SAMPLE EFFICIENT SUBSPACE-BASED REPRESENTATIONS FOR NONLINEAR META-LEARNING

Halil Ibrahim Gulluk[⋆α] *Yue Sun*[†α] *Samet Oymak*[‡] *Maryam Fazel*[†]

[⋆] Bogazici University
[†] University of Washington
[‡] University of California, Riverside

ABSTRACT

Constructing good representations is critical for learning complex tasks in a sample efficient manner. In the context of meta-learning, representations can be constructed from common patterns of previously seen tasks so that a future task can be learned quickly. While recent works show the benefit of subspace-based representations, such results are limited to linear-regression tasks. This work explores a more general class of nonlinear tasks with applications ranging from binary classification, generalized linear models and neural nets. We prove that subspace-based representations can be learned in a sample-efficient manner and provably benefit future tasks in terms of sample complexity. Numerical results verify the theoretical predictions in classification and neural-network regression tasks.

***Index Terms*—** representation learning, binary classification, generalized linear models, nonlinear problems

1. INTRODUCTION

Meta-learning (and multi-task learning) has proved to be a powerful technique when available training data is limited. The central idea is exploiting the information (e.g. training data) provided by earlier related tasks to quickly adapt a new task using few samples. This idea has a rich history [1, 2] and has shown promise in modern machine learning tasks, e.g., in image classification [3], machine translation [4] and reinforcement learning [5], all of which may involve numerous tasks to be learned with limited data per task.

Modern deep learning algorithms typically exploit the shared information between tasks by learning useful representations [6, 7]. The multi-task system was studied by [1], and the idea of meta-learning or transfer learning is investigated empirically in modern machine learning framework, showing that the shared representation benefits for training on the new tasks [8, 9, 10, 11]. An instructive and well-studied problem for meta-learning is mixed linear regression, for which efficient algorithms and sample complexity bounds are discussed in [12, 13, 14]. If the tasks lie on a shared low-dimensional subspace, learning this subspace would serve as an efficient representation which helps reduce the search space for future tasks. Once the search space is low dimensional, in order to get the same accuracy, the amount of data required for training is reduced compared to training over the full parameter space. [15, 16, 17] propose sample complexity bounds for representation learning for linear multi-task systems. There are study of mixed linear tasks combined with other structures, such as boolean combination of features [18], half-spaces [19] and sparse representations [20].

The recent papers [21, 22] propose meta-learning procedures that involve dimension reduction, clustering and few-shot learning. Here a low-dimensional task subspace is used as the search space for few-shot learning for the new task. Another related approach [23, 24] sets up a nonconvex optimization problem with matrix factors of appropriate sizes, which captures the low dimensional structure. One can apply gradient descent to this nonconvex problem, and studying its behavior requires a nontrivial landscape analysis of the matrix factorization problem.

However, existing provable algorithms for representation learning are restricted to linear-regression tasks, whereas typical machine learning tasks involve nonlinearity. This can arise from the use of nonlinear models as well as nonlinear label link function (e.g. generalized linear models). A good example is classification problems which represent much of the machine learning applications including computer vision and natural language processing [3, 4]. In classification tasks, the model is a map from images to labels, and the labels are discrete and not linear with respect to the input images (i.e. logistic link function). Another example is the use of nonlinear models such as deep networks, which contain many nonlinear activation functions within their layers. The existing results for representation learning for the linear-regression setting cannot be easily extended to the nonlinear case.

Can we learn efficient subspace representations for nonlinear tasks such as generalized linear models and neural nets?

We consider a realizable setup where the input data is high-dimensional, the *relevant features* lie in a low dimen-

[α] Equal contribution.

sional subspace and the labels depend only on the *relevant features*. These assumptions are the same as in the existing literature, however we additionally allow for the scenario where labels are possibly an arbitrary nonlinear function of the relevant features. We make the following contributions.

• **Efficient representations for nonlinear tasks:** We show that subspace found via method-of-moments (MOM) leads to a consistent estimate of the ground-truth subspace despite arbitrary task nonlinearities, when the data is normally distributed. We combine this with non-asymptotic learning results to establish sample complexity bounds for representation learning.

• **Few-shot learning and Applications:** We specialize our results to practical settings with tasks involving binary classification and neural nets. We theoretically and empirically show that subspace-based representation can greatly improve sample efficiency of future tasks.

2. PROBLEM FORMULATION

The meta-learning setup that will be considered in this work consists of two phases: (i) meta-training: prior tasks are used to learn a good representation and (ii) few-shot learning: the new task is learned with few samples. In the meta-training phase, we learn the low dimensional space spanned by parameters. In the few-shot learning phase, we use the subspace to learn the model of a new task ideally with few samples.

In the first phase, there are multiple tasks to infer from, each with its own distribution. We consider a realizable model where the input and label is associated via a labeling function. One accesses batches of data, each of whom is collected from a task, however we may not know which task it comes from. We make this setup more precise using the following definitions. Below, the ground-truth representation will be denoted by a matrix $\boldsymbol{W} \in \mathbb{R}^{r\times d}$ row space of which corresponds to the subspace of interest.

Definition 2.1. ***Meta-training data.*** *Fix a matrix $\boldsymbol{W} \in \mathbb{R}^{r\times d}$ satisfying $\boldsymbol{W}\boldsymbol{W}^T = \boldsymbol{I}$. The j-th task is associated with function $f^j : \mathbb{R}^r \to \mathbb{R}$. Given input $\mathbf{x}$, the label y is distributed as $p_j(y|\mathbf{x}) = p_j(y|\boldsymbol{W}\mathbf{x})$[1] and the expectation satisfies $\boldsymbol{E}(y) = f^j(\boldsymbol{W}\mathbf{x})$. Suppose there are n_j samples from the j-th task sampled i.i.d. from this distribution and we denote the dataset $\mathcal{S}^j = (\mathbf{x}_{i,j}, y_{i,j})_{i=1}^{n_j}$. Define the full meta-training dataset to be $\mathcal{S} = \bigcup_{j=1}^{k} \mathcal{S}^j$.*

Here, f^j is allowed to be any Lipschitz non-linear function, i.e., a neural network[2].

