Identification of Negative Transfers in Multitask Learning Using Surrogate Models

Anonymous authors
Paper under double-blind review

Abstract

Multitask learning is widely used to train low-resource tasks by combining multiple related source tasks. Yet, in practice, naively using all the source tasks does not always improve prediction for the target task due to negative transfers. Therefore, identifying beneficial subsets of source tasks for the target task is a critical problem. This problem is challenging because the number of subsets grows exponentially with the number of source tasks; efficient heuristics for subset selection may not capture the complex relationships between subsets and the multitask learning performances. This paper designs a scalable approach to address this problem. First, we sample random subsets of source tasks and compute their performances to fit a surrogate model to predict the performance of unsampled subsets. We show that fitting our model only requires sampling a number of subsets that is linear in the number of source tasks; Moreover, the fitted model provides a relevance score of each source task to the target task. Second, we use these scores to select source tasks to optimize the performance of the target task. Through extensive experiments on various datasets, we demonstrate that our approach can predict negative transfers four times more accurately than existing techniques and can consistently improve upon existing optimization methods for multitask learning.

1 Introduction

Multitask learning (MTL) is an approach to combining several tasks together and learning one model for all tasks simultaneously (Caruana, 1997b). The premise is that by combining the data samples of several tasks together, the dataset size of each task increases, thus improving the learning performance for every task. However, naively using all the source tasks may worsen performance compared to single-task learning (STL) for target tasks if there exist source tasks that are unrelated to them. This problem is commonly referred to as negative transfers in the literature but is challenging to predict for many tasks (Rosenstein et al., 2005).

More broadly, the importance of developing a better understanding of multiple learning performance is well recognized. In the seminal work of Caruana (1997a, Chapter 3), heuristics for judging "task-relatedness" in several applications are discussed; For instance, the tasks that share many input features are more likely to be related to each other. A classical result by Ben-David et al. (2010) introduces an $\mathcal{H}$-divergence notion that quantifies the distance between two label distributions and relates the bias of source domains to this notion. MTL may perform worse than STL if the bias is too large. In weakly-supervised learning, several programmatic labeling functions are used to annotate a corpus of unlabeled data, and each labeling function can be treated as a source task (Ratner et al., 2016). The task labels often conflict with each other, causing negative transfer during training even though the tasks share the same input features (Ratner et al., 2019). In multitask learning of language prediction tasks, negative transfers are observed between different categories of tasks (e.g., question answering vs. sequence labeling) and different sizes of datasets (Vu et al., 2020).

Motivated by the need to reduce negative transfers between different tasks, researchers have developed optimization methods for multitask learning from a variety of fields. The most thorough approach to addressing these issues is to train all possible combinations of source tasks with the target task and find which subset of source tasks improves performance on the main target task. This combinatorial approach is usually impractical. A more efficient solution is to train combinations of every *single* source task with the

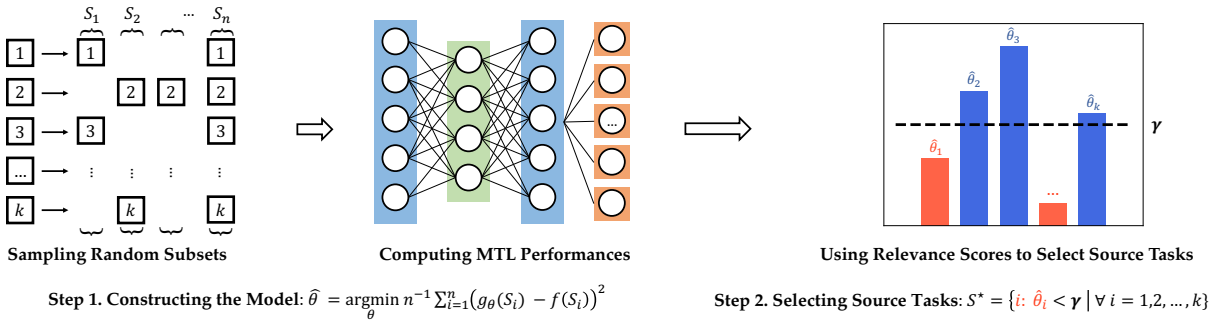


Figure 1: Overview of the pipeline. **Step 1:** Sample n random subsets of source tasks among k source tasks. Let $S_1, S_2, \ldots, S_n$ denote these random samples. For $i = 1, 2, \ldots, n$, train a model using the combined data of S_i and the target task; Evaluate the loss of the trained model on the target task, denoted as $f(S_i)$. Then, estimate the relevance scores θ in the surrogate model by minimizing the MSE between the surrogate model $g_\theta(S_i)$ and $f(S_i)$. **Step 2:** Optimize MTL performance by selecting a subset of source tasks whose relevance scores are below some threshold γ, which is equivalent to minimizing the surrogate model $g_{\hat{\theta}}(S)$ over S.

target task to determine if one task helps and then merge the helpful tasks together. This approach is known as computing first-order task affinities, which captures first-order transfers from one source task to the target task (Fifty et al., 2021). Due to computational costs, approximations of higher-order transfers from multiple source tasks to one target task have been studied, e.g., by averaging the first-order affinity scores of each source task (Standley et al., 2020).

This paper designs and analyzes a scalable approach to identify negative transfers from multiple source tasks to one target task. The key idea is to construct a surrogate model to approximate the performance of multitask learning, given a subset of source tasks. This is different from prior works that measure task-relatedness based on either gradient similarity (Yu et al., 2020), or feature space alignment of neural networks (Nguyen et al., 2020; Raghu et al., 2020; Wu et al., 2020). It is also different from existing discrepancy notions between source and target domains, which are difficult to measure on deep neural networks. Our approach builds on a recent work (Ilyas et al., 2022) that demonstrates the power of datamodels for predicting the predictions of black-box models such as deep neural networks. Different from their work, we apply surrogate models to approximate multitask learning performances of deep neural networks.

Our approach introduces a relevance score of every source task to the target task while accounting for the presence of other source tasks. Given k source tasks and any j from 1 to k, let θ_j denote the relevance score of task j. Conceptually, θ_j is analogous to the important score of a feature in random forests when hundreds of other features are available. To estimate the relevance scores, we introduce a surrogate model $g_\theta(S)$, parametrized by the relevance scores, to approximate MTL performances. Given any subset of source tasks S, let $f(S)$ be a loss function that measures the performance of combining S and the target task. Thus, the loss value of $f(S)$ provides a relevance measure between S and the target task; Since θ_j is used to fit loss values, a lower value of θ_j indicates more usefulness of source task j to the target task.

We specify the surrogate model as $g_\theta(S) = \sum_{j \in S} \theta_j$ and minimize the mean squared error between $g_\theta(S_i)$ and $f(S_i)$ over n random subsets, for i from 1 to n. We use such a linear specification of θ because computing the performance of each subset requires training an MTL model, which is not scalable unless n grows almost linearly in k. We rigorously analyze the sample complexity of our approach in Theorem 2.1. After fitting θ, we predict the performance of an unseen subset S as $g_\theta(S)$ and compare it with the STL performance of the target task to determine if S provides a negative transfer.

Provided with the relevance scores, we optimize the performance of the target task by selecting source tasks whose score is below some threshold γ, which is determined via cross-validation. We observe that this algorithm is equivalent to minimizing the surrogate model $g_\theta(S)$ over subsets S. We analyze this algorithm in a setting that includes one group of source tasks closer to the target task and another group further from the target task. The analysis reveals that for each task i, its relevance score θ_i is proportional to the sum of

the MTL performances of all subsets that include i. Moreover, these performances preserve the distance gaps from the source tasks. See Theorem 3.1 for the precise result.

Taken together, our approach constitutes an efficient, accurate pipeline to predict and optimize multitask learning performances. See Figure 1 for an illustration. We conduct extensive experiments to validate our approach in numerous data modalities and performance metrics. We summarize a list of our results as follows:

- The runtime for constructing surrogate models until convergence scales linearly in k, and the predicted performances accurately fit the true MTL performances of unseen subsets, measured by Spearman's correlation (0.8 averaged among 16 evaluations). Our approach achieves 4 times higher accuracy for predicting positive vs. negative transfers than known approximation schemes, measured by the F_1-score.
- By selecting source tasks based on the predicted MTL performances and only using the selected source tasks, we observe consistent benefits over existing optimization methods. For one example (while deferring others to Section 4), we consider a weak supervision dataset that includes as many as 164 labeling functions (Zhang et al., 2021a). By selecting labeling functions with our approach and then applying MTL, we obtain up to 3.6% absolute accuracy lift compared with existing methods.
- We further visualize the tasks that are selected by our approach and find that there is indeed a separation between the selected tasks in terms of their labeling accuracy. Besides, our MTL approach can also be cast to learning from multiple groups of heterogeneous subpopulations, where the performance is measured by the worst-group accuracy among all groups.

Organization. We begin by describing the problem setup and surrogate models in Section 2. In Section 3, we use the estimated relevance scores to optimize the performance of multitask learning. We present the experiments in Section 4. We discuss related works in Section 5. Lastly, we summarize the paper in Section 6. The appendix provides complete proof of our theoretical results and omitted results from the experiments.

2 Predicting Multitask Learning Performances

This section describes the design and analysis of surrogate models for multitask learning. We begin by defining the problem setup. Then, we describe the construction of surrogate models and the estimation of the relevance scores. Lastly, we analyze the sample complexity of the construction procedure.

2.1 Preliminaries

Problem Setup. Let $t = 0$ denote the main target task of interest. Suppose the task's features and labels are drawn from an unknown distribution, denoted as $\mathcal{D}_t$. Let $\mathcal{X}$ denote the feature space. Let the set of all possible labels be denoted as $\mathcal{Y}$. We are given a dataset, which includes a list of examples drawn independently from $\mathcal{D}_t$. Besides, we are also given k datasets from related source tasks, which are all supported on $\mathcal{X} \times \mathcal{Y}$.

A naive approach to optimize MTL is combining all the datasets and evaluating the trained model on the target task. However, this might result in worse performance than single-task learning. Thus, it is crucial to identify if a source task would help or hurt. The most thorough solution for addressing this question is by enumerating all possible combinations of source tasks, leading to a total of 2^k combinations. For each combination of source tasks, train a multitask model using the selected source tasks and the main task. While this procedure optimizes the performance of MTL, it is too slow for large k.

How can we optimize the performance of MTL efficiently? Relatedly, given a set of source tasks, can we predict their transfer effects upon the target task efficiently? Below, we define two common transfer effects.

Positive vs. Negative Transfer. Consider any multitask learning algorithm, denoted as $\mathcal{A}$, which trains a joint model given any set of tasks. For any subset $S \subseteq \{1, 2, \ldots, k\}$, we say that S provides a negative transfer to t if the performance of $\mathcal{A}(S \cup \{t\})$ is worse than $\mathcal{A}(\{t\})$ (e.g., in terms of higher loss values). Likewise, we say that S provides a positive transfer to t if the performance of $\mathcal{A}(S \cup \{t\})$ is better than $\mathcal{A}(\{t\})$. Our goal is to design a scalable method to predict such positive and negative transfer effects.

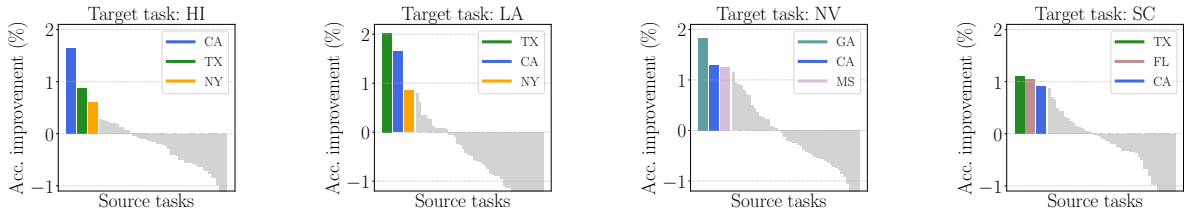


Figure 2: Illustration of mixed outcomes in multitask learning: In each figure, we train 50 multitask models based on one fixed target task and one source task from 50 source tasks. ***x* axis**: Each bar represents one source task from 50 source tasks. ***y* axis**: Test accuracy of MTL with one source task minus the test accuracy of STL with the target task alone. For further description of the experiment setup, see Section 4.1.

It is worth highlighting that both types of transfers are often observed in practice. To give an example, we consider a binary classification dataset that involves a total of 51 tasks. We pick one of them as the target task, use the rest as source tasks, and consider the case where $|S| = 1$. This leads to training 50 models for each target task, one for every combination of one source task and the target task. The results are shown in Figure 2, which provides illustrations for four different target tasks. The y-axis corresponds to the accuracy difference between the MTL and STL results. For all four target tasks, we consistently find a mix of positive and negative transfers.

2.2 Constructing the Model

First, we construct a surrogate model to fit multitask learning performances. Let ϕ be an encoder that is shared by all tasks. For any input features $x \in \mathcal{X}$, the encoder ϕ maps x into a feature vector. For every source task in S and the target task, there is a separate prediction layer for each of them. Let $\psi_0, \psi_1, \psi_2, \ldots, \psi_k$ denote the prediction layers, which map the feature vectors to the output.

