T, \mathbf{\Sigma}_F)$.

Let us replace the constraint in (OPT-REP) by $\underline{\theta} \leq \boldsymbol{\theta} \leq 1 - \frac{d-n_2}{n_2}\underline{\theta}$. This changes the "optimization" step in Algorithm 1. Theorem 3 does not require an explicit computation of the optimal representation by enforcing $\underline{\theta}$. 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 $\underline{\theta}$ after-the-fact as the minimum singular value of this representation to apply Theorem 3 without assuming an explicit $\underline{\theta}$.

Let $\mathbf{\Lambda}_{\underline{\theta}}(R) = \textsc{ComputeOptimalRep}(R, \mathbf{\Sigma}_F, \hat{\boldsymbol{M}}, \sigma, n_2)$ denote the estimated optimal representation and $\mathbf{\Lambda}^*_{\underline{\theta}}(R) = \textsc{ComputeOptimalRep}(R, \mathbf{\Sigma}_F, \tilde{\mathbf{\Sigma}}_T, \sigma, n_2)$ 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 $\mathbf{\Lambda}_{\underline{\theta}}(R)$, $\mathbf{\Lambda}^*_{\underline{\theta}}(R)$ be as defined above, and $r_F = \mathbf{tr}(\mathbf{\Sigma}_F)$, $r_T = \mathbf{tr}(\mathbf{\Sigma}_T), \tilde{r}_T = \mathbf{tr}(\tilde{\mathbf{\Sigma}}_T)$. The risk of meta-learning algorithm satisfies*[8]

$$\textit{risk}(\mathbf{\Lambda}_{\underline{\theta}}(R), \mathbf{\Sigma}_T, \mathbf{\Sigma}_F) - \textit{risk}(\mathbf{\Lambda}^*_{\underline{\theta}}(R), \mathbf{\Sigma}_T, \mathbf{\Sigma}_F) \lesssim \frac{n_2^2}{d(R-n_2)(2n_2 - R\underline{\theta})\underline{\theta}} \left[(\tilde{r}_T + \sigma^2)\sqrt{\frac{r_F}{N}} + \sqrt{\frac{r_T}{T}} \right].$$

Notice that as the number of previous tasks T and total representation-learning samples N observed increases, the risk of the estimated $\mathbf{\Lambda}_{\underline{\theta}}(R)$ approaches that of the optimal $\mathbf{\Lambda}^*_{\underline{\theta}}(R)$ 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 [38]. We plot it in Fig 1(b) on the left side of the peak as a comparison.

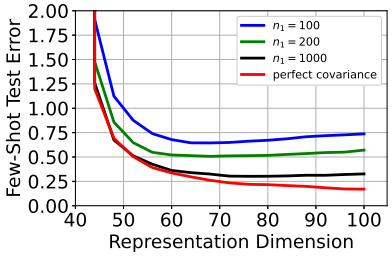

Figure 5: End to end learning guarantees. $d = 100, n_2 = 40, T = 200$, $\mathbf{\Sigma}_T = (\boldsymbol{I}_{20}, 0.05 \cdot \boldsymbol{I}_{80})$, $\mathbf{\Sigma}_F = \boldsymbol{I}_{100}$.

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 $\hat{\boldsymbol{M}}$, use it to compute $\mathbf{\Lambda}$ 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 $\mathbf{\Lambda}$ in a larger space. The optimal $\mathbf{\Lambda}$ in a feasible *sub*set 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.

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.

[7] Note that Sec.6 of [38] gives the exact value of $\text{risk}(\mathbf{\Lambda}^*, \mathbf{\Sigma}_T, \mathbf{\Sigma}_F)$ so we have an end to end error guarantee.

[8] 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

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

[2] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory*, pages 191–210, 2015.

[3] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 2020.

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

[5] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[6] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58, 2014.