Definition 2.2. ***Binary classification.*** *Suppose f^j takes values over $[0, 1]$,*

$$y_{i,j} = \begin{cases} 1, & \text{with probability } f^j(\boldsymbol{W}\mathbf{x}_{i,j}), \\ 0, & \text{with probability } 1 - f^j(\boldsymbol{W}\mathbf{x}_{i,j}). \end{cases}$$

Definition 2.3. ***Generalized linear models (GLM)*** *(which include logistic/linear regression) can be modeled by choosing f^j to be parameterized by a vector $\boldsymbol{\theta}_j \in \mathbb{R}^r$ and a link function $\phi : \mathbb{R} \to \mathbb{R}$ as $f^j(\boldsymbol{W}\mathbf{x}_{i,j}) := \phi(\boldsymbol{\theta}_j^T \boldsymbol{W}\mathbf{x}_{i,j})$.*

When the dimension of the span of parameters is small, [21] performs a dimension reduction algorithm to find the low-dimensional subspace that the parameters span. This is done by selecting the top eigenvectors of the covariance estimate of the cross-correlation between input and labels.

Definition 2.4. ***Moment estimator of covariance.*** *We define the covariance estimator as*

$$\hat{\boldsymbol{M}} = \sum_{j=1}^{k} \frac{2}{n_j^2} \Bigg[\Big(\sum_{i=1}^{n_j/2} y_{i,j}\mathbf{x}_{i,j}\Big)\Big(\sum_{i=n_j/2+1}^{n_j} y_{i,j}\mathbf{x}_{i,j}\Big)^\top + \Big(\sum_{i=n_j/2+1}^{n_j} y_{i,j}\mathbf{x}_{i,j}\Big)\Big(\sum_{i=1}^{n_j/2} y_{i,j}\mathbf{x}_{i,j}\Big)^\top \Bigg]. \tag{2.1a}$$

Subspace estimation. To estimate the subspace $\boldsymbol{W}$, we use rank-r approximation of $\hat{\boldsymbol{M}}$ to retrieve its principal eigenvector subspace. Let $\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^\top$ be the eigen-decomposition of $\hat{\boldsymbol{M}}$. Denote $\hat{\lambda}_j$ as the jth eigenvalue of $\hat{\boldsymbol{\Lambda}}$. Let $\hat{\boldsymbol{U}}_r$ be the first r columns of $\hat{\boldsymbol{U}}$, thus the rank-r approximation is $\hat{\boldsymbol{M}}_r = \hat{\boldsymbol{U}}_r \mathrm{diag}(\hat{\lambda}_1, ..., \hat{\lambda}_r)\hat{\boldsymbol{U}}_r^\top$. In the next section, we will prove that the range of $\hat{\boldsymbol{U}}$ is close to the row space of $\boldsymbol{W}$.

In Algorithm 1, the output $\hat{\boldsymbol{U}}_r$ is the estimator of the task subspace $\boldsymbol{W}$. $\hat{\boldsymbol{U}}_r$ is used as a training step for the few-shot learning phase. For the new task, we search for the function f^* that minimizes the population loss. We shall provide an instructive analysis for the cross-entropy loss, which is usually employed for classification tasks.

Definition 2.5. ***Few-shot learning (Population).*** *In the few-shot learning phase, suppose $\mathbf{x}, y \sim \mathscr{P}_{x,y}$ satisfy $\boldsymbol{E}(y|\mathbf{x}) = f^*(\boldsymbol{W}\mathbf{x})$. Let $\mathcal{F}$ be a family of functions as the search space for few-shot learning model. Let $\mathcal{L} : \mathcal{F} \times \mathbb{R}^{r\times d} \to \mathbb{R}$ be population cross-entropy loss, defined as*

$$\mathcal{L}(f; \boldsymbol{P}) = -\boldsymbol{E}_{\mathscr{P}_{x,y}}\big(y \log f(\boldsymbol{P}\mathbf{x}) + (1-y)\log(1 - f(\boldsymbol{P}\mathbf{x}))\big). \tag{2.2}$$

We search for the solution induced by $\hat{\boldsymbol{U}}_r$ by

$$\hat{f} = \operatorname*{argmin}_{f\in\mathcal{F}} \mathcal{L}(f; \hat{\boldsymbol{U}}_r^\top) \tag{2.3}$$

Observe that, without representation learning, one has to search for both f and $\boldsymbol{P}$. However with representation learning, we fix $\boldsymbol{P} = \hat{\boldsymbol{U}}_r$ and only search for f.

[1] In words, the label only depends on the relevant features induced by $\boldsymbol{W}$.

[2] In our theoretical results, we treat f as a general linear function, and in experiments we will use a neural network with a specific structure.

Algorithm 1 Meta-training and Few-shot Learning

Require: Dataset $\mathcal{S}$, representation size r, function space $\mathcal{F}$
Compute $\hat{\boldsymbol{M}}$ via method-of-moments (2.1).
Rank r approximation:
$\hat{\boldsymbol{M}}_r \leftarrow \hat{\boldsymbol{U}}_r \mathrm{diag}(\hat{\boldsymbol{\Lambda}}_{1,1}, ..., \hat{\boldsymbol{\Lambda}}_{r,r}) \hat{\boldsymbol{U}}_r^\top$.
Either $\hat{f} \leftarrow \mathrm{argmin}_{f\in\mathcal{F}} \mathcal{L}(f; \boldsymbol{U}_r^\top)$. $\mathcal{L}$ is defined as (2.2).
Or $\hat{f}_e \leftarrow \mathrm{argmin}_{f\in\mathcal{F}} \mathcal{L}_e(f; \boldsymbol{U}_r^\top)$. $\mathcal{L}_e$ is defined as (2.6)
return $\hat{\boldsymbol{U}}_r$ and $\hat{f}$ or $\hat{f}_e$.