Given any subset S, we train ϕ and ψ_i, for $i \in S \cup \{0\}$, by minimizing the average loss over the combined training data along with the target task. Let $\phi^{(S)}$ and $\psi_i^{(S)}$, for $i \in S \cup \{0\}$, be the trained model. We evaluate its loss on the target task's validation dataset. Let $\widetilde{\mathcal{D}}_t = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ denote a set of m independent samples from $\mathcal{D}_t$, which is used as a validation set for the target task. Let ℓ be a non-negative loss such as the cross-entropy loss. We define *multitask learning performances* as:

$$f(S) = \frac{1}{m} \sum_{i=1}^{m} \ell\Big(\psi_0^{(S)}\big(\phi^{(S)}(x_i)\big), y_i\Big), \text{ for any } S \subseteq \{1, 2, \ldots, k\}. \tag{1}$$

Our main idea is to construct a surrogate model, parametrized by a relevance score θ_i for each source task i. We use a linear specification inspired by a recent work of Ilyas et al. (2022)[1]

$$g_\theta(S) = \sum_{i \in S} \theta_i, \text{ where } \theta_i \text{ is the } i\text{-th entry of } \theta, \text{ for any } i = 1, 2, \ldots, k. \tag{2}$$

The procedure for estimating θ is as follows. First, sample n subsets of source tasks from $\{1, 2, \ldots, k\}$, denoted as $S_1, S_2, \ldots, S_n$. We sample each subset from the uniform distribution over subsets with a fixed size of α; We will justify this choice later in Section 3.2. For instance, to capture the transfer from five source tasks to the target task, we can set $\alpha = 5$. Then, compute the value of $f(S_i)$ by training one MTL model for every $i = 1, 2, \ldots, n$. Lastly, minimize the mean squared error (MSE) between $g_\theta(S_i)$ and $f(S_i)$, averaged over all i:

$$\hat{\mathcal{L}}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \Big(g_\theta(S_i) - f(S_i)\Big)^2. \tag{3}$$

Let $\hat{\theta}$ denote the minimizer of the above MSE. For brevity, we refer to $\hat{\theta}$ as the task model.

[1] We use this specification for scalability consideration. Note that it is possible to consider more complex specifications, such as adding quadratic variables $\theta_{1,2}, \theta_{1,3} \ldots, \theta_{k-1,k}$. The construction procedure and the analysis is conceptually the same. However, the sample complexity for fitting these quadratic variables is $\mathrm{O}(k^2)$, rendering it infeasible for large k, e.g., $k \geq 100$.

2.3 Sample Complexity

We show that given $n = \mathrm{O}\big(k \log^2(k)\big)$, we can estimate $\hat{\theta}$ accurately. To be precise, let $\mathcal{U}$ denote the uniform distribution over all subsets of size α drawn from $\{1, 2, \ldots, k\}$. Let T denote an unseen subset drawn from $\mathcal{U}$. The population risk for a given θ is defined as the expected MSE between $f(T)$ and $g_\theta(T)$:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{f} \mathop{\mathbb{E}}_{T} \left[\big(f(T) - g_\theta(T) \big)^2 \right]. \tag{4}$$

Let the minimizer of the above risk be denoted as $\theta^\star$. We prove that $\hat{\theta}$ converges to $\theta^\star$ using Rademacher complexity-based arguments. Let the function class of ψ_t and ϕ be denoted as $\mathcal{H}$. Let the loss function class be $\mathcal{F} = \{\ell(\psi_t(\phi(x)), y) \mid \forall\, \psi_t, \phi \text{ from } \mathcal{H}\}$. Let $\sigma_1, \sigma_2, \ldots, \sigma_m$ be m independent Rademacher random variables, collectively as $\sigma_{1:m}$. The Rademacher complexity of $\mathcal{F}$ over $\widetilde{\mathcal{D}}_t$ is defined as

$$\mathcal{R}_m(\mathcal{F}) = \mathop{\mathbb{E}}_{\widetilde{\mathcal{D}}_t} \mathop{\mathbb{E}}_{\sigma_{1:m}} \left[\sup_{h \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i \cdot h\big(x_i, y_i\big) \right]. \tag{5}$$

We follow the convention of big-O notations for stating the result. Given two functions $h(n)$ and $h'(n)$, we use $h(n) = \mathrm{O}(h'(n))$ or $h(n) \lesssim h'(n)$ to indicate that $h(n) \leq C \cdot h'(n)$ for some fixed constant C when n is large enough. Our result is stated formally below.

Theorem 2.1. *Suppose the functions in $\mathcal{F}$ are all bounded from above by a fixed constant $C > 0$. Suppose α is less than $k/2$. Let n be the number of sampled subsets and m be the size of the set used to evaluate f. With probability at least 0.99, $\hat{\theta}_n$ converges to $\theta^\star$ as n, m are both large enough:*

$$\left\| \hat{\theta} - \theta^\star \right\| \lesssim C \sqrt{\frac{\big(k \log^2\big(k\big)\big)\alpha^4}{n}} + \sqrt{\frac{\big(\log(k)\big)\alpha}{m}} + \mathcal{R}_m(\mathcal{F}), \tag{6}$$

where $\| \cdot \|$ denotes the Euclidean norm of a vector.

Based on the above result, it is clear from equation (6) that provided with $\mathrm{O}(k \log^2(k))$ random samples, the first error term relating to n shrinks to a negligible value (one may think of α as a fixed constant such as 5 or 10). There are two error terms decreasing with m, the size of $\widetilde{\mathcal{D}}_t$ used to evaluate f. The Rademacher complexity $\mathcal{R}_m(\mathcal{F})$ is known to be of order $\mathrm{O}(m^{-1/2})$ when $\mathcal{H}$ represents a family of neural networks (Bartlett et al., 2017). These two error terms are due to the variance of f since it is measured on a finite set.

Proof Overview. We introduce a few notations to examine $g_\theta(S)$ and the covariance of $\mathcal{U}$. Let $\mathcal{I}_n \in \{0, 1\}^{n \times k}$ be a zero-one matrix; For any $i = 1, 2, \ldots, n$, the i-th row is $\mathbb{1}_{S_i}$, the characteristic vector of S_i. Let $\hat{f}$ be a vector in which $\hat{f}_i = f(S_i)$, for any $i = 1, 2, \ldots, n$. The $\hat{\theta}$ that minimizes equation (3) is equal to

$$\hat{\theta} = \big(\mathcal{I}_n^\top \mathcal{I}_n\big)^{-1} \mathcal{I}_n^\top \hat{f}. \tag{7}$$

Let $v = \mathcal{I}_n^\top \hat{f}$ and let v_i be the i-th entry of v, for $i = 1, \ldots, k$. Based on the definition of $\mathcal{I}_n$, we observe that

$$v_i = \sum_{1 \leq j \leq n:\, i \in S_j} f(S_j), \text{ for any } 1 \leq i \leq k. \tag{8}$$

Next, let $\boldsymbol{\mathcal{I}} \in \{0, 1\}^{|\mathcal{U}| \times k}$ be a zero-one matrix, where $|\mathcal{U}|$ is the number of subsets in $\mathcal{U}$. Each row of $\boldsymbol{\mathcal{I}}$ corresponds to the characteristic vector of a subset. Let $\boldsymbol{f}$ be a vector such that each entry of this vector corresponds to the MTL performances (cf. equation (1)) of a subset in distribution $\mathcal{U}$. The population risk minimizer $\theta^\star$ for reducing $\mathcal{L}(\theta)$ in equation (4) is equal to

$$\theta^\star = \big(\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}\big)^{-1} \boldsymbol{\mathcal{I}}^\top \mathbb{E}\,[\boldsymbol{f}].$$

Our proof involves two steps. First, we deal with the error due to the randomness of the random subsets. Let

$$\bar{\theta} = \big(\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}\big)^{-1} \boldsymbol{\mathcal{I}}^\top \boldsymbol{f}.$$

We state the following result, which shows that $\hat{\theta}$ converges to $\bar{\theta}$ as n increases.

Lemma 2.2. *In the setting of Theorem 2.1, conditional on $f(T)$ for any subset $T \in \mathcal{U}$, with probability $1-2\delta$ over the randomness of $S_1, S_2, \ldots, S_n$, for any $\delta \geq 0$, the Euclidean distance between $\hat{\theta}$ and $\bar{\theta}$ satisfies:*

$$\left\|\hat{\theta} - \bar{\theta}\right\| \leq 4C\sqrt{\frac{\left(k\log^2(2k\delta^{-1})\right)\alpha^4}{n}} + 8C\sqrt{\frac{k\alpha^2}{\delta n}}. \tag{9}$$

The proof of the above result relies on a novel union bound taken over all subsets in $\mathcal{S}$. Crucially, there are at most 2^k subsets. By taking the logarithm of 2^k after the union bound, we get a factor of k as shown in equation (9). Second, we prove the convergence from $\bar{\theta}$ to $\theta^\star$, as m increases.

Lemma 2.3. *In the setting of Theorem 2.1, for any $\delta > 0$, with probability at least $1-\delta$ over the randomness of $S_1, S_2, \ldots, S_n$ and $f(S_1), f(S_2), \ldots, f(S_n)$, the Euclidean distance between $\bar{\theta}$ and $\theta^\star$ satisfies:*

$$\left\|\bar{\theta} - \theta^\star\right\| \leq \frac{\mathcal{R}_m(\mathcal{F})}{\sqrt{2}} + \sqrt{\frac{\left(\log\left(\delta^{-1}k\right)\right)\alpha}{4m}}. \tag{10}$$

Combining Lemma 2.2 and Lemma 2.3 together, we have thus proved that equation (6) holds. The proof of the above two results can be found in Appendix B. This result justifies why we use a linear specification, as we can scale up the sample complexity.

3 Optimizing Multitask Learning Performance

Next, we design an algorithm to optimize the performance of the target task using the relevance scores of each source task. We analyze this algorithm in a setting where the tasks are separated into two groups, with one group being more similar (measured by Euclidean distances) to the target task than the other group.

3.1 Using Relevance Scores to Select Source Tasks

Provided with the surrogate model, we can optimize the target task performance by using the predicted MTL performances. A natural way is to select the subset that minimizes the value of $g_{\hat{\theta}}(S)$ over subset S. Due to the linear specification of $g_{\hat{\theta}}$, this is equivalent to selecting source tasks with a small $\hat{\theta}_i$. Thus, we select a source task i if $\hat{\theta}_i$ is below the desired threshold γ, which can be determined via cross-validation. Then, we train a model by combining the selected source tasks with the target task. The complete procedure is shown below. We will rigorously justify the existence of a threshold afterward.

Algorithm 1 Optimizing MTL Performance Using Relevance Scores

Input: k source tasks; Training and validation datasets of the target task.
Require: Size of each subset α; Number of sampled subsets n; MTL algorithm f; Task selection threshold γ.
Output: Trained model $\phi^{(S^\star)}, \psi_t^{(S^\star)}$.

1: For $i = 1, \ldots, n$, sample a random subset S_i from $\{1, 2, \ldots, k\}$ with size α; evaluate $f(S_i)$ following equation (1).
2: Estimate the relevance scores $\hat{\theta}$ following equation (3).
3: Select source tasks based on their relevance scores: $S^\star = \left\{i : \hat{\theta}_i < \gamma \ \mid \forall\, i = 1, 2, \ldots, k\right\}$.
4: Train a model by combining $S^\star$ and t; denote the trained model as $\phi^{(S^\star)}$, and $\psi_i^{(S^\star)}$ for all $i \in S^\star \cup \{t\}$.

3.2 Analysis of the Algorithm

We analyze the above procedure in a synthetic setting where the data is generated from a linear process. We assume that the input features for each task are drawn from an isotropic Gaussian distribution with p dimensions. For each task i from 0 to k, let $\beta^{(i)} \in \mathbb{R}^p$ denote the linear process for task i. Given a data point from task i with feature vector x, its label is generated as $y = x^\top \beta^{(i)} + \epsilon$, where ϵ is a random variable with mean 0 and variance σ^2. Suppose there are two groups of tasks depending on their distances to $\beta^{(t)}$, given by a, b such that $b > a > 0$. For every $i = 1, \ldots, k$, task i is called a *good* task if $\|\beta^{(i)} - \beta^{(t)}\| \leq a$; On the other hand, i is a *bad* task if $\|\beta^{(i)} - \beta^{(t)}\| \geq b$. We show that there exists a threshold that separates good tasks from bad tasks, stated formally as follows.

Theorem 3.1. *In the setting described above, suppose f is bounded from above by a fixed constant $C > 0$. Suppose there are $d \gtrsim k \log k + p + a^4 k^4 (a^2 - b^2)^{-2}$ data points from every source and target task. Suppose $n \gtrsim C^2 k^2 (a^2 - b^2)^{-2}$ and $m \gtrsim p \log p$. With probability at least 0.99, there exists a threshold γ such that*

- *For any $i = 1, 2, \ldots, k$, if task i is a good task, then $\hat{\theta}_i < \gamma$.*
- *Otherwise, if task i is a bad task, then $\hat{\theta}_i > \gamma$.*

The intuition behind the above result is that $\hat{\theta}_i$ averages the MTL performances of all subsets involving i. If i is a good task, then the averaged performances will be lower, leading to a lower relevance score. Moreover, there exists a threshold that separates the relevance scores of good tasks and bad tasks. We give a toy example to illustrate why $\hat{\theta}$ can preserve the Euclidean distance gaps from the β's. Our experiments later also confirm the existence of such a separation (cf. Figure 5).