[7] Quentin Bouniot, Ievgen Redko, Romaric Audigier, Angélique Loesch, Yevhenii Zotkin, and Amaury Habrard. Towards better understanding meta-learning methods through multi-task representation learning theory. *arXiv preprint arXiv:2010.01992*, 2020.

[8] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[9] Giovanni Cavallanti, Nicolo Cesa-Bianchi, and Claudio Gentile. Linear algorithms for online multitask classification. *The Journal of Machine Learning Research*, 11:2901–2934, 2010.

[10] Xiangyu Chang, Yingcong Li, Samet Oymak, and Christos Thrampoulidis. Provable benefits of overparameterization in model compression: From double descent to pruning neural networks. *arXiv preprint arXiv:2012.08749*, 2020.

[11] Sitan Chen, Jerry Li, and Zhao Song. Learning mixtures of linear regressions in subexponential time via fourier moments. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 587–600, 2020.

[12] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

[13] Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[15] Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

[16] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018.

[17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[18] Halil Ibrahim Gulluk, Yue Sun, Samet Oymak, and Maryam Fazel. Sample efficient subspace-based representations for nonlinear meta-learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3685–3689. IEEE, 2021.

[19] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

[20] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[21] Weihao Kong, Raghav Somani, Sham Kakade, and Sewoong Oh. Robust meta-learning for mixed linear regression with small batches. *arXiv preprint arXiv:2006.09702*, 2020.

[22] Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. *arXiv preprint arXiv:2002.08936*, 2020.

[23] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32:8572–8583, 2019.

[24] Yuanzhi Li and Yingyu Liang. Learning mixtures of linear regressions with nearly optimal complexity. In *Conference On Learning Theory*, pages 1125–1144, 2018.

[25] Karim Lounici, Massimiliano Pontil, Sara Van De Geer, Alexandre B Tsybakov, et al. Oracle inequalities and optimal inference under group sparsity. *The annals of statistics*, 39(4):2164–2204, 2011.

[26] James Lucas, Mengye Ren, Irene Kameni, Toniann Pitassi, and Richard Zemel. Theoretical bounds on estimation error for meta-learning. *arXiv preprint arXiv:2010.07140*, 2020.

[27] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

[28] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

[29] Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*, 2019.

[30] Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime, 2020.

[31] Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *arXiv preprint arXiv:2005.08054*, 2020.

[32] Vidya Muthukumar, Kailas Vodrahalli, and Anant Sahai. Harmless interpolation of noisy data in regression. *CoRR*, abs/1903.09139, 2019.

[33] Preetum Nakkiran, Prayaag Venkat, Sham Kakade, and Tengyu Ma. Optimal regularization can mitigate double descent. *arXiv preprint arXiv:2003.01897*, 2020.

[34] Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *ICML Workshop on Understanding and Improving Generalization in Deep Learning*, 2019.

[35] Christos Thrampoulidis, Ehsan Abbasi, and Babak Hassibi. Lasso with non-linear measurements is equivalent to one with linear measurements. *Advances in Neural Information Processing Systems*, 28:3420–3428, 2015.

[36] Nilesh Tripuraneni, Chi Jin, and Michael I Jordan. Provable meta-learning of linear representations. *arXiv preprint arXiv:2002.11684*, 2020.

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[38] Denny Wu and Ji Xu. On the optimal weighted ℓ_2 regularization in overparameterized linear regression, 2020.

[39] Jiaqi Yang, Wei Hu, Jason D Lee, and Simon S Du. Provable benefits of representation learning in linear bandits. *arXiv preprint arXiv:2010.06531*, 2020.

[40] Jiaqi Yang, Wei Hu, Jason D Lee, and Simon S Du. Impact of representation learning in linear bandits. In *International Conference on Learning Representations*, 2021.

[41] Kai Zhong, Prateek Jain, and Inderjit S Dhillon. Mixed linear regression with multiple components. In *Advances in neural information processing systems*, pages 2190–2198, 2016.

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]