Remark 2.1. *For the GLM Definition 2.3, we can choose $\mathcal{F}$ to be the ℓ_2 norm constrained functions for some $a \le \infty$*

$$\mathcal{F} = \{\mathbf{x} \to \phi(\boldsymbol{\theta}^T \mathbf{x}) \;\big|\; \|\boldsymbol{\theta}\|_2 \le a,\ \boldsymbol{\theta} \in \mathbb{R}^r\}, \tag{2.4}$$

Let the new data be generated with $f^*(\boldsymbol{W}\mathbf{x}) = \phi(\boldsymbol{\theta}^{*\top}\boldsymbol{W}\mathbf{x})$ for some ground-truth parameter $\boldsymbol{\theta}^*$. We use $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{P})$ to denote the cross-entropy loss in this setting. $\hat{f}$ (parameterized by $\hat{\boldsymbol{\theta}}$) is given by

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}}\ \mathcal{L}(\boldsymbol{\theta}; \hat{\boldsymbol{U}}_r^\top),\ \textit{such that}\ \|\boldsymbol{\theta}\| \le a. \tag{2.5}$$

Definition 2.6. ***Few-shot learning (Finite sample GLM).*** *Suppose there are in total n samples for new task $(\mathbf{x}_i, y_i)_{i=1}^n$ and $(\mathbf{x}_i, y_i)$ satisfies $\boldsymbol{E}(y_i|\mathbf{x}_i) = f^*(\boldsymbol{W}\mathbf{x}_i)$. Lef $\mathcal{F}$ be a family of functions for few-shot learning phase. We consider the setup of Remark 2.1, where ϕ is the logistic function.*

$$\mathcal{L}_e(\boldsymbol{\theta}; \boldsymbol{P}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(\phi(\boldsymbol{\theta}^\top \boldsymbol{P}\mathbf{x}_i)) + (1-y_i)\log(1-\phi(\boldsymbol{\theta}^\top \boldsymbol{P}\mathbf{x}_i))). \tag{2.6}$$

Given an ℓ_2-norm constraint $a \le \infty$, the empirical risk minimizer (ERM) is defined as follows

$$\hat{\boldsymbol{\theta}}_e = \underset{\boldsymbol{\theta}}{\mathrm{argmin}}\ \mathcal{L}_e(\boldsymbol{\theta}; \boldsymbol{U}_r^\top) \quad \textit{such that} \quad \|\boldsymbol{\theta}\|_2 \le a. \tag{2.7}$$

3. MAIN RESULTS

In this section, we shall establish error bounds for Algorithm 1. This involves three parts. Theorem 3.2 establishes the quality of the moment estimator $\hat{\boldsymbol{M}}$. Theorem 3.3 upper bounds the population cross-entropy risk of $\hat{f}$ in the few-shot learning stage. Theorem 3.4 upper bounds the population risk of the ERM estimator $\hat{f}_e$, which is learned from finite data. We define

$$\boldsymbol{h}^j(\boldsymbol{W}) : \mathbb{R}^{r\times d} \to \mathbb{R}^d = \boldsymbol{E}_{\mathbf{x}}[f^j(\boldsymbol{W}\mathbf{x})\mathbf{x}]$$
$$\boldsymbol{M} := \boldsymbol{W}^\top \boldsymbol{W}\left(\frac{1}{k}\sum_{j=1}^{k} \boldsymbol{h}^j(\boldsymbol{W})(\boldsymbol{h}^j(\boldsymbol{W}))^\top\right)\boldsymbol{W}^\top\boldsymbol{W}.$$

In Algorithm 1, $\hat{\boldsymbol{M}}$ is a finite sample estimate of $\boldsymbol{M}$.

Lemma 3.1. *$\boldsymbol{M}, \hat{\boldsymbol{M}}$ satisfies the following. (a) rank$(\boldsymbol{M}) \le r$. (b) range-space$(\boldsymbol{M}) \subset$ row-space$(\boldsymbol{W})$. (c) $\mathbb{E}[\hat{\boldsymbol{M}}] = \boldsymbol{M}$.*

In words, $\boldsymbol{M}$ returns a consistent estimate of the representation space in the sense that its range is guaranteed to be the subspace of the representation. Observe that to fully recover representation, $\boldsymbol{M}$ should contain a diverse set of tasks that can *cover* the representation subspace. For GLM, one needs at least $k \ge r$ tasks to ensure range of $\boldsymbol{M}$ is equal to the row-space of $\boldsymbol{W}$. Additionally, $\hat{\boldsymbol{M}}$ estimator is also consistent. All missing proofs can be found in [25].

We next present the error on the estimator $\hat{\boldsymbol{M}}$. This theorem applies to standard normal data $\mathbf{x} \sim \mathcal{N}(0, \boldsymbol{I})$. While this may initially seem restrictive, we remark that identity covariance is mostly used for notational convenience. Additionally, in similar spirit to Central Limit Theorem, machine learning and signal processing algorithms often exhibit distributional universality: For instance, subgaussian distributions often behave very similar or even identical to gaussian distributions in sufficiently high-dimensions [26, 27]. We leave such generalizations to more general distributions as a future work.

Theorem 3.2. *Suppose the data is generated as Def. 2.1, $n_j \ge N$ for all j and $\mathbf{x}_{i,j} \sim \mathcal{N}(0, \boldsymbol{I})$. Suppose $y\mathbf{x}$ is a subGaussian random vector with covariance upper bounded by*

$$\|\mathbf{Cov}(y\mathbf{x})\| \le \sigma^2. \tag{3.1}$$

(These conditions hold when $f^j(x) < \sigma$.) Let $\delta \in (0,1)$, $\epsilon \in (0,1)$, σ be defined in (3.1). Then there exists a constant $c > 0$ such that with probability at least $1-\delta$, if

$$k \ge \frac{cd}{N}\log^2\left(\frac{kd}{\delta}\right)\max\left\{\frac{1}{\epsilon^2}, \frac{1}{\epsilon}\log\left(\frac{kd}{\delta}\right)\right\},$$

then $\|\hat{\boldsymbol{M}} - \boldsymbol{M}\| \le \epsilon\sigma^2$.