Example 3.2 (One-dimensional case)**.** *Consider a one-dimensional case where $p = 1$ and every β is a real value. Let $\beta^{(t)} = 0$. Let $0 < \beta^{(i)} < a$ if i is a good task. Let $b < \beta^{(i)}$ if i is a bad task.*

- Our first observation is that $\hat{\theta}_i$ is proportional to v_i (cf. equation (8)), as shown in Lemma 3.3.
- Our second observation is that v_i is proportional to $\beta^{(i)}$. This is because v_i is the sum of $f(S)$ among all $S \in \mathcal{U}$ involving i, and $f(S)$ is the average of $\beta^{(j)}$ among $j \in S$. Thus, v_i is the average of all β's from the n random subsets, while $\beta^{(i)}$ has a larger weight in v_i than the other β's because i is always in S.

Taken together, we conclude that the relevance scores can preserve the relative values of β's in this example.

Proof Overview. We now generalize the intuition from the one-dimensional case, beginning with the first observation. We show that the relevance scores preserves the distance gap of every pair of tasks from the β's.

Lemma 3.3. *In the setting of Theorem 3.1, with probability $1 - \delta$, for any $\delta > 0$, the following holds:*

$$\left| \frac{1}{n}(\hat{\theta}_i - \hat{\theta}_j) - \frac{k}{\alpha n}(v_i - v_j) \right| \lesssim \frac{\log(\delta^{-1} k)}{\sqrt{n}}, \text{ for any } 1 \le i < j \le k. \tag{11}$$

The above result is by analyzing the covariance of $\mathcal{I}_n$, which is proportional to identity plus a constant shift. Let $\mathrm{Id}_{k \times k}$ be a k by k identity matrix and $e \in \mathbb{R}^k$ be a vector whose entries are all equal to one. By the definition of $\mathcal{I}_n$ and Woodbury matrix identity, we have

$$\mathbb{E}\left[\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right] = \frac{\alpha}{k}\mathrm{Id}_{k \times k} + \frac{\alpha(\alpha - 1)}{k(k-1)} e e^\top \quad \Rightarrow \quad \mathbb{E}\left[\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right]^{-1} = \frac{k}{\alpha}\left(\mathrm{Id}_{k \times k} - \frac{\alpha - 1}{k\alpha - 1} e e^\top\right). \tag{12}$$

Crucially, if we multiply v on the right-hand side of equation (12), then we will get $\frac{k}{\alpha}(v - \frac{\alpha-1}{k\alpha-1}(e^\top v)e)$. Recall that e is the all ones vector, which has the same entry in every coordinate after rescaling. Then, recall $\hat{\theta}$ from equation (3). By matrix concentration inequalities, the spectral norm of the deviation from $\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}$ to its expectation satisfies

$$\left\| \frac{\mathcal{I}_n^\top \mathcal{I}_n}{n} - \mathbb{E}\left[\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right] \right\|_2 \lesssim \frac{\alpha \log(k\delta^{-1})}{\sqrt{n}}. \tag{13}$$

See equation (20), Appendix B.1 for the proof. Thus, combining equations (12) and (13), we claim that $\hat{\theta}$ is equal to $\alpha^{-1} k v$ minus a shared term for every task, modulo the deviation error of order $\mathrm{O}(n^{-1/2})$. By subtracting $\hat{\theta}_i - \frac{k}{\alpha} v_i$ and $\hat{\theta}_j - \frac{k}{\alpha} v_j$, we can cancel out the shared term, leading to equation (11).

Next, we formalize the second observation from the one-dimensional case. Based on equation (8), v_i is a sum of $f(S)$ for all subsets S such that $i \in S$. We then show that $f(S)$ is the sum of $\beta^{(j)}$ for all $j \in S$, based on the pooling structure of our MTL model. Thus, the Euclidean distance between $\beta^{(i)}$ and $\beta^{(t)}$ will also reflect in v_i. For complete proof of Theorem 3.1 (and Lemma 3.3), see Appendix B.4. This result substantiates our intuition that $\hat{\theta}_i$ provides the relevance score of each source task i to the target task while accounting for the presence of other source tasks.

4 Experiments

We apply our approach to three settings. The first setting is about applying weak supervision to unlabeled data, and we apply our algorithm to select labeling functions for combining the weak labels of the labeling functions. The second setting involves language prediction tasks from NLP benchmarks. Again, we use our algorithm to select source tasks to improve the performance of target tasks. The third setting involves learning from multiple groups of heterogeneous subpopulations, where the goal is to train a model with robust performance across all groups. We cast this multi-group learning problem into an MTL framework, and apply our algorithm to select a subset of groups to improve the robustness of target tasks. For all these settings, we show that surrogate models can predict negative transfers accurately, and fit MTL performances well; Moreover, our approach provides consistent benefits over various existing optimization methods for multitask learning. *We plan to release the code after the review process, and it is also available for review upon request.*

4.1 Experimental Setup

Datasets. First, we apply our approach to several text classification tasks from a weak supervision dataset (Zhang et al., 2021a). Each dataset uses several labeling functions to create labels for every unlabeled example. The labels generated by different labeling functions may conflict with each other. We view each labeling function as a source task. The goal is to predict an unlabeled set of examples which is viewed as the target task. A validation dataset that includes the correct labels is available for cross-validation. We include the dataset statistics in Table 1.

Second, we consider MTL with natural language processing tasks. We collect twenty-five datasets across a broad range of tasks, spanning sentiment classification, natural language inference, question answering, etc., from GLUE, SuperGLUE, TweetEval, and ANLI. We view one task as the target and use the rest as source tasks. The goal is to select a subset of source tasks for the best multitask learning performance. We provide the statistics of the twenty-five tasks in Table 3, Appendix C.1.

Third, we consider multi-group learning settings in which a dataset involves multiple subpopulation groups. We consider income prediction tasks based on US census data (Ding et al., 2021). The goal is to predict whether an individual's income is above \$50,000 or not, using ten features, including the individual's education level, age, sex, etc. There are 51 states in this dataset; we view each state as one task. When we perform prediction, we use one state as the target task and use the rest of the fifty states as source tasks. We use the racial group of each individual to split a state population into nine subpopulation groups. We evaluate the robustness of a model by the worst-group accuracy. This metric measures the accuracy of the worst-performing group among all groups. We use six states as the target task. See Table 2 for dataset statistics.

Implementation. We use a standard approach for conducting MTL, i.e., hard parameter sharing. For text classification, we use BERT-Base as the encoder. For tabular features, we use a fully-connected layer with a hidden size of 32. The surrogate modeling procedure requires three parameters: the size of a subset, the number of samples, and the loss function. We select the size from a range between $3, 5, 10, 15$. We select the number of samples from a range between $50, 200, 400, 800$, depending on k. We also collect a holdout set of size 100 for constructing the surrogate model. For classification tasks, we set the loss function as the negative classification margin, i.e., the difference between the correct-class probability and the highest incorrect-class probability. After estimating the surrogate model g from equation (3), we use $g(S)$ as the predicted multitask loss for an unseen subset S. We compare $g(S)$ with the STL performance of task t to determine whether the transfer from S to t is positive or negative. We measure the F_1-score for the minority class (between the positive and negative classes) on the holdout set.

4.2 Results for Predicting Negative Transfers

We validate that our fitted models can accurately identify positive vs. negative transfers from source tasks. Then, we show that these models can be constructed efficiently by reporting the runtime.

Results. We test the accuracy of using surrogate models to predict positive vs. negative transfers. We first evaluate the four examples shown in Figure 2. We set the size of α as 5 and n as 400. By using the model

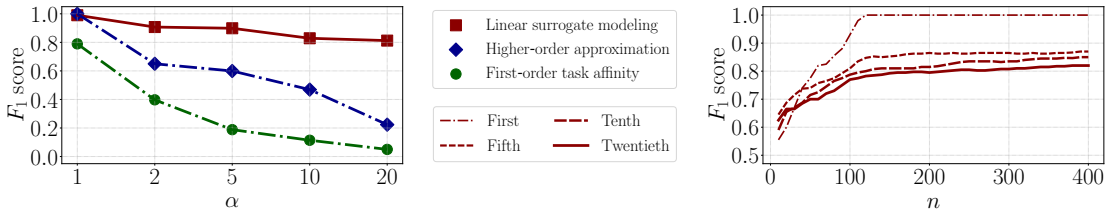


Figure 3: **Left**: Our approach can consistently predict positive/negative transfers from up to 20 source tasks to the target task. **Right**: Convergence of surrogate models as n increases up to 400, leading to an F_1-score of 0.8 for predicting positive/negative transfers from up to 20 source tasks to one target task.

to compare the MTL performances with STL performances, we can correctly predict the transfers with an F_1-score of **0.82**, averaged over the four target tasks. Second, we conduct the same tests for weak supervision and NLP tasks. Similarly, we find that task models can predict positive vs. negative transfers with F_1-score of **0.8** on average for ten different target tasks.

Furthermore, we compare these results with two baselines that either compute first-order task affinity scores or higher-order approximations by averaging first-order affinity scores. Our approach yields much more accurate predictions across different subset sizes of α, ranging from 5 up to 20. Figure 3 provides the illustration for one target task, which is conducted on the folktables dataset, along with fifty source tasks.

Lastly, we measure the Spearman correlation between the predicted performances and true performances. We observe an average coefficient of **0.8** across 16 target tasks. See Appendix A for the details.

Runtime. Next, we report the runtime cost, collected on an NVIDIA Titan RTX card. Our approach requires training n models, one for each random subset. We find that collecting $n \leq 8k$ samples suffice for the fitted model to converge. See Figure 6, Appendix A for the convergence curves of 16 target tasks.

Next, we show that the runtime of our procedure scales linearly with k, the number of source tasks. Let β denote the average size of all the datasets from the source tasks. We plot the number of GPU hours w.r.t. $\alpha\beta k$ in Figure 4. The results confirm our hypothesis. Lastly, we compare the runtime of our approach against the baselines. Compared with computing first-order affinity scores, our approach has the same complexity in k. For example, ours takes 1.24 GPU hours in one target task, while computing first-order affinity scores take 0.87 GPU hours.

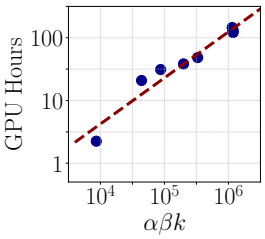


Figure 4: Illustration of linear scaling between k and number of GPU hours.

On the other hand, naively computing f for all possible subsets requires training $\binom{k}{\alpha}$ models. We manage to reduce this down to $O((k \log^2(k))\alpha^4)$ by sampling. Taken together, we conclude that our approach scales efficiently to many tasks.

4.3 Results for Improving MTL Performance

Next, we apply our approach to MTL on weak supervision and NLP tasks. We compare our approach with the following baselines. First, we consider training by naively combining all source and target tasks. Second, we consider MTL optimization methods, including HOA (Standley et al., 2020), TAG (Fifty et al., 2021), and a bilevel optimization method (TAWT, Chen et al. (2022)). To set the threshold γ in our algorithm, we use grid search from -0.5 to 0.5 at an interval of 0.1. We choose this range because it covers the values of most coefficients in our experiments.

Multitask weak supervision. First, we apply our algorithm to five weak supervision datasets, which involve text classification from multiple weak labels. We select a subset of labeling functions so that using their weak labels to train an end model best improves performance on the target task. We also compare against methods that use a label model to aggregate the weak labels and then train an end model on the aggregated label. These include taking a majority vote on the weak labels, applying probabilistic modeling to combine the noisy labels (Ratner et al., 2016), and MeTaL (Ratner et al., 2019).

Table 1: Accuracy/F1-score from surrogate modeling followed by task selection (ours), as compared with MTL methods and weak supervision methods that use a label model to aggregate the weak labels.

Dataset (Metrics)	Youtube (Acc.)	TREC (Acc.)	CDR (F1)	Chemprot (Acc.)	Semeval (Acc.)
Training	1,586	4,965	8,430	12,861	1,749
Validation	120	500	920	1,607	178
Test	250	500	4,673	1,607	600
# source tasks	10	68	33	26	164
Naive MTL	94.72±0.85	64.10±0.50	58.20±0.55	53.43±0.53	89.00±1.06
HOA	94.93±1.80	74.67±4.66	59.76±0.97	45.57±0.41	89.94±4.42
TAG	95.20±0.65	77.50±3.62	59.31±0.15	53.67±2.74	89.06±1.47
TAWT	94.53±1.05	72.40±2.36	59.85±0.30	53.76±2.96	86.83±1.78
Majority voting	95.36±1.71	66.56±2.31	58.89±0.50	57.32±0.98	85.03±0.83
Probabilistic modeling	93.84±1.61	68.64±3.57	58.48±0.73	57.00±1.20	83.93±0.83
MeTaL	92.32±1.44	58.28±1.95	58.48±0.90	56.17±0.66	71.74±0.57
Alg. 1 (Ours)	**97.47±0.82**	**81.80±1.14**	**61.22±0.39**	**57.54±0.55**	**93.50±0.24**

We first examine the labeling functions selected by our approach. Recall that our procedure places a threshold over the coefficients to separate good and bad source tasks. Here, we use the number of correct and incorrect labels as a proxy of relatedness between a labeling function and the target task. Figure 5 shows the results, measured on Chemprot and TREC. We observe a clear separation between selected and excluded tasks. This shows that our algorithm manages to select beneficial source tasks.