Recall that $\hat{\boldsymbol{M}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^\top$ and $\hat{\boldsymbol{U}}_r$ is the first r columns of $\hat{\boldsymbol{U}}$. Denote the estimate of $\boldsymbol{W}$ via $\hat{\boldsymbol{W}}$ given by adjusting $\boldsymbol{U}_r$

$$\hat{\boldsymbol{W}} = (\hat{\boldsymbol{U}}_r\hat{\boldsymbol{Q}})^\top,\ \hat{\boldsymbol{Q}} = \underset{\boldsymbol{Q}\in\mathbb{R}^{r\times r}, \boldsymbol{Q}\boldsymbol{Q}^\top = I}{\mathrm{argmin}} \|\hat{\boldsymbol{U}}_r\boldsymbol{Q} - \boldsymbol{W}^\top\| \tag{3.2}$$

With the definition of $\hat{\boldsymbol{W}}$, $\|\hat{\boldsymbol{W}} - \boldsymbol{W}\|$ defines a distance between the row space of $\boldsymbol{W}$ and the column space of $\hat{\boldsymbol{U}}_r$. If the span of the two subspaces are the same, then there exists an orthonormal matrix $\boldsymbol{Q}$ such that $\hat{\boldsymbol{U}}_r\boldsymbol{Q} = \boldsymbol{W}^\top$.

In the next step, we will use $\hat{\boldsymbol{U}}_r$ for few shot learning and find $\hat{f}$ that minimize the population loss. Recall that the search space for $\hat{f}$ is $\mathcal{F}$. For binary classification, we assume

1. **Rotation invariance:** For any function $f \in \mathcal{F}$, any orthonormal matrix $\boldsymbol{Q} \in \mathbb{R}^{r\times r}$ and any matrix $\boldsymbol{P} \in \mathbb{R}^{r\times d}$, there exists $g \in \mathcal{F}$ such that $f(\boldsymbol{P}\mathbf{x}) = g(\boldsymbol{Q}\boldsymbol{P}\mathbf{x})$.
2. **Lipschitz:** $\mathcal{F} \subseteq \{f \mid \log f,\ \log(1-f)$ are L Lipschitz$\} \cap \{f \mid 0 < f(x) < 1, \forall x \in \mathbb{R}^r\}$.

Theorem 3.3. *Let $\mathcal{F}$ satisfy the assumptions above. Let $\hat{\boldsymbol{W}}$ be same as in (3.2) and $\hat{f}$ be same as in (2.3). Then we have*

$$\mathcal{L}(\hat{f}; \hat{\boldsymbol{U}}_r^\top) - \mathcal{L}(f^*; \boldsymbol{W}) \lesssim L\sqrt{r}\|\hat{\boldsymbol{W}} - \boldsymbol{W}\|.$$

$\mathcal{L}(f^*; \boldsymbol{W})$ assumes the knowledge of the true function f^* and the representation $\boldsymbol{W}$. This shows that the inaccuracy of the moment estimator $\hat{M}$ costs us $O(L\sqrt{r}\|\hat{\boldsymbol{W}} - \boldsymbol{W}\|)$.

Theorem 3.3 bounds the population risk of $\hat{f}$, when we use $\hat{\boldsymbol{U}}_r^\top$ as the representation subspace. Next we discuss the population risk of the finite sample solution $\hat{f}_e$, which should be worse than $\hat{f}$ due to the limited samples deviating from true data distribution of the new task. Theorem 3.4 bounds the risk in terms of the sample size n.

Theorem 3.4. *Consider the setup in Def. 2.6 with n i.i.d. examples with ground-truth model $\boldsymbol{\theta}^*$. Solve for $\hat{\boldsymbol{\theta}}_e$ via (2.7). There exist constants $c > 1$, $\delta \in (0,1)$, with probability at least $1 - n^{-c+1} - \delta$, the solution pair $(\hat{\boldsymbol{\theta}}_e, \hat{\boldsymbol{U}}_r)$ satisfies*

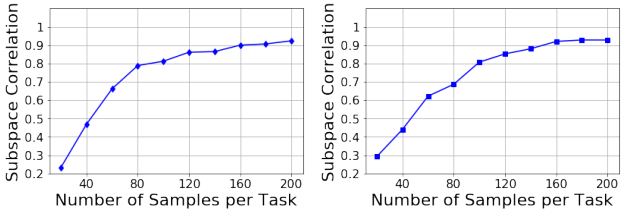
$$\mathcal{L}(\hat{\boldsymbol{\theta}}_e; \hat{\boldsymbol{U}}_r^\top) - \mathcal{L}(\boldsymbol{\theta}^*; \boldsymbol{W}) \leq \frac{a\sqrt{r}\log(n)(c_1 + \sqrt{\log(1/\delta)})}{\sqrt{n}} + L\sqrt{r}\|\hat{\boldsymbol{W}} - \boldsymbol{W}\|.$$

Note that the first term grows as $\sqrt{r/n}$. This means that the amount of data n we request for few-shot learning is $n \approx r$. If we do not run representation learning before few-shot learning, the error would instead be proportional to $\sqrt{d/n}$ where d is the ambient dimension and we would need $n \approx d$ examples to learn the new task.

4. EXPERIMENT

We generate synthetic datasets with k different tasks and n samples for all tasks. As dimension of the data and dimension of the subspace we choose $d = 50$ and $r = 5$, respectively.

We study two different setups. In the first one data is generated according to Def. 2.2 with f^j's being softmax function for all j's. For the second setup, there is an underlying 3-layer neural network that fits the data perfectly i.e. $y = f^j(\boldsymbol{W}\mathbf{x})$. In both setups our aim is to retrieve subspace representations of the data using Algorithm 1 and learning the new task in few-shots. For neural network experiments, we assume that the



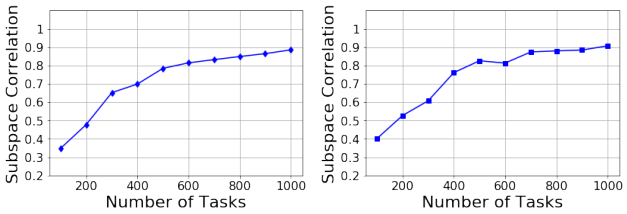
(a) (b)

Fig. 1: Subspace correlations with fixed number of tasks and varying numbers of samples per task. (a) Binary classification, (b) Neural network.

data are generated from a ground truth neural network which has 3 layers, defined as

$$y_{i,j} = f^j(\mathbf{x}_{i,j}) + \epsilon_{i,j} = \boldsymbol{W}_{j3}(\boldsymbol{W}_{j2}(\boldsymbol{W}_{j1}(\boldsymbol{U}_r^\top \mathbf{x}_{i,j}))_+)_+ + \epsilon_{i,j}$$

where $\epsilon_{i,j} \sim \mathcal{N}(0,1)$ is gaussian noise, $(\cdot)_+$ is the ReLU activation function, $\boldsymbol{U}_r \in \mathbb{R}^{50\times 5}$ is representation matrix which is same for all j's. The weight matrices $\boldsymbol{W}_{j1}, \boldsymbol{W}_{j2}$ and $\boldsymbol{W}_{j3}$ are different for each task and they are random gaussian matrices in $\mathbb{R}^{20\times 5}, \mathbb{R}^{20\times 20}, \mathbb{R}^{20}$ respectively.



(a) (b)

Fig. 2: Subspace correlations with fixed number of samples per task and varying number of tasks. (a) Binary classification, (b) Neural network.

In Fig. 1 and Fig. 2 we use the subspace correlation as the metric for evaluating the accuracy of subspace recovery, which is defined by $\frac{\|\hat{\boldsymbol{U}}_r{}^\top \boldsymbol{U}_r\|^2}{\|\boldsymbol{U}_r\|^2}$. In Fig. 1 $k = 100$ is fixed but n's vary from 20 to 200. In Fig. 2 $n = 30$ for all tasks while k changes from 100 to 1000. It can be seen from Fig. 1 and Fig. 2 that as nk gets bigger, the subspace correlation becomes closer to 1, which is compatible with Theorem 3.2

In Fig. 3, the downstream task accuracies for binary classification are depicted. For the new task, a new 1-layer neural network without any activation function is trained with and without the retrieved representations of the earlier tasks. We find the parameters of the neural network by minimizing the cross entropy loss via SGD. For this setup, during meta-training, we set $n = 50$ for all tasks and $k = 2000$. The number of training samples for the new task is varied from 20 to 400. After training the new task, we evaluate the test error with 1000 new samples and feed them into the trained neural network for classification. We check the classification accuracy of the neural network, and plot it for varying number of training data. It can be seen that if the number of few-shot training

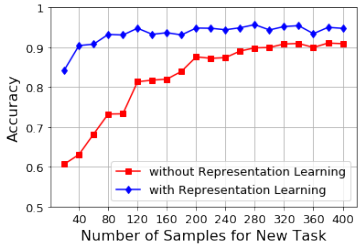


Fig. 3: Accuracy for downstream task

samples is small, accuracy improves much faster when we use representation learning. This validates that dimension reduction reduces the degrees-of-freedom during few-shot learning, thus the optimal model can be learned with fewer samples. As the sample size grows, the relative benefit of representation is smaller but still noticeable.

5. REFERENCES

[1] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[2] Jonathan Baxter, "A model of inductive bias learning," *Journal of artificial intelligence research*, vol. 12, pp. 149–198, 2000.

[3] 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*. Ieee, 2009, pp. 248–255.

[4] 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*.

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

[6] Jürgen Schmidhuber, *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*, Ph.D. thesis, Technische Universität München, 1987.

[7] Sebastian Thrun and Lorien Pratt, *Learning to learn*, Springer Science & Business Media, 2012.

[8] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[9] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.

[10] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[11] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of maml," *arXiv preprint arXiv:1909.09157*, 2019.

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

[13] Yuanzhi Li and Yingyu Liang, "Learning mixtures of linear regressions with nearly optimal complexity," in *Conference On Learning Theory*, 2018, pp. 1125–1144.

[14] 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*.

[15] 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*, vol. 39, no. 4, pp. 2164–2204, 2011.

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

[17] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes, "The benefit of multitask representation learning," *The Journal of Machine Learning Research*, vol. 17, no. 1.

[18] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala, "Efficient representations for lifelong learning and autoencoding," in *Conference on Learning Theory*.

[19] Irina Rish, Genady Grabarnik, Guillermo Cecchi, Francisco Pereira, and Geoffrey J Gordon, "Closed-form supervised dimensionality reduction with generalized linear models," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 832–839.

[20] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil, "Convex multi-task feature learning," *Machine learning*, vol. 73, no. 3, pp. 243–272, 2008.

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

[22] 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.

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

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

[25] Ibrahim Gulluk, Yue Sun, Samet Oymak, and Maryam Fazel, "Sample efficient subspace-based representations for nonlinear meta-learning," https://github.com/sunyue93/RLnonlinear/blob/main/RLnonlinear.pdf, 2000.

[26] Samet Oymak and Joel A Tropp, "Universality laws for randomized dimension reduction, with applications," *Information and Inference: A Journal of the IMA*, vol. 7, no. 3, pp. 337–446, 2018.

[27] Ehsan Abbasi, Fariborz Salehi, and Babak Hassibi, "Universality in learning from linear measurements," in *Advances in Neural Information Processing Systems*, 2019, pp. 12372–12382.

[28] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan, "A short note on concentration inequalities for random vectors with subgaussian norm," *arXiv preprint arXiv:1902.03736*, 2019.

[29] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of machine learning*, MIT press, 2018.

[30] Beatrice Laurent and Pascal Massart, "Adaptive estimation of a quadratic functional by model selection," *Annals of Statistics*, pp. 1302–1338, 2000.

A. PROOF OF MAIN THEOREMS

Theorem A.1. *Suppose the data are generated as Def. 2.1. Let $\delta \in (0,1)$, $\epsilon \in (0,1)$, σ be defined in (3.1). Define*

$$\boldsymbol{h}^j(\boldsymbol{W}) : \mathbb{R}^{r\times d} \to \mathbb{R}^d = \boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)} f^j(\boldsymbol{W}\mathbf{x})\mathbf{x}$$

And let

$$\boldsymbol{M} = \boldsymbol{W}^\top \boldsymbol{W}\left(\frac{1}{k}\sum_{j=1}^{k} \boldsymbol{h}^j(\boldsymbol{W})(\boldsymbol{h}^j(\boldsymbol{W}))^\top\right)\boldsymbol{W}^\top\boldsymbol{W}$$

Then there exists a constant c, with probability at least $1-\delta$, let

$$k = \frac{cd}{n}\log^2(\frac{kd}{\delta})\max\{\frac{1}{\epsilon^2}, \frac{1}{\epsilon}\log(\frac{kd}{\delta})\}$$

we have

$$\|\hat{\boldsymbol{M}} - \boldsymbol{M}\| \le \epsilon\sigma^2.$$

Proof. We first give the lemma for the mean of the random vector $y\mathbf{x}$.