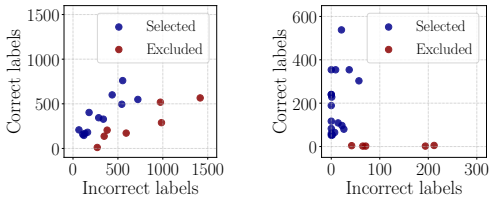


Figure 5: Separation between selected and excluded tasks in terms of labeling accuracy.

Next, we compare the experimental results, shown in Table 1. Compared with naively MTL, which trains all tasks together, our algorithm improves the test performance by **6.4%** on average. Compared with MTL optimization methods and weak supervision methods, our algorithm outperforms their results by up to **3.6%** absolute and **2.3%** on average.

NLP tasks. Next, we test our approach on NLP tasks. We collect 25 datasets from GLUE, SuperGLUE, TweetEval, and ANLI. See Table 3, Appendix C.2 for a complete list. We evaluate our approach by first selecting source tasks and then applying MTL. We test on five target tasks: CoLA, RTE, CB, COPA, and WSC. For each task, we use the rest 24 tasks as source tasks.

We first compare our approach with SLT and naive MTL. We observe that naive MTL can perform worse than STL, e.g., on CoLA and WSC datasets. By contrast, our approach always outperforms STL (by **5.5%**) and naive MTL (by **5.4%**), on average. We then compare our approach with TAG and HOA. Our approach shows an average improvement of **2.2%** and is especially effective for tasks with a small training set.

4.4 Results for Improving Robustness in Multi-group Learning

Lastly, we apply our approach to multi-group learning settings in which the input distribution contains a heterogeneous mixture of subpopulations. The objective of these problems is to learn a model that performs robustly for all groups. In particular, we apply our approach to three performance metrics: worst-group accuracy, democratic disparity, and equality of opportunity. We also compare against STL methods, including group distributional robust optimization (GroupDRO, Sagawa et al. (2020)) and supervised contrastive learning (correct-n-contrast, Zhang et al. (2022)). Table 2 presents the comparison.

Compared with single-task learning, including GroupDRO and correct-n-contrast, we find that task modeling improves the worst-group accuracy by **1.17%** on average. Compared with existing MTL optimization methods, our approach shows a favorable gain of up to **1.9%** absolute accuracy. Measured by two fairness metrics, namely democratic disparity and equality of opportunity, our algorithm outperforms all baseline methods by **1.8%** on average. These results can be found in Table 5 of Appendix C.2.

Table 2: Worst-group accuracies using MTL with source tasks selected by our algorithm, as compared to STL, MTL optimization methods, and exhaustive search over combinations of up to two source tasks.

Dataset	HI	KS	LA	NJ	NV	SC
Training	4,638	9,484	12,400	28,668	8,884	14,927
Validation	1,546	3,161	4,133	9,556	2,961	4,976
Test	1,547	3,162	4,134	9,557	2,962	4,976
Smallest group size	67	75	58	52	61	203
GroupDRO	74.56±0.58	75.50±0.59	74.90±0.38	76.95±0.20	73.06±0.66	75.56±1.36
Correct-n-contrast	74.37±0.27	75.52±1.19	74.25±0.15	77.60±0.10	73.22±0.40	76.23±0.98
Naive MTL	73.63±0.46	75.22±0.73	73.24±1.01	77.28±0.25	73.22±1.12	76.23±0.49
HOA	74.67±0.32	75.22±1.48	73.69±0.86	77.49±0.25	73.88±0.66	76.80±0.65
TAG	74.48±0.41	75.97±1.18	73.24±1.01	77.41±0.48	74.05±0.84	76.41±0.50
TAWT	73.53±0.44	75.14±1.39	73.51±1.38	76.47±1.31	72.89±0.81	76.59±0.97
Exhaustive search ($\alpha \leq 2$)	75.10±0.37	**77.03±0.76**	73.60±1.02	77.40±0.24	73.21±1.10	77.16±0.21
Alg. 1 (Ours)	**75.47±0.73**	76.96±0.69	**75.62±0.11**	**78.17±0.36**	**75.21±0.52**	**77.62±0.34**

4.5 Ablation Studies

Benefit of modeling higher-order transfers. We validate the benefit of modeling higher-order task transfers over approaches that only precompute first-order or second-order task affinities. First, compared with approaches that compute first-order task affinities, our approach improves the accuracy by **3.0%**, as is clear from Tables 1 and 2. Second, we precompute the MTL performance for every combination of two source tasks. We run an exhaustive search over $k(k-1)/2$ combinations to find the best combination for MTL. We test on six target tasks with $k = 50$, which requires training $1,225$ MTL models with two source tasks and one target task each time. We find that our selection procedure consistently outperforms the best two-task subsets by **1.21%** absolute accuracy. This is shown on the last line of Table 2.

Sensitivity of model parameters. There are three parameters that require tuning: the subset size α, the number of samples n, and the loss function ℓ. We set α and n based on the cross-validation results on a holdout set. Besides, we find that choosing ℓ as the classification margin function performs the best in practice. The threshold γ is usually set as 0.3 or 0.4 for weak supervision datasets, which selects most source tasks except the highly noisy labelings. For the NLP and multi-group learning tasks, γ is usually set as -0.5, which selects less than 10 source tasks. Thus, there is only a limited number of beneficial source tasks. Lastly, we find that the selected tasks remain the same when using different random seeds to train the task model. For details, see Appendix C.3.

5 Related Work

We note that there is a vast body of work on multitask learning from a variety of fields. A recurring theme for multitask learning research is inspired by a desire to imitate human intelligence as we continue to learn new information and extrapolate the learned information to new tasks and domains (Thrun et al., 1998). In the early literature, many studies focus on MTL with linear and kernel-based models. A common approach is to set up separate parameters for each task while adding explicit regularization to the combined parameters (Evgeniou et al., 2004; Argyriou et al., 2007; 2008). For linear models particularly, this approach can be related to low-rank matrix approximation (Ando et al., 2005). Inspired by the development of deep learning, recent works focus on MTL with deep neural networks (Yang et al., 2017). More broadly, there are several excellent surveys (Zhang et al., 2021b; Jiang et al., 2022) that provide more comprehensive references. Within this vast literature, the primary contribution of our work is in the identification of negative transfers and the design of principled optimization methods. Below, we discuss the topics most relevant to our study.

Understanding Black-box Predictions. Our approach builds on the recent work of datamodels (Ilyas et al., 2022). However, there are two major differences between our work and their work. First, we apply the idea of surrogate models to multitask learning, whereas their work focuses on the single-task supervised learning setting. Second, besides empirical demonstrations, we have also conducted a theoretical analysis

of our approach for multitask learning. Our findings reinforce the result of Ilyas et al. (2022) that the performances of deep neural networks can be extrapolated efficiently and accurately. Recent work has sought to explain why datamodels can perform well using harmonic analysis (Saunshi et al., 2022). It would be interesting to see if their techniques can be used to explain the empirical findings of our work in the context of MTL. More broadly, there is a line of work on developing techniques to understand the influence of data in black-box models; see, e.g., Koh et al. (2017) and Yeh et al. (2018) for further references.

Formal Notions of Task-relatedness. There is a rich discussion about formulating notions of task-relatedness in the literature (Ben-David et al., 2003). Ben-David et al. (2010) introduces a discrepancy notion called $\mathcal{H}$-divergence, which leads to a generalization bound for minimizing the empirical risk of naive MTL. Transfer exponents are another measure of discrepancy between two distributions (Hanneke et al., 2019). Geometric distance measures for linear data models have also been considered in few-shot learning (Du et al., 2020) and meta-learning (Kong et al., 2020; Saunshi et al., 2021).

Note that none of these task-relatedness measures can be measured on deep neural networks, due to the complexity of these models. One heuristic solution is to measure the cosine similarity between the gradients of each task's loss functions during training (Yu et al., 2020; Dery et al., 2021; Chen et al., 2022). Another solution is to measure the similarity of the predicted probabilities between tasks (Nguyen et al., 2020). This leads to a noisy estimate of task-relatedness, which is best for capturing first-order transfers. Standley et al. (2020) combines domain knowledge from visual intelligence to build a task relation taxonomy for 26 tasks. Compared with their approach, our approach is more generic and is applicable to any MTL settings with little to no domain knowledge, and efficiently captures higher-order transfer in a principled framework. Rather than defining an explicit relatedness measure, our work uses surrogate models as an implicit measure of task-relatedness. This perspective circumvents the design of explicit task-relatedness measures for deep neural networks but is still useful for predicting transfers and for optimizing the performance of MTL.

Optimization Methods for MTL. We draw motivation from recent work which models and integrates weak supervision for rapidly training deep models (Ratner et al., 2016). In particular, we build on previous multitask weak supervision approaches (Ratner et al., 2019), while adding new capability to deal with conflicts between labeling functions in the end model. Our setting is related to the task grouping problem (Kumar et al., 2012), which aims to assign tasks into several groups, with each group of tasks learned in a separate MTL model. Different from this problem, we select source tasks for learning a target task, which has been studied in several recent works (Guo et al., 2019; Chen et al., 2022). There are also works that apply low-rank tensor factorization to the parameters of several neural networks as an extension of the earlier regularization literature (Wimalawarne et al., 2014; Yang et al., 2017). Complementary to these works, we assign a relevance score to each source task and estimate these scores by fitting a surrogate model.

Lastly, we note that task relations are characteristically different between different benchmarks, due to the nature of the data. In this paper, we focus on developing a methodology for predicting MTL performances using rigorous theoretical and empirical arguments. Our extensive experiments demonstrate the usefulness of the methodology. It would be interesting to apply our methodology to large-scale benchmarks beyond what we have studied (Zamir et al., 2018; Aribandi et al., 2022). Besides, it would be interesting to see if our approach can be applied to other related settings such as federated learning (Wang et al., 2020) and multitask reinforcement learning (Wang et al., 2022), where the problem of identifying negative transfers also arises.

6 Conclusion

This paper studied how to efficiently predict negative transfers from multiple source tasks to one target task. The main contribution is the design and analysis of surrogate models for predicting MTL performances. Both theoretical and empirical results are provided to show that our approach is efficient, accurate, and advances over prior MTL techniques. Our work opens up many interesting questions for future work. Although we demonstrated the empirical strength of linear models for MTL, a rigorous explanation is lacking; Can recent analytic tools for understanding datamodels (Saunshi et al., 2022) be used here? Can more advanced sampling techniques, such as adaptive sampling, help speed up the training of surrogate models, which might enable the training of more powerful models? Understanding task relations is a complex and challenging research question in multitask learning. We hope our work inspires more principled studies in this direction.

References

Ando, R. K. and Zhang, T. (2005). "A framework for learning predictive structures from multiple tasks and unlabeled data". In: *Journal of Machine Learning Research*.

Argyriou, A., Evgeniou, T., and Pontil, M. (2008). "Convex multi-task feature learning". In: *Machine learning*.

Argyriou, A., Pontil, M., Ying, Y., and Micchelli, C. (2007). "A spectral regularization framework for multi-task structure learning". In: *Advances in neural information processing systems* 20.

Aribandi, V., Tay, Y., Schuster, T., Rao, J., Zheng, H. S., Mehta, S. V., Zhuang, H., Tran, V. Q., Bahri, D., Ni, J., et al. (2022). "ExT5: Towards Extreme Multi-Task Scaling for Transfer Learning". In: *ICLR*.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. (2017). "Spectrally-normalized margin bounds for neural networks". In: *NeurIPS* 30.

Bartlett, P. L. and Mendelson, S. (2002). "Rademacher and Gaussian complexities: Risk bounds and structural results". In: *Journal of Machine Learning Research* 3.Nov, pp. 463–482.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). "A theory of learning from different domains". In: *Machine learning* 79.1, pp. 151–175.

Ben-David, S. and Schuller, R. (2003). "Exploiting task relatedness for multiple task learning". In: *Learning theory and kernel machines*. Springer, pp. 567–580.

Caruana, R. (1997a). "Multitask Learning". PhD thesis. CMU. unpublished thesis.

– (1997b). "Multitask learning". In: *Machine learning*.

Chen, S., Crammer, K., He, H., Roth, D., and Su, W. J. (2022). "Weighted Training for Cross-Task Learning". In: *ICLR*.

Dery, L. M., Dauphin, Y., and Grangier, D. (2021). "Auxiliary task update decomposition: The good, the bad and the neutral". In: *ICLR*.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. (2021). "Retiring adult: New datasets for fair machine learning". In: *NeurIPS* 34.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. (2020). "Few-shot learning via learning the representation, provably". In: *ICML*.

Evgeniou, T. and Pontil, M. (2004). "Regularized multi–task learning". In: *KDD*.

Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., and Finn, C. (2021). "Efficiently identifying task groupings for multi-task learning". In: *NeurIPS* 34.

Guo, H., Pasunuru, R., and Bansal, M. (2019). "AutoSeM: Automatic Task Selection and Mixing in Multi-Task Learning". In: *NAACL*.

Hanneke, S. and Kpotufe, S. (2019). "On the value of target data in transfer learning". In: *NeurIPS* 32.

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. (2022). "Datamodels: Predicting predictions from training data". In: *ICML*.

Jiang, J., Shu, Y., Wang, J., and Long, M. (2022). "Transferability in Deep Learning: A Survey". In: *arXiv preprint arXiv:2201.05867*.

Koh, P. W. and Liang, P. (2017). "Understanding black-box predictions via influence functions". In: *International conference on machine learning*. PMLR, pp. 1885–1894.

Kong, W., Somani, R., Song, Z., Kakade, S., and Oh, S. (2020). "Meta-learning for mixed linear regression". In: *International Conference on Machine Learning*. PMLR, pp. 5394–5404.

Kumar, A. and Daume III, H. (2012). "Learning task grouping and overlap in multi-task learning". In: *ICML*.

Nguyen, C., Hassner, T., Seeger, M., and Archambeau, C. (2020). "Leep: A new measure to evaluate transferability of learned representations". In: *International Conference on Machine Learning*.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. (2020). "Rapid learning or feature reuse? towards understanding the effectiveness of maml". In: *ICLR*.

Ratner, A., Hancock, B., Dunnmon, J., Sala, F., Pandey, S., and Ré, C. (2019). "Training complex models with multi-task weak supervision". In: *AAAI*.

Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). "Data programming: Creating large training sets, quickly". In: *NeurIPS* 29.

Rosenstein, M. T., Marx, Z., Kaelbling, L. P., and Dietterich, T. G. (2005). "To Transfer or Not To Transfer". In: `http://socrates.acadiau.ca/courses/comp/dsilver/ITWS05_Submitted_Papers/ITWS10-RosensteinM05_ITWS.pdf`.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2020). "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization". In: *ICLR*.

Saunshi, N., Gupta, A., Braverman, M., and Arora, S. (2022). "Understanding Influence Functions and Datamodels via Harmonic Analysis". In: *arXiv preprint arXiv:2210.01072*.

Saunshi, N., Gupta, A., and Hu, W. (2021). "A Representation Learning Perspective on the Importance of Train-Validation Splitting in Meta-Learning". In: *International Conference on Machine Learning*.

Standley, T., Zamir, A., Chen, D., Guibas, L., Malik, J., and Savarese, S. (2020). "Which tasks should be learned together in multi-task learning?" In: *ICML*. PMLR, pp. 9120–9132.

Thrun, S. and Pratt, L. (1998). "Learning to learn: Introduction and overview". In: *Learning to learn*. Springer, pp. 3–17.

Vershynin, R. (2011). "Spectral norm of products of random and deterministic matrices". In: *Probability theory and related fields* 150.3, pp. 471–509.

Vu, T., Wang, T., Munkhdalai, T., Sordoni, A., Trischler, A., Mattarella-Micke, A., Maji, S., and Iyyer, M. (2020). "Exploring and predicting transferability across NLP tasks". In: *EMNLP*.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. (2020). "Federated learning with matched averaging". In: *ICLR*.

Wang, Z., Zhang, C., and Chaudhuri, K. (2022). "Thompson Sampling for Robust Transfer in Multi-Task Bandits". In: *ICML*.

Wimalawarne, K., Sugiyama, M., and Tomioka, R. (2014). "Multitask learning meets tensor factorization: task imputation via convex optimization". In: *NeurIPS* 27.

Wu, S., Zhang, H. R., and Ré, C. (2020). "Understanding and improving information transfer in multi-task learning". In: *ICLR*.

Yang, F., Zhang, H. R., Wu, S., Su, W. J., and Ré, C. (2021). "Analysis of information transfer from heterogeneous sources via precise high-dimensional asymptotics". In: *arXiv*.

Yang, Y. and Hospedales, T. (2017). "Deep multi-task representation learning: A tensor factorisation approach". In: *ICLR*.

Yeh, C.-K., Kim, J., Yen, I. E.-H., and Ravikumar, P. K. (2018). "Representer point selection for explaining deep neural networks". In: *Advances in neural information processing systems*.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. (2020). "Gradient surgery for multi-task learning". In: *NeurIPS* 33, pp. 5824–5836.

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. (2018). "Taskonomy: Disentangling task transfer learning". In: *CVPR*, pp. 3712–3722.

Zhang, J., Yu, Y., Li, Y., Wang, Y., Yang, Y., Yang, M., and Ratner, A. (2021a). "WRENCH: A Comprehensive Benchmark for Weak Supervision". In: *NeurIPS Datasets and Benchmarks Track*.

Zhang, M., Sohoni, N. S., Zhang, H. R., Finn, C., and Ré, C. (2022). "Correct-N-Contrast: A Contrastive Approach for Improving Robustness to Spurious Correlations". In: *ICML*.

Zhang, Y. and Yang, Q. (2021b). "A survey on multi-task learning". In: *IEEE Transactions on Knowledge and Data Engineering*.

A Results for Constructing Surrogate Models

Section 2.3 shows that the sample complexity for learning task models is linear in the number of source tasks. Here, we provide empirical evidence to support this result. We plot the convergence of task modeling on sixteen target tasks from three datasets described in Section 4.1. We measure the MSE between task model predictions and empirical training results on the holdout set of size 100, following the experimental setup described in Section 4.2. Figure 6 shows the results. The red line shows the variance of f, measured across five random seeds. We observe that:

- The MSE of task models consistently converges close to the variance of the prediction loss.
- The Spearman correlation coefficient between the predictions and the true performances is **0.8** on average.

Thus, we conclude that linear surrogate models can be accurately fitted with less than $8k$ samples, and the fitted model can accurately predict the performances of unsampled subsets.

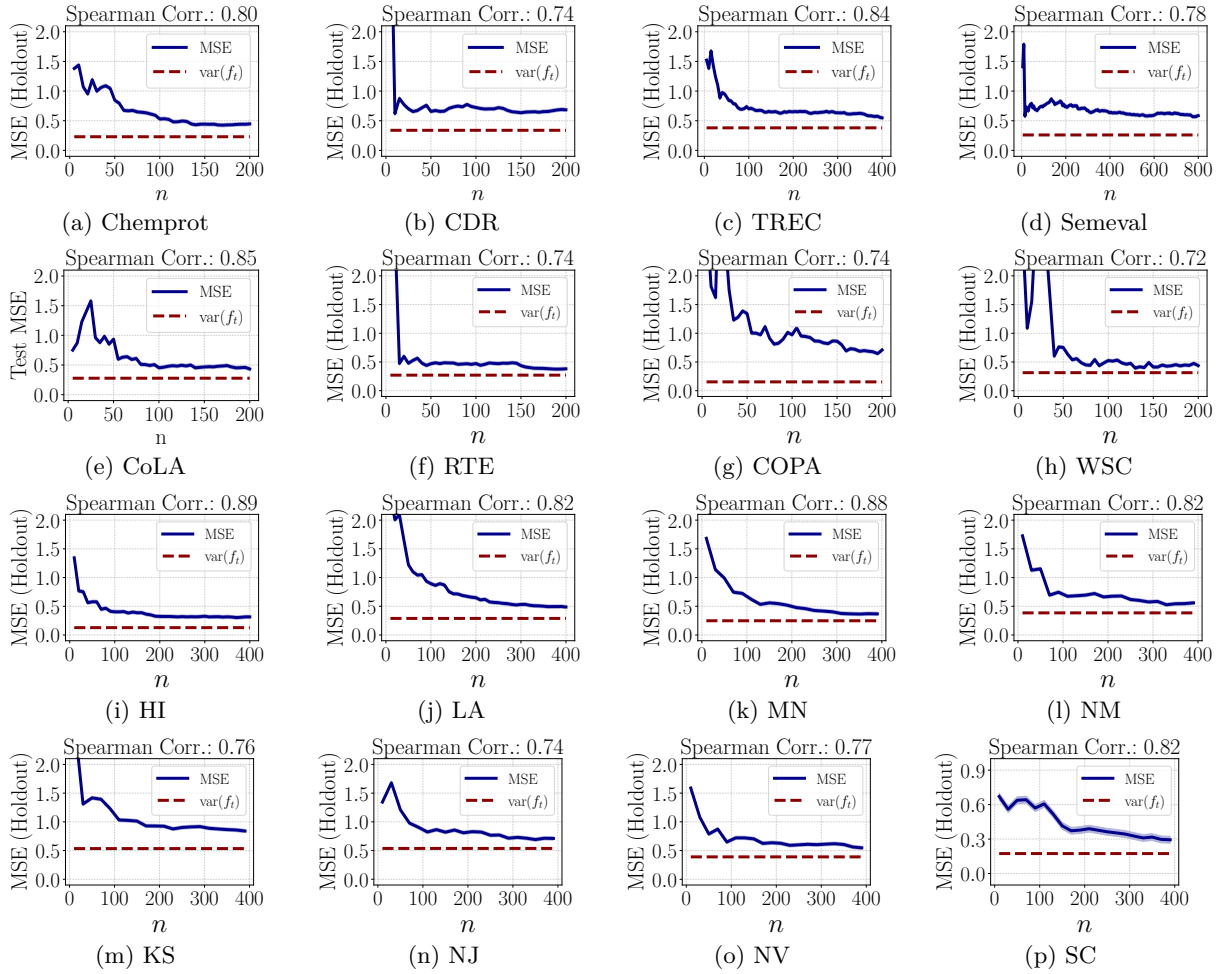


Figure 6: The MSE of linear surrogate models converges close to the variance of MTL performances. **(a-d)** Weak supervision tasks. **(e-h)** NLP tasks. **(i-p)** Multi-group learning tasks.

B Complete Proofs of Theoretical Results

Notations. We use $f(n) = (1 + o(1))g(n)$ to indicate that $|f(n) - g(n)|/g(n)$ approaches zero as n goes to infinity. For a matrix denoted as X, denote the spectral norm (or the largest singular value) of X as $\|X\|_2$. Denote the Frobenius norm of X as $\|X\|_F$.

B.1 Proof of Lemma 2.2

In the first part of the proof, we prove the convergence from $\hat{\theta}$ to $\bar{\theta}$ by dealing with the randomness of $S_1, S_2, \ldots, S_n$. Recall that $\mathcal{U}$ is the uniform distribution over subsets of $\{1, 2, \ldots, k\}$ with size α. Let $|\mathcal{U}| = \binom{k}{\alpha}$ denote the number of subsets from $\mathcal{U}$.

Proof of Lemma 2.2. Recall the definitions of $\hat{\theta}$ and $\bar{\theta}$ from Section 2. They can be written equivalently as follows:

$$\hat{\theta} = \left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} \frac{v}{n} \quad \text{and} \quad \bar{\theta} = \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1} \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{U}|}.$$

We will use the triangle inequality to separate the error between $\hat{\theta}$ and $\bar{\theta}$ into two parts:

$$\begin{aligned}
\left\|\hat{\theta} - \bar{\theta}\right\| &= \left\| \left(\left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} - \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1} \right) \frac{v}{n} + \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1} \left(\frac{v}{n} - \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{U}|}\right) \right\| \\
&\leq \left\| \left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} - \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1} \right\|_2 \cdot \left\| \frac{v}{n} \right\| && (14) \\
&\quad + \left\| \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1} \right\|_2 \cdot \left\| \frac{v}{n} - \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{U}|} \right\|. && (15)
\end{aligned}$$

We compare $\frac{v}{n}$ and $\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{U}|}$. Recall that both vectors have k coordinates, each corresponding to one task. For any task $i = 1, \ldots, k$, let $\mathcal{E}_i$ denote the difference between the i-th coordinate of $\frac{v}{n}$ and the i-th coordinate of $\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{U}|}$:

$$\mathcal{E}_i = \frac{1}{n} \sum_{1 \leq j \leq n:\ i \in S_j} f(S_j) - \frac{1}{|\mathcal{U}|} \sum_{T \in \mathcal{U}:\ i \in T} f(T). \tag{16}$$

Notice that the sampling of $S_1, S_2, \ldots, S_n$ is independent of the randomness in f. Therefore, we have that the expectation of $\mathcal{E}_i$ is zero:

$$\mathbb{E}\left[\mathcal{E}_i\right] = 0, \text{ for any } i = 1, 2, \ldots, k.$$

Next, we apply Chebyshev's inequality to analyze the deviation of $\mathcal{E}_i$ from its expectation. We consider the variance of $\mathcal{E}_i$, which is equal to the expectation of $\mathcal{E}_i^2$ since the mean of $\mathcal{E}_i$ is zero:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{E}_i^2\right] &= \mathbb{E}\left[\left(\frac{1}{n} \sum_{1 \leq j \leq n:\ i \in S_j} f(S_j) - \frac{1}{|\mathcal{U}|} \sum_{T \in \mathcal{U}:\ i \in T} f(T) \right)^2 \right] \\
&= \mathbb{E}\left[\frac{1}{n^2} \left(\sum_{1 \leq j \leq n: i \in S_j} f(S_j) \right)^2 - \frac{2}{n\,|\mathcal{U}|} \sum_{1 \leq j \leq n:\ i \in S_j} f(S_j) \sum_{T \in \mathcal{U}:\ i \in T} f(T) + \frac{1}{|\mathcal{U}|^2} \Big(\sum_{T \in \mathcal{U}:\ i \in T} f(T) \Big)^2 \right]
\end{aligned} \tag{17}$$