Lemma A.2. *We assume that the data are generated as in Def. 2.1, and we study the j-th task whose activation function is f^j. Define*

$$\boldsymbol{h}^j(\boldsymbol{W}) : \mathbb{R}^{r\times d} \to \mathbb{R}^d = \boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)} f^j(\boldsymbol{W}\mathbf{x})\mathbf{x}$$

Denote the joint distribution of $(\mathbf{x}, y)$ as $\mathscr{P}_{x,y}$. Then $\boldsymbol{E}_{(\mathbf{x},y)\sim\mathscr{P}_{x,y}}(y\mathbf{x}) = \boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{h}^j(\boldsymbol{W})$.

Proof. Denote The expectation can be expanded as

$$\begin{aligned}
&\boldsymbol{E}_{(\mathbf{x},y)\sim\mathscr{P}_{x,y}}(y\mathbf{x})\\
&= \boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)} f^j(\boldsymbol{W}\mathbf{x})\mathbf{x}\\
&= \boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)}\left(f^j(\boldsymbol{W}\mathbf{x})\boldsymbol{W}^\top\boldsymbol{W}\mathbf{x} + f^j(\boldsymbol{W}\mathbf{x})(I-\boldsymbol{W}^\top\boldsymbol{W})\mathbf{x}\right).
\end{aligned}$$

Note that, because $\mathbf{x}\sim\mathcal{N}(0,I)$ so $\boldsymbol{W}\mathbf{x}$ and $(I-\boldsymbol{W}^\top\boldsymbol{W})\mathbf{x}$ are Gaussian. $\boldsymbol{W}\boldsymbol{W}^\top = I$ implies

$$\begin{aligned}
&\boldsymbol{E}(\boldsymbol{W}\mathbf{x})\mathbf{x}^\top((I-\boldsymbol{W}^\top\boldsymbol{W}))\\
&= \boldsymbol{W}(I-\boldsymbol{W}^\top\boldsymbol{W}) = 0.
\end{aligned}$$

So $\boldsymbol{W}\mathbf{x}$ and $(I-\boldsymbol{W}^\top\boldsymbol{W})\mathbf{x}$ are independent, so

$$\begin{aligned}
&\boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)} f^j(\boldsymbol{W}\mathbf{x})(I-\boldsymbol{W}^\top\boldsymbol{W})\mathbf{x}\\
&= (\boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)} f^j(\boldsymbol{W}\mathbf{x}))(\boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)}(I-\boldsymbol{W}^\top\boldsymbol{W})\mathbf{x}) = 0
\end{aligned}$$

Then

$$\begin{aligned}
\boldsymbol{E}_{(\mathbf{x},y)\sim\mathscr{P}_{x,y}}(y\mathbf{x}) &= \boldsymbol{W}^\top\boldsymbol{W}(\boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)} f^j(\boldsymbol{W}\mathbf{x})\mathbf{x})\\
&= \boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{h}_j(\boldsymbol{W}).
\end{aligned}$$

□

The following lemmas are similar to [21] Sec A.1. We extend the concentration inequalities from bounded random vectors to Gaussian random vectors. For completeness we place the lemmas here.

Lemma A.3. *([28] Cor. 7) Let $\delta \in (0,1)$, $t > 0$, with probability $1-\delta$, there exists a constant c such that for every j,*

$$\|\frac{1}{t}\sum_{i=1}^{t} y_{i,j}\mathbf{x}_{i,j} - \boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{h}^j(\boldsymbol{W})\| \le c\sigma\sqrt{\frac{d}{t}\log(\frac{kd}{\delta})} \quad \text{(A.1)}$$

Denote this event as $\mathcal{E}$.

With the covariance of $y\mathbf{x}$ being bounded by σ^2, the following inequalities are true.

$$\begin{aligned}
\boldsymbol{E}\left((\boldsymbol{v}^T\cdot(\frac{1}{t}\sum_{i=1}^{t} y_{i,j}\mathbf{x}_{i,j} - \theta^j))^2\right) &\le \sigma^2/t, \text{ for } \|\boldsymbol{v}\| = 1.\\
\boldsymbol{E}\left(\|\frac{1}{t}\sum_{i=1}^{t} y_{i,j}\mathbf{x}_{i,j} - \theta^j\|^2\right) &\le \sigma^2 d/t.
\end{aligned}$$

Lemma A.4. *Define*

$$\begin{aligned}
\boldsymbol{Z}_j = &(\frac{2}{n_j}\sum_{i=1}^{n_j/2} y_{i,j}\mathbf{x}_{i,j})(\frac{2}{n_j}\sum_{i=n_j/2+1}^{n_j} y_{i,j}\mathbf{x}_{i,j})^\top\\
&- \boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{h}^j(\boldsymbol{W})(\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{h}^j(\boldsymbol{W}))^\top,
\end{aligned}$$

Then there exists a constant c such that on the event $\mathcal{E}$ (thus with probability $1-\delta$), for all $j = 1, ..., k$,

$$\|\boldsymbol{Z}_j\| \le \frac{c\sigma^2 d}{n_j}\log(\frac{kd}{\delta}). \quad \text{(A.2)}$$

The proof is almost same as [21], the only difference is that we replace the bound in ([21] Prop A.1) by (A.1). With the similar replacement, we propose the following lemma.

Lemma A.5. *Let $\delta \in (0,1)$. There exists a constant c, such that for any $\epsilon \in (0,1)$, and*

$$k = \frac{cd}{n_j}\log^2(\frac{kd}{\delta})\max\{\frac{1}{\epsilon^2}, \frac{1}{\epsilon}\log(\frac{kd}{\delta})\}, \quad \text{(A.3)}$$

with probability $1-\delta$,

$$\|\frac{1}{k}\sum_{j=1}^{k}\boldsymbol{Z}_j\| \le \epsilon\sigma^2.$$

This means that when (A.3), with probability $1-\delta$,

$$\left\|\hat{\boldsymbol{M}}-\boldsymbol{W}^\top\boldsymbol{W}\left(\frac{1}{k}\sum_{j=1}^{k}\boldsymbol{h}^j(\boldsymbol{W})(\boldsymbol{h}^j(\boldsymbol{W}))^\top\right)\boldsymbol{W}^\top\boldsymbol{W}\right\|\le\epsilon\sigma^2.$$

□

Now we will prove Theorem 3.3. We first review the notations and assumptions mentioned in the theorem.