Notice that for any $T \in \mathcal{U}$ such that $i \in T$, the probability that T is sampled in the training dataset of size n is equal to

$$\frac{\binom{|\mathcal{U}|-1}{n-1}}{\binom{|\mathcal{U}|}{n}} = \frac{n}{|\mathcal{U}|}.$$

For any two subsets $T \neq T'$ that are both from $\mathcal{U}$ such that $i \in T$ and $i \in T'$, the probability that T and T' are both sampled in the training set (of size n) is equal to

$$\frac{\binom{|\mathcal{U}|-1}{n-1}}{\binom{|\mathcal{U}|}{n}} \cdot \frac{\binom{|\mathcal{U}|-1}{n-1}}{\binom{|\mathcal{U}|}{n}} = \frac{n^2}{|\mathcal{U}|^2}.$$

Thus, by taking the expectation over the randomness of the sampled subsets in equation (17) conditional on f, we can cancel out the cross terms for every pair of two tasks $i \neq i'$, leaving only the squared terms as:

$$\mathbb{E}\left[\mathcal{E}_i^2\right] = \mathbb{E}\left[\left(\frac{1}{n^2}\frac{n}{|\mathcal{U}|} - \frac{2}{n\,|\mathcal{U}|}\frac{n}{|\mathcal{U}|} + \frac{1}{|\mathcal{U}|^2}\right)\sum_{T\in\mathcal{U}:\ i\in T}\left(f(T)\right)^2\right] \leq \frac{C^2}{n}\cdot\frac{|T\in\mathcal{U}:\ i\in T|}{|\mathcal{U}|} \leq \frac{C^2}{n},$$

since the value of f is bounded from above by an absolute constant C. Therefore,

$$\mathbb{E}\left[\sum_{i=1}^{k}\mathcal{E}_i^2\right] \leq \frac{C^2 k}{n}.$$

By Markov's inequality, for any $a > 0$,

$$\Pr\left[\sum_{i=1}^{k}\mathcal{E}_i^2 \geq \frac{a^2 k}{n}\right] \leq \frac{C^2}{a^2}.$$

Therefore, with probability at least $1-\delta$, for any $\delta > 0$, conditional on the randomness of f, we have that

$$\left\|\frac{v}{n} - \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{U}|}\right\| \leq C\sqrt{\frac{k}{\delta n}}. \tag{18}$$

Next, we use random matrix concentration results to analyze the difference between the indicator matrix of the sampled subsets and the indicator matrix of all subsets in $\mathcal{U}$. Denote by

$$E = \frac{\mathcal{I}_n^\top \mathcal{I}_n}{n} - \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|} \quad \text{and} \quad A = \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}.$$

By the Sherman-Morrison formula calculating matrix inversions, we get

$$\begin{aligned}
\left\|\left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} - \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1}\right\|_2 &= \left\|(E+A)^{-1} - A^{-1}\right\|_2 \\
&= \left\|A^{-1}\Big(AE^{-1} + \mathrm{Id}_{k\times k}\Big)^{-1}\right\|_2 \\
&= \left\|A^{-1}E\Big(A+E\Big)^{-1}\right\|_2 \\
&\leq \big(\lambda_{\min}(A)\big)^{-1}\cdot\|E\|_2\cdot\big(\lambda_{\min}(A+E)\big)^{-1} \\
&\leq \frac{\|E\|_2}{\lambda_{\min}(A)(\lambda_{\min}(A) - \|E\|_2)}.
\end{aligned} \tag{19}$$

We now use the matrix Bernstein inequality (cf. Theorem 6.1.1 in Tropp (2015)) to deal with the spectral norm of E. Let

$$X_i = \mathbb{1}_{S_i}\mathbb{1}_{S_i}^\top - \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{U}|}, \text{ for any } \ i = 1,\ldots,n.$$

In expectation over $\mathcal{U}$, we know that $\mathbb{E}\left[X_i\right] = 0$, for any $i = 1,\ldots,n$. Additionally, $\|X_i\|_2 \leq 2\alpha$, since it is a linear combination of indicator vectors with α entries of ones in each indicator vector. Therefore, for all $t \geq 0$,

$$\Pr\left[\|E\|_2 \geq t\right] = \Pr\left[\left\|\sum_{i=1}^{n} X_i\right\|_2 \geq nt\right] \leq 2k\cdot\exp\left(-\frac{(nt)^2/2}{(2\alpha)^2 n + (2\alpha)nt/3}\right).$$

With some standard calculations, this implies that for any $\delta \geq 0$, with probability at least $1-\delta$,

$$\|E\|_2 \leq \frac{4\alpha\cdot\log\left(2k\delta^{-1}\right)}{\sqrt{n}}. \tag{20}$$

By applying equation (18) into equation (14) and equation (20) into equation (15), we have shown that with probability at least $1-2\delta$, for any $\delta \geq 0$,

$$\left\|\hat{\theta}-\bar{\theta}\right\| \leq \left\|\frac{v}{n}\right\|_2 \cdot \frac{4\alpha \cdot \log\left(2k\delta^{-1}\right)}{\sqrt{n}} + \frac{1}{\left(\lambda_{\min}(A)\right)^2\left(\lambda_{\min}(A)-\|E\|_2\right)} \cdot C\sqrt{\frac{k}{\delta n}}. \tag{21}$$

Lastly, we examine the norm of $\frac{v}{n}$. Let z_i be the number of subsets S_j among $1 \leq j \leq n$ such that $i \in S_j$, for any $i = 1, \ldots, n$. Recall that the value of f is bounded from above by an absolute constant C. Thus, based on the definition of v from equation (8), we have:

$$\left\|\frac{v}{n}\right\| \leq \frac{1}{n}\sqrt{C^2\sum_{i=1}^{k} z_i^2} \leq \frac{C}{n}\left(\sum_{i=1}^{k} z_i\right) = C\alpha, \tag{22}$$

since the size of each subset is strictly equal to α.

Regarding the minimum eigenvalue of A, notice that the diagonal entry of $\frac{\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}}{|\mathcal{U}|}$ is equal to $\binom{k-1}{\alpha-1}$. The off-diagonal entries of this matrix are equal to $\binom{k-2}{\alpha-2}$. Thus, based on standard algebra, one can prove that

$$\lambda_{\min}(A) \geq 1 - \frac{\binom{k-2}{\alpha-2}}{\binom{k-1}{\alpha-1}} = 1 - \frac{\alpha-1}{k-1} \geq 1 - \frac{\alpha}{k}. \tag{23}$$

Applying equations (22) and (23) back into equation (21), we conclude that with probability at least $1-2\delta$, $\hat{\theta}$ the estimation error between $\hat{\theta}$ and $\bar{\theta}$ grows at a rate of $\sqrt{\frac{k}{n}}$ as follows:

$$\left\|\hat{\theta}-\bar{\theta}\right\| \leq 4C\alpha^2\log(2k\delta^{-1}) \cdot \sqrt{\frac{k}{n}} + \left(1-\frac{\alpha}{k}\right)^{-3} C\alpha \cdot \sqrt{\frac{k}{\delta n}}.$$

Thus, we have proved that equation (9) holds, and the proof is complete. □

B.2 Proof of Lemma 2.3

In the second part, we prove the convergence from $\bar{\theta}$ to $\theta^\star$ by dealing with the randomness of f.

Proof of Lemma 2.3. Based on the definitions of $\bar{\theta}$ and $\theta^\star$, their difference can be written as follows:

$$\begin{aligned}
\|\bar{\theta}-\theta^\star\| &= \left\|\left(\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}\right)^{-1}\boldsymbol{\mathcal{I}}^\top\left(\boldsymbol{f}-\mathbb{E}\left[\boldsymbol{f}\right]\right)\right\| \\
&\leq \left\|\left(\frac{\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1}\frac{\boldsymbol{\mathcal{I}}^\top}{\sqrt{|\mathcal{U}|}}\right\|_2 \cdot \left\|\frac{\boldsymbol{f}-\mathbb{E}\left[\boldsymbol{f}\right]}{\sqrt{|\mathcal{U}|}}\right\| \\
&= \sqrt{\left(\frac{\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\right)^{-1}} \cdot \frac{\|\boldsymbol{f}-\mathbb{E}\left[\boldsymbol{f}\right]\|}{\sqrt{|\mathcal{U}|}} \\
&\leq \left(1-\frac{\alpha}{k}\right)^{-\frac{1}{2}} \cdot \frac{\|\boldsymbol{f}-\mathbb{E}\left[\boldsymbol{f}\right]\|}{\sqrt{|\mathcal{U}|}}. \qquad \text{(by equation (23))}
\end{aligned} \tag{24}$$

For each subset $T \in \mathcal{U}$, recall that $f(T)$ is the MTL outcome of combining the datasets of all tasks of T with the main target task. We will apply a Rademacher complexity-based generalization bound to analyze the generalization error $f(T) - \mathbb{E}\left[f(T)\right]$. Recall the Rademacher complexity of $\mathcal{F}$ with m samples from $\mathcal{D}_t$ is defined in equation (5). By Bartlett et al. (2002, Theorem 5), with probability at least $1-\delta$, we can get:

$$f(T) \leq \mathbb{E}\left[f(T)\right] + \frac{\mathcal{R}_m(\mathcal{F})}{2} + \sqrt{\frac{\log\left(1/\delta\right)}{2m}}. \tag{25}$$

Similarly, one can get the result for the other directions of the error estimate. With a union bound over all subsets $T \in \mathcal{U}$, with probability at least $1-\delta$, we get:

$$f(T) \leq \mathbb{E}\left[f(T)\right] + \frac{\mathcal{R}_m(\mathcal{F})}{2} + \sqrt{\frac{\alpha \log\left(\frac{k}{\delta}\right)}{2m}}, \text{ for all } T \in \mathcal{U}, \tag{26}$$

since

$$\log\left(\frac{\binom{k}{\alpha}}{\delta}\right) \leq \alpha \log\left(\frac{k}{\delta}\right).$$

Let $z = \sqrt{\alpha \log\left(k\delta^{-1}\right)/(2m)}$. Applying equation (26) back into equation (24), we have shown

$$\begin{aligned}\left\|\bar{\theta} - \theta^\star\right\| &\leq \left(1 - \frac{\alpha}{k}\right)^{-\frac{1}{2}} \sqrt{\frac{1}{|\mathcal{U}|}\sum_{T\in\mathcal{U}}\left(\frac{\mathcal{R}_m(\mathcal{F})}{2} + z\right)^2} \\ &= \left(1 - \frac{\alpha}{k}\right)^{-\frac{1}{2}} \left(\frac{\mathcal{R}_m(\mathcal{F})}{2} + z\right).\end{aligned}$$

Thus, based on the condition that $\alpha \leq k/2$, the proof of equation (10) is complete. □

Proof of Theorem 2.1. Notice that equation (6) follows by combining equation (9) from Lemma 2.2 and equation (10) from Lemma 2.3, together with the condition that $\alpha \leq 1/2$. Thus, the proof of the theorem is finished.

B.3 Convergence of the Empirical Risk

Based on the results from Lemma 2.2 and Lemma 2.3, we can also prove the convergence of the loss values. This is stated precisely in the following result.

Corollary B.1 (of Theorem 2.1)**.** *In the setting of Theorem 2.1, we have that*

$$\mathcal{L}(\theta^\star) - \hat{\mathcal{L}}_n(\hat{\theta}) \lesssim C\alpha \cdot \mathcal{R}_m(\mathcal{F}) + C\alpha^{1.5}\sqrt{\frac{\log(\delta^{-1}k)}{m}} + C^2\alpha^{3.5}\log\left(\frac{k}{\delta}\right)\sqrt{\frac{k}{n}} + C^2\alpha^{2.5}\sqrt{\frac{k}{\delta n}}. \tag{27}$$

Proof. To analyze the generalization error of $\hat{\mathcal{L}}_n(\hat{\theta})$, based on equation (2), we can expand out the loss term as

$$\begin{aligned}\hat{\mathcal{L}}_n(\hat{\theta}) &= \frac{1}{n}\left\|\mathcal{I}_n\hat{\theta} - \hat{f}\right\|^2 \\ &= \frac{1}{n}\left\|\mathcal{I}_n\hat{\theta} - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] + \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] - \hat{f}\right\|^2 \\ &= \frac{1}{n}\left\|\mathcal{I}_n\hat{\theta} - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\|^2 + \frac{2}{n}\langle\mathcal{I}_n\hat{\theta}_n - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right], \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] - \hat{f}\rangle + \frac{1}{n}\left\|\mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] - \hat{f}\right\|^2.\end{aligned} \tag{28}$$

Based on Lemma 2.2, the distance between $\hat{\theta}$ and $\theta^\star$ is at the order of $\mathrm{O}(n^{-1/2})$ with high probability. We will use this result to deal with the first term in equation (28) as follows:

$$
\begin{aligned}
&\frac{1}{n}\left\|\mathcal{I}_n\hat{\theta}_n - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\|^2 - \frac{1}{n}\left\|\mathcal{I}_n\theta^\star - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\|^2 \qquad (29)\\
=&\left|\frac{1}{n}\langle \mathcal{I}_n^\top\mathcal{I}_n, \hat{\theta}(\hat{\theta})^\top - \theta^\star(\theta^\star)^\top\rangle - \frac{2}{n}\langle \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right], \hat{\theta}-\theta^\star\rangle\right|\\
\le&\left\|\frac{1}{n}\mathcal{I}_n^\top\mathcal{I}_n\right\|_2 \cdot \left\|\theta^\star(\theta^\star)^\top - \hat{\theta}(\hat{\theta})^\top\right\|_F + \frac{2}{n}\left\|\mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\| \cdot \left\|\theta^\star - \hat{\theta}\right\| \qquad \text{(by triangle inequality)}\\
\le&\alpha\left\|\theta^\star(\theta^\star)^\top - \hat{\theta}(\hat{\theta})^\top\right\|_F + 2C\alpha\cdot e_1,
\end{aligned}
$$

where e_1 denotes the right hand side of equation (6). In the last step, the first part uses the fact that $\mathcal{I}_n^\top\mathcal{I}_n/n$ is the average of n rank one matrices, each with spectral norm α since they have exactly α ones. The second part uses an argument similar to equation (22) and the result of equation (6). Next,

$$
\begin{aligned}
\left\|\theta^\star(\theta^\star)^\top - \hat{\theta}_n(\hat{\theta}_n)^\top\right\|_F &= \left\|\theta^\star(\theta^\star-\hat{\theta}_n)^\top + (\theta^\star-\hat{\theta}_n)(\hat{\theta}_n)^\top\right\|_F\\
&\le \left\|\theta^\star(\theta^\star-\hat{\theta}_n)^\top\right\|_F + \left\|(\theta^\star-\hat{\theta}_n)(\hat{\theta})^\top\right\|_F \qquad \text{(by triangle inequality)}\\
&\le \Big(\|\theta^\star\| + \left\|\hat{\theta}\right\|\Big)e_1. \qquad \text{(by equation (6))}
\end{aligned}
$$

We show that the norm of $\theta^\star$ and $\hat{\theta}_n$ are both bounded by a constant factor times $\sqrt{k}$. To see this,

$$
\begin{aligned}
\|\theta^\star\| &= \left\|(\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}})^{-1}\boldsymbol{\mathcal{I}}^\top\,\mathbb{E}\left[\boldsymbol{f}\right]\right\|\\
&\le \left\|\Big(\frac{\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}}{|\mathcal{U}|}\Big)^{-1}\right\|_2 \cdot \left\|\frac{\boldsymbol{\mathcal{I}}^\top\,\mathbb{E}\left[\boldsymbol{f}\right]}{|\mathcal{U}|}\right\|\\
&\le \Big(1-\frac{\alpha}{k}\Big)^{-1}\cdot C\sqrt{\alpha} \qquad \text{(by equation (23) and the condition that } f \text{ is bounded by } C)
\end{aligned}
$$

Notice that the spectral norm of the difference between $\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}/|\mathcal{U}|$ and $\mathcal{I}_n^\top\mathcal{I}_n/n$ has been analyzed in equation (20). Thus, with similar steps as above, we can show that

$$
\left\|\hat{\theta}\right\| \le \left(\Big(1-\frac{\alpha}{k}\Big)^{-1} + \frac{4\alpha\log\left(2k\delta^{-1}\right)}{\sqrt{n}}\right)C\sqrt{k}.
$$

To wrap up our analysis above, we have shown that equation (29) is at most

$$
e_3 = \alpha\left(2(1-\alpha/k)^{-1} + \frac{4\alpha\log\left(2k\delta^{-1}\right)}{\sqrt{n}}\right)C\sqrt{\alpha}\cdot e_1 + 2C\alpha\cdot e_1.
$$

Next, we consider the second term in equation (28). Let $e_2 = \frac{\mathcal{R}_m(\mathcal{F})}{2} + \sqrt{\frac{\alpha\log(k/\delta)}{2m}}$ be the deviation error indicated in equation (26). Thus, every entry of $\hat{f} - \mathbb{E}\left[\hat{f}\right]$ is at most e_2. Besides, each entry of $\mathcal{I}_n\hat{\theta}_n - \mathbb{E}\left[\hat{f}\right]$ is less than

$$
\sqrt{\alpha}\|\hat{\theta}_n\| + C,
$$

because $\|\mathcal{I}_n\|_2 \le \sqrt{\alpha}$ and f is bounded from above by C. Thus, the second term in equation (28) is less than

$$
e_4 = e_2\left(\sqrt{\alpha}\cdot\Big(\Big(1-\frac{\alpha}{k}\Big)^{-1} + \frac{4\alpha\log\left(2k\delta^{-1}\right)}{\sqrt{n}}\Big)C\sqrt{\alpha} + C\right).
$$

For the population loss $\mathcal{L}(\theta^\star)$, notice that

$$\begin{aligned}\mathcal{L}(\theta^\star) &= \mathop{\mathbb{E}}_{\boldsymbol{f}}\left[\frac{1}{|\mathcal{U}|}\left\|\boldsymbol{\mathcal{I}}\theta^\star - \boldsymbol{f}\right\|^2\right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{f}}\left[\frac{1}{|\mathcal{U}|}\left\|\boldsymbol{\mathcal{I}}\theta^\star - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}] + \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}] - \boldsymbol{f}\right\|^2\right] \\ &= \frac{1}{|\mathcal{U}|}\left\|\boldsymbol{\mathcal{I}}\theta^\star - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}]\right\|^2 + \frac{1}{|\mathcal{U}|}\left(\mathop{\mathbb{E}}_{\boldsymbol{f}}\left[\left\|\boldsymbol{f} - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}]\right\|^2\right]\right)\end{aligned} \tag{30}$$

We know that each entry of $\boldsymbol{\mathcal{I}}\theta^\star - \mathbb{E}[\boldsymbol{f}]$ is at most $(1-\alpha/k)^{-1}\sqrt{\alpha} + C$. Thus, by Hoeffding's inequality, with probability at least $1-\delta$, we have

$$\left|\frac{1}{n}\left\|\mathcal{I}_n\theta^\star - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\| - \frac{1}{|\mathcal{U}|}\left\|\boldsymbol{\mathcal{I}}\theta^\star - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}]\right\|\right| \leq \left((1-\alpha/k)^{-1}\sqrt{\alpha} + C\right)\sqrt{\frac{\log\left(\delta^{-1}\right)}{n}}. \tag{31}$$

Lastly, we consider the third term in equation (28), compared with the second term in equation (30). For every $T \in \mathcal{U}$, let $e_T = f(T) - \mathbb{E}[f(T)]$. By equation (26), we know that e_T is of order $O(m^{-1/2})$, for every $T \in \mathcal{U}$. Therefore

$$\left|\frac{1}{n}\sum_{i=1}^{n} e_{S_i}^2\right| \leq \left(\frac{\mathcal{R}_m(\mathcal{F})}{2} + \sqrt{\frac{\alpha\log(k/\delta)}{2m}}\right)^2, \tag{32}$$

which is of order $O(m^{-1})$. Similarly, the same holds for the variance of $\boldsymbol{f}$ in the second term of equation (30).

Comparing equations (31) and (28), we have shown that

$$\begin{aligned}\mathcal{L}(\theta^\star) - \hat{\mathcal{L}}_n(\hat{\theta}) &\leq \left((1-\alpha/k)^{-1}\sqrt{\alpha} + C + C^2\right)\sqrt{\frac{\log(\delta^{-1})}{n}} + C\cdot e_2 + e_3 + e_4 \\ &\lesssim (C + C\alpha)\left(\mathcal{R}_m(\mathcal{F}) + \frac{\sqrt{\alpha\log(k\delta^{-1})}}{\sqrt{m}}\right) + \frac{C^2\alpha^{7/2}\log\left(2k\delta^{-1}\right) + 8C^2\alpha^{5/2}\delta^{-1/2}\sqrt{k}}{\sqrt{n}}.\end{aligned}$$

The above follows by incorporating the definitions of the error terms e_2, e_3, e_4. Thus, we have proved that equation (27) holds. The proof is now finished. □

B.4 Proof of Theorem 3.1

Recall that $\mathcal{I}_n \in \{0,1\}^{n\times k}$ is the indicator matrix corresponding to the task indices from the training dataset. Given a set of tasks S with size α, denote their feature matrices and label vectors as (X_1, Y_1), (X_2, Y_2), $\ldots$, (X_α, Y_α). With hard parameter sharing (Yang et al., 2021), we minimize

$$\ell(B) = \sum_{i=1}^{\alpha}\left\|X_iB - Y_i\right\|^2. \tag{33}$$

The minimizer of $\ell(B)$, denoted as $\hat{B}$, is equal to the following

$$\hat{B} = \left(\sum_{i=1}^{\alpha} X_i^\top X_i\right)^{-1}\left(\sum_{i=1}^{\alpha} X_i^\top Y_i\right).$$

For isotropic covariates, by matrix concentration results, the loss of using B on the validation set of the target task is equal to

$$f(S) = \left\|\hat{B} - \beta^{(t)}\right\|^2 + O\left(\sqrt{\frac{p}{m}}\right).$$

First, we state the proof of Lemma 3.3 from Section 3.1.

Proof of Lemma 3.3. We have that $Y_i = X_i\beta^{(i)} + \epsilon^{(i)}$, where $\epsilon^{(i)}$ is a random vector whose entries are sampled independently with mean 0 and variance σ^2. We have

$$f(S) = \left\| \left(\sum_{i=1}^{\alpha} X_i^\top X_i \right)^{-1} \sum_{i=1}^{\alpha} X_i^\top \epsilon^{(i)} \right\|^2. \tag{34}$$

For a task i, we know that its coefficient is equal to the i-th entry of

$$\left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} \frac{\mathcal{I}_n^\top \hat{f}}{n}.$$

Let $Z = \mathcal{I}_n^\top \mathcal{I}_n / n$. By equation (12), for any $i \neq j$, we observe that

$$\begin{aligned}
\left| \frac{\hat{\theta}_i - \hat{\theta}_j}{n} - \frac{k}{\alpha} \cdot \frac{v_i - v_j}{n} \right| &= \left| (e_i - e_j)^\top \left(Z^{-1} - \mathbb{E}\left[Z\right]^{-1} \right) \frac{v}{n} \right| \\
&\leq \|e_i - e_j\| \cdot \left\| Z^{-1} - \mathbb{E}\left[Z\right]^{-1} \right\|_2 \cdot \left\| \frac{v}{n} \right\| \\
&\leq 2C\alpha \cdot \left\| Z^{-1} - \mathbb{E}\left[Z\right]^{-1} \right\|_2 && \text{(by equation (22))} \\
&\leq \frac{4\alpha \log\left(2k\delta^{-1}\right)}{\sqrt{n}} \frac{2}{(1-\alpha/k)^2}. && \text{(by equations (19), (20), (23))}
\end{aligned}$$

The last step follows by applying equations (20) and (23) into equation (19). Thus, we have finished the proof of equation (11). □

Second, we show that provided n, and d are sufficiently large, a separation exists in the coefficients of v between good and bad tasks.

Proof of Theorem 3.1. We calculate v_i/n for all $i = 1, \ldots, k$ and compare its value between a good task and a bad task. We first compare their expectations over the randomly sampled subsets. By equation (18), we get

$$\left| \frac{v_i}{n} - \frac{1}{|\mathcal{U}|} \sum_{T \in \mathcal{U}:\ i \in T} f(T) \right| \leq \frac{Ck\delta^{-1/2}}{\sqrt{n}}, \text{ and}$$
$$\left| \frac{v_j}{n} - \frac{1}{|\mathcal{U}|} \sum_{T \in \mathcal{U}:\ j \in T} f(T) \right| \leq \frac{Ck\delta^{-1/2}}{\sqrt{n}}.$$

Therefore, by applying the triangle inequality with the above two results, we get

$$\left| \frac{v_i - v_j}{n} - \frac{\sum_{T \in \mathcal{U}: i \in T} f(T) - \sum_{T \in \mathcal{U}: j \in T} f(T)}{|\mathcal{U}|} \right| \leq \frac{2Ck\delta^{-1/2}}{\sqrt{n}}. \tag{35}$$

To deal with equation (35), we apply a union bound over the sample covariance matrix of every subset T in $\mathcal{U}$ to show that they are close to their expectation. By Gaussian covariance estimation results (e.g., Wainwright (2019, equation (6.12))), for a fixed $T \in \mathcal{U}$ such that $T = \{i_1, i_2, \ldots, i_\alpha\}$, we get

$$\left| \frac{1}{\alpha d} \sum_{j \in T} X_j^\top X_j - \mathrm{Id}_{p \times p} \right| \leq 2\sqrt{\frac{p}{\alpha d}} + 2\epsilon + \left(\sqrt{\frac{p}{\alpha d}} + \epsilon \right)^2, \tag{36}$$

with probability at least $1 - 2\exp\left(-\frac{1}{2}\alpha d \epsilon^2\right)$. With a union bound over all $T \in \mathcal{U}$, we have that the above holds with probability at least $1 - \delta$ for all $T \in \mathcal{U}$, for ϵ that is equal to

$$\epsilon = \sqrt{\frac{2\alpha k \log(2k\delta^{-1})}{\alpha d}}.$$

Let ε_1 denote the error term from equation (36), by inserting the value of ϵ:

$$\varepsilon_1 = 2\sqrt{\frac{p}{\alpha d}} + 2\sqrt{\frac{2\alpha \log(2k\delta^{-1})}{\alpha d}} + \left(\sqrt{\frac{p}{\alpha d}} + \epsilon\right)^2.$$

Let

$$u_T = \frac{1}{\alpha d} \sum_{j \in T} X_j^\top \epsilon^{(j)}, \text{ for any } T \in \mathcal{U}.$$

One can verify that

$$\left| f(T) - \|u_T\|^2 \right| \leq \left((1-\varepsilon_1)^{-2} - 1\right) \|u_T\|^2 \leq 3\varepsilon_1 \|u_T\|^2.$$

Notice that

$$\mathbb{E}\left[\|u_T\|^2\right] = \mathbb{E}\left[\frac{1}{(\alpha d)^2} \operatorname{Tr}\left[\sum_{j \in T} X_j^\top \varepsilon^{(j)} (\varepsilon^{(j)})^\top X_j\right]\right].$$

If j is a good task, then the expectation over $\varepsilon^{(j)}$ is equal to $a^2 \operatorname{Id}$ by the assumption of Theorem 3.1. If j is a bad task, on the other hand, then the expectation over $\varepsilon^{(t)}$ is equal to $b^2 \operatorname{Id}$.