We define the SVD of $\hat{\boldsymbol{M}}$ as $\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{V}}^\top$. Let the first r columns of $\hat{\boldsymbol{U}}$ be $\boldsymbol{U}_r$.

Let $\hat{\boldsymbol{W}}$ be defined as

$$\hat{\boldsymbol{W}}=\hat{\boldsymbol{U}}_r\hat{\boldsymbol{Q}}, \tag{A.4}$$

$$\hat{\boldsymbol{Q}}=\operatorname*{argmin}_{\boldsymbol{Q}\in\mathbb{R}^{r\times r},\boldsymbol{Q}\boldsymbol{Q}^\top=I}\|\hat{\boldsymbol{U}}_r\boldsymbol{Q}-\boldsymbol{W}\|. \tag{A.5}$$

The population cross entropy loss is defined as

$$\mathcal{L}(f;\boldsymbol{P}):\mathcal{F}\times\mathbb{R}^{r\times d}\to\mathbb{R}=$$
$$-\boldsymbol{E}_{\mathbf{x},y\sim\mathscr{P}_{x,y}}(y\log f(\hat{\boldsymbol{P}}\mathbf{x})+(1-y)\log(1-f(\hat{\boldsymbol{P}}\mathbf{x}))).$$

We will search for a function $f^*\in\mathcal{F}$ that minimizes the population cross entropy loss. $\mathcal{F}$ is a set of functions satisfying

1. For any function $f\in\mathcal{F}$, any orthonormal matrix $\boldsymbol{Q}\in\mathbb{R}^{r\times r}$ and any matrix $\boldsymbol{P}\in\mathbb{R}^{r\times d}$, there exists $g\in\mathcal{F}$ such that $f(\boldsymbol{P}\mathbf{x})=g(\boldsymbol{Q}\boldsymbol{P}\mathbf{x})$.
2. $\mathcal{F}\subseteq\{f\mid\log f,\ \log(1-f)\text{ are }L\text{ Lipschitz}\}\bigcap\{f\mid 0<f(x)<1,\forall x\in\mathbb{R}^r\}$.

We solve for $\hat{f}$, defined as

$$\hat{f}=\operatorname*{argmin}_{f\in\mathcal{F}}\ \mathcal{L}(f;\hat{\boldsymbol{U}}_r^\top). \tag{A.6}$$

Now we are ready to state the theorem.

Theorem A.6. *Based on the definition above, we have that*

$$\mathcal{L}(\hat{f};\hat{\boldsymbol{U}}_r^\top)-\mathcal{L}(f^*;\boldsymbol{U}_r^\top)\lesssim L\sqrt{r}\|\hat{\boldsymbol{W}}-\boldsymbol{W}\|.$$

Proof. We learn the model from the following optimization algorithm.

$$\begin{aligned}\hat{f}&=\operatorname*{argmin}_{f\in\mathcal{F}}\ \mathcal{L}(f;\hat{\boldsymbol{U}}_r^\top)\\&=\operatorname*{argmin}_{f\in\mathcal{F}}\ -\boldsymbol{E}_{\mathbf{x},y\sim\mathscr{P}_{x,y}}(y\log f(\hat{\boldsymbol{U}}_r^\top\mathbf{x})\\&\quad+(1-y)\log(1-f(\hat{\boldsymbol{U}}_r^\top\mathbf{x})))\\&=\operatorname*{argmin}_{f\in\mathcal{F}}\ -\boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)}\left(f^*(\boldsymbol{W}\mathbf{x})\log f(\hat{\boldsymbol{U}}_r^\top\mathbf{x})\right.\\&\quad\left.+(1-f^*(\boldsymbol{W}\mathbf{x}))\log(1-f(\hat{\boldsymbol{U}}_r^\top\mathbf{x}))\right).\end{aligned}$$

Denote $\tilde{f}\in\mathcal{F}$ as the function such that $\tilde{f}(\hat{\boldsymbol{W}}\mathbf{x})=\hat{f}(\hat{\boldsymbol{U}}_r^\top\mathbf{x})$. So that $\mathcal{L}(\hat{f};\hat{\boldsymbol{U}}_r^\top)=\mathcal{L}(\tilde{f};\hat{\boldsymbol{W}})$. Since $\hat{f}$ minimizes the cross entropy loss, we have that $\mathcal{L}(\tilde{f};\hat{\boldsymbol{W}})\le\mathcal{L}(f^*;\hat{\boldsymbol{W}})$.

Now we have

$$\begin{aligned}&\mathcal{L}(\tilde{f};\hat{\boldsymbol{W}})-\mathcal{L}(f;\boldsymbol{W})\\&\le\mathcal{L}(f^*;\hat{\boldsymbol{W}})-\mathcal{L}(f^*;\boldsymbol{W})\\&\le-\boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)}\left(f^*(\boldsymbol{W}\mathbf{x})(\log f^*(\hat{\boldsymbol{W}}\mathbf{x})-\log f^*(\boldsymbol{W}\mathbf{x}))\right.\\&\quad\left.+(1-f^*(\boldsymbol{W}\mathbf{x}))(\log(1-f^*(\hat{\boldsymbol{W}}\mathbf{x}))-\log(1-f^*(\boldsymbol{W}\mathbf{x})))\right)\\&\le\boldsymbol{E}_{\mathbf{x}\sim\mathcal{N}(0,I)}L\|(\hat{\boldsymbol{W}}-\boldsymbol{W})x\|\\&\lesssim L\sqrt{r}\|\hat{\boldsymbol{W}}-\boldsymbol{W}\|.\end{aligned}$$

□

Now we prove that the upper bound of the error for few-shot learning with finite samples.

Theorem A.7. *Suppose we generate n sample data for few shot learning solve for $\hat{\theta}_e$ from (2.7). There exist constants $c>0$, $\delta\in(0,1)$, with probability at least $1-n^{-c+1}-\delta$, the solution $\hat{\theta}_e$ and $\hat{\boldsymbol{U}}_r$ satisfy*

$$\begin{aligned}&\mathcal{L}(\hat{\theta}_e;\hat{\boldsymbol{U}}_r^\top)-\mathcal{L}(\theta^*;\boldsymbol{W})\\&\lesssim\frac{a\sqrt{r}\log(n)(c_1+\sqrt{\log(1/\delta)})}{\sqrt{n}}+\sqrt{r}\|\hat{\boldsymbol{W}}-\boldsymbol{W}\|.\end{aligned}$$

Proof. We first review that, in few-shot learning, the true parameter is θ^*, and the empirical loss function with finite data is

$$\begin{aligned}&\mathcal{L}_e(\theta;\boldsymbol{P}):\mathcal{F}\times\mathbb{R}^{r\times d}\to\mathbb{R}=\\&-\frac{1}{n}\sum_{i=1}^{n}(y_i\log\phi(\theta^\top\boldsymbol{P}\mathbf{x}_i)+(1-y_i)\log(1-\phi(\theta^\top\boldsymbol{P}\mathbf{x}_i)))\end{aligned} \tag{A.7}$$

Denote each term in the summation as

$$\begin{aligned}&\mathcal{L}_e^i(\theta;\boldsymbol{P}):\mathcal{F}\times\mathbb{R}^{r\times d}\to\mathbb{R}=\\&-(y_i\log\phi(\theta^\top\boldsymbol{P}\mathbf{x}_i)+(1-y_i)\log(1-\phi(\theta^\top\boldsymbol{P}\mathbf{x}_i)))\end{aligned}$$

We search for the solution by

$$\hat{\theta}_e=\operatorname*{argmin}_{\theta}\ \mathcal{L}_e(\theta;\boldsymbol{U}_r^\top),\ \text{such that}\ \|\theta\|\le a. \tag{A.8}$$

In Theorem A.6, we know that

$$\mathcal{L}(\hat{\theta};\hat{\boldsymbol{U}}_r^\top)-\mathcal{L}(\theta^*;\boldsymbol{U}_r^\top)\lesssim L\sqrt{r}\|\hat{\boldsymbol{W}}-\boldsymbol{W}\|.$$

And we will bound the difference between $\mathcal{L}$ and $\mathcal{L}_e$ by Rademacher complexity theory.

Lemma A.8. *[29] Denote*

$$M=\max_{\|\theta\|\le a}|\theta^\top\mathbf{x}_i|\le a\max\|\mathbf{x}_i\|. \tag{A.9}$$

Let $\mathscr{U}$ be the independent random variables uniformly chosen from $\{-1,1\}$. Let $\mathcal{R}$ be the Rademacher complexity of the logistic functionals defined on the data,

$$\mathcal{R}=\frac{1}{n}\boldsymbol{E}_{\epsilon_i\in\mathscr{U}}\sup_{\|\theta\|\le a}\sum_{i=1}^{n}\epsilon_i\mathcal{L}_e^i(\theta;\hat{\boldsymbol{U}}_r^\top)$$

Then for any θ such that $\|\theta\| \le a$, with probability $1-\delta$, we have that

$$|\mathcal{L}_e(\theta; \boldsymbol{U}_r^\top) - \mathcal{L}(\theta; \boldsymbol{U}_r^\top)| \le \mathcal{R} + \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}} M.$$

Thus we have that

$$|\mathcal{L}(\hat{\theta}_e; \hat{\boldsymbol{U}}_r^\top) - \mathcal{L}_e(\hat{\theta}_e; \hat{\boldsymbol{U}}_r^\top)| \le \mathcal{R} + \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}} M,$$
$$|\mathcal{L}(\hat{\theta}; \hat{\boldsymbol{U}}_r^\top) - \mathcal{L}_e(\hat{\theta}; \hat{\boldsymbol{U}}_r^\top)| \le \mathcal{R} + \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}} M.$$

Because $\hat{\theta}_e$ minimizes the empirical loss $\mathcal{L}_e$,

$$\mathcal{L}_e(\hat{\theta}_e; \hat{\boldsymbol{U}}_r^\top) \le \mathcal{L}_e(\hat{\theta}; \hat{\boldsymbol{U}}_r^\top).$$

Thus

$$\mathcal{L}(\hat{\theta}_e; \hat{\boldsymbol{U}}_r^\top) - \mathcal{L}(\hat{\theta}; \hat{\boldsymbol{U}}_r^\top) \le 2\left(\mathcal{R} + \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}} M\right). \quad \text{(A.10)}$$

So the remaining target is to bound $\mathcal{R}$.

We first denote $\boldsymbol{v}_i = \hat{\boldsymbol{U}}_r^\top \mathbf{x}_i$, so $\boldsymbol{v}_i \in \mathbb{R}^r$ and $\boldsymbol{v}_i$ are jointly independent standard normal random variables. In binary classification, $\mathcal{L}_e^i$ is always 1 Lipschitz in $\theta^\top \boldsymbol{v}_i$, the Rademacher complexity can be upper bounded by

$$\mathcal{R} \le \tilde{\mathcal{R}} := \frac{1}{n} \boldsymbol{E}_{\epsilon_i \in \mathscr{U}} \sup_{\|\theta\| \le a} \sum_{i=1}^n \epsilon_i \theta^\top \boldsymbol{v}_i.$$

We first refer to the following lemma.

Lemma A.9. *[30] Let $c > 0$, X follow χ_r^2 distribution, then*

$$\boldsymbol{P}(X - r \ge 2\sqrt{cr \log n} + 2c \log n) \le n^{-c}.$$

For a constant $c > 1$, via union bound we have that $\max \|\boldsymbol{v}_i\|^2 \le r + 2\sqrt{cr \log n} + 2c \log n$ for all $i = 1, .., n$ with probability $1 - n^{-c+1}$. We use $\max \|\boldsymbol{v}_i\| \lesssim c(\sqrt{r} + \log(n))$ for simplicity.

Conditioned on this event, we apply the Rademacher complexity for linear model that

$$\tilde{\mathcal{R}} \le \frac{\max \|\theta\| \cdot \max \|\boldsymbol{v}_i\|}{\sqrt{n}}$$
$$\le \frac{ca(\sqrt{r} + \log(n))}{\sqrt{n}}.$$

Combining with (A.10) and Theorem A.6 we get the bound in Theorem A.7. We can also bound M in (A.9). □