Let $s(T)$ denote the number of good tasks in T, for any $T \subseteq \{1, 2, \ldots, k\}$. Thus,

$$\mathbb{E}\left[\|u_T\|^2\right] = \frac{p\Big(a^2 s(T) + b^2\big(\alpha - s(T)\big)\Big)}{\alpha^2 d}. \tag{37}$$

To argue about the deviation error of $\|u_T\|^2$, we use the following two estimates (see, e.g., Vershynin (2011)), which holds with high probability:

$$\left| \left(\varepsilon^{(j)}\right)^\top X_j X_j^\top \varepsilon^{(j)} - \mathbb{E}\left[\left(\varepsilon^{(j)}\right)^\top X_j X_j^\top \varepsilon^{(j)}\right] \right| \lesssim p\sqrt{d} a^2, \text{ for any } j = 1, \ldots, k;$$
$$\left| \left(\varepsilon^{(i)}\right)^\top X_i X_j^\top \varepsilon^{(j)} \right| \lesssim p\sqrt{d} a^2, \text{ for any } 1 \leq i < j \leq k.$$

Therefore, we get that for any $T \in \mathcal{U}$,

$$\left| \|u_T\|^2 - \mathbb{E}\left[\|u_T\|^2\right] \right| \leq \frac{p\sqrt{d} a^2}{d^2}. \tag{38}$$

To finish the proof, consider a good task i versus a bad task j. We need the gap in the expectation term between the good/bad tasks to dominate the standard deviation from the error terms. The gap in the expectations is based on equation (37). The standard deviation terms are upper bounded by the sum of equations (35) and (38).

Thus, provided that

$$(1 - 3\varepsilon_1)\frac{p(a^2 - b^2)}{\alpha^2 d} \geq (1 + 3\varepsilon_1)\frac{p\sqrt{d} a^2}{d^2} + \frac{2Ck\delta^{-1/2}}{\sqrt{n}}, \tag{39}$$

there must exist a threshold separating all the good tasks from the bad ones. We can verify that condition (39) is satisfied when

$$n \gtrsim C^2 \cdot k^2 \cdot \frac{1}{(a^2 - b^2)^2}, \quad \text{and}$$
$$d \gtrsim \left(\frac{a^2}{a^2 - b^2}\right)^2 k^4 + k \log\left(\frac{2k}{\delta}\right) + p.$$

To apply Algorithm 1, we set the threshold γ as k/α times any value between the left-hand and right-hand side of equation (39) (recall that k/α is inherited from Lemma 3.3). Thus, when n and d satisfy the condition above, combined with Lemma 3.3, with high probability, for any i such that $\hat{\theta}_i < \gamma$, i must be a good task. When $\hat{\theta}_i > \gamma$, i much be a bad task. Thus, we have finished the proof. □

C Experiment Details

We describe details that were left out of the paper's main text. First, we describe the additional experimental setup and the implementation specifics. Second, we present results to further validate the sample complexity of task modeling. Third, we provide the experimental results that are omitted from Section 4, including the results for fairness measures and ablation studies.

C.1 Implementation Details

For evaluating multitask learning with natural language processing tasks, we collect twenty-five tasks from several benchmarks, including GLUE, SuperGLUE, TweetEval, and ANLI. Due to the computation constraint, we did not include the tasks with a training set size larger than 100k. The collection spans numerous categories of tasks, including sentence classification, natural language inference, and question answering. Table 3 shows the statistics of the twenty-five tasks.

Table 3: Dataset description and statistics of twenty-five text datasets.

Task	Benchmark	Train. Set	Dev. Set	Task Category	Metrics
CoLA	GLUE	8.5k	1k	Grammar acceptability	Matthews corr.
MRPC	GLUE	3.7k	1.7k	Sentence Paraphrase	Acc./F1
RTE	GLUE	2.5k	3k	Natural language inference	Acc.
SST-2	GLUE	67k	1.8k	Sentence classification	Acc.
STS-B	GLUE	7k	1.4k	Sentence similarity	Pearson/Spearman corr.
WNLI	GLUE	634	146	Natural language inference	Acc.
BoolQ	SuperGLUE	9.4k	3.3k	Question answering	Acc.
CB	SuperGLUE	250	57	Natural language inference	Acc./F1
COPA	SuperGLUE	400	100	Question answering	Acc.
MultiRC	SuperGLUE	5.1k	953	Question answering	$F1_a$/EM
WiC	SuperGLUE	6k	638	Word sense disambiguation	Acc.
WSC	SuperGLUE	554	104	Coreference resolution	Acc.
Emoji	TweetEval	45k	5k	Sentence classification	Macro-averaged F1
Emotion	TweetEval	3.2k	374	Sentence classification	Macro-averaged F1
Hate	TweetEval	9k	1k	Sentence classification	Macro-averaged F1
Irony	TweetEval	2.9k	955	Sentence classification	$F1^{(i)}$
Offensive	TweetEval	12k	1.3k	Sentence classification	Macro-averaged F1
Sentiment	TweetEval	45k	2k	Sentence classification	Macro-averaged Recall
Stance (Abortion)	TweetEval	587	66	Sentence classification	Avg. of $F1^{(a)}$ and $F1^{(f)}$
Stance (Atheism)	TweetEval	461	52	Sentence classification	Avg. of $F1^{(a)}$ and $F1^{(f)}$
Stance (Climate)	TweetEval	355	40	Sentence classification	Avg. of $F1^{(a)}$ and $F1^{(f)}$
Stance (Feminism)	TweetEval	597	67	Sentence classification	Avg. of $F1^{(a)}$ and $F1^{(f)}$
Stance (H. Clinton)	TweetEval	620	69	Sentence classification	Avg. of $F1^{(a)}$ and $F1^{(f)}$
ANLI (A1)	ANLI	1.7k	1k	Natural language inference	Acc.
ANLI (A2)	ANLI	4.5k	1k	Natural language inference	Acc.

Next, we report the results for baselines by running the open-sourced implementations from the respective publications. We describe the hyperparameters for baselines as follows. For higher-order approximation and task affinity grouping, we compute the task affinity scores between source and target tasks. Then, we select m tasks with the largest task affinity scores as source tasks for each target task. m is searched between 0 and the number of total tasks.

For gradient decomposition, we search the number of decomposition basis and auxiliary task gradient direction parameters, following the search space in Dery et al. (2021).

For weighted training, we search the task weight learning rate in $[10^{-2}, 10^2]$. The hyper-parameters are tuned on the validation dataset by grid search. For each target task, we search 10 times over the hyper-parameter space. We use the same number of trials in tuning hyper-parameters for baselines.

Table 4: Accuracy/Correlation scores on the development set using surrogate modeling followed by thresholding (ours), as compared with STL and MTL methods.

Dataset Metrics	CoLA Matthews Corr.	RTE Accuracy	CB Accuracy	COPA Accuracy	WSC Accuracy
Train	8500	2500	250	400	554
Validation	1000	3000	57	100	104
STL	59.38±0.70	67.94±0.74	70.36±1.82	64.00±2.19	60.00±2.76
Naive MTL	57.11±0.81	69.31±0.97	71.78±1.39	66.00±2.02	58.20±1.98
HOA	60.09±0.75	69.03±2.03	80.71±2.62	67.20±2.56	61.35±3.12
Alg. 1 (Ours)	**60.43±0.79**	**70.83±1.97**	**83.57±2.43**	**69.20±3.71**	**65.38±2.31**

Table 5: Violation of two fairness-related measures (demographic parity and equality of opportunity) on six multi-group learning tasks with tabular features, averaged over ten random seeds. **Lower is better.**

Demographic parity	HI	KS	LA	NJ	NV	SC
STL	12.95±1.76	4.09±1.15	26.30±1.21	26.06±0.53	12.62±1.99	22.51±0.47
Naive MTL	8.25±1.31	4.06±1.17	21.24±0.66	27.73±0.94	13.35±0.51	18.83±0.80
HOA	8.63±2.95	6.15±3.00	22.83±0.53	26.14±0.29	13.15±0.64	19.39±1.05
TAG	8.93±2.35	3.97±0.61	20.72±0.86	25.21±0.68	12.24±0.82	18.77±0.85
TAWT	18.12±1.80	4.84±0.71	25.77±0.94	25.66±0.38	12.40±0.74	23.16±0.42
Alg. 1 (Ours)	**7.63±2.12**	**1.06±0.62**	**17.25±1.13**	**24.96±0.63**	**11.34±1.31**	**17.66±0.80**
Equality of opportunity	HI	KS	LA	NJ	NV	SC
STL	9.86±1.29	1.43±3.62	29.64±3.24	22.43±1.02	13.61±3.67	29.93±0.77
Naive MTL	3.86±0.84	2.03±2.11	21.26±1.35	24.43±1.49	12.14±2.21	21.22±1.75
HOA	3.55±2.85	4.34±3.18	22.88±1.72	22.98±1.18	12.92±2.23	23.31±1.77
TAG	4.27±0.25	1.18±0.97	20.66±1.43	21.89±0.69	11.66±1.58	19.89±1.10
TAWT	4.21±2.25	1.40±2.14	30.38±2.17	23.26±0.30	11.77±1.01	30.86±0.84
Alg. 1 (Ours)	**0.24±1.32**	**0.21±1.34**	**14.14±2.32**	**21.48±0.90**	**9.65±3.49**	**18.54±1.61**

C.2 Omitted Results from Section 4.3

Complete results for NLP tasks. In Table 4, we report the complete experimental results for applying our approach to NLP tasks, as reported in Section 4.3.

Optimizing fairness-related metrics. We show that task modeling is applicable to various performance metrics for capturing task affinity. Besides the average performance and worst-group performance discussed in Section 4.3, we consider two fairness measures: demographic parity and equality of opportunity (Ding et al., 2021).

The demographic parity measure is defined as:

$$\left|\Pr\left[\hat{y}=1 \mid g=\text{black}\right]-\Pr\left[\hat{y}=1 \mid g=\text{white}\right]\right|,$$

which measures the difference in the positive rates between white and African American demographic groups.

The equality of opportunity measure is defined as:

$$\left|\Pr\left[\hat{y}=1 \mid y=1, g=\text{black}\right]-\Pr\left[\hat{y}=1 \mid y=1, g=\text{white}\right]\right|,$$

which measures the difference in the true positive rates between the two groups.

We consider the binary classification tasks with multiple subpopulation groups. Table 5 shows the comparative results. First, similar to the worst-group accuracy results, we find that multitask approaches (including ours and previous methods) decrease the violation of both fairness measures compared to ERM, suggesting the benefit of combining related datasets. Second, our approach consistently reduces both fairness measure violations more by **1.26%** and **2.31%** on average than previous multitask learning approaches, respectively.

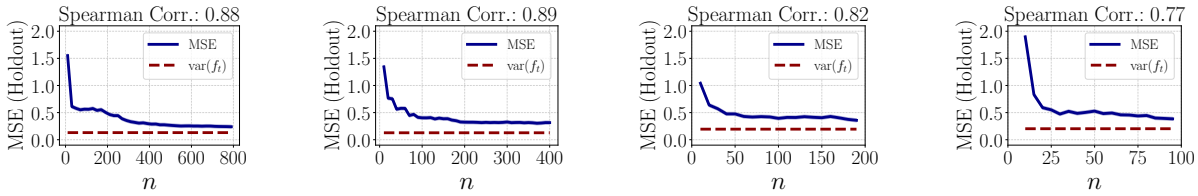
C.3 Ablation Studies for Constructing Surrogate Models

Loss function: We consider three choices of prediction losses, including zero-one accuracy, cross-entropy loss, and classification margin. We observe that using the classification margin is more effective than the other two metrics. The Spearman correlation of using the margin is 0.86 on average over two tasks (HI and LA). In contrast, the Spearman correlations of using the loss and accuracy are 0.61 and 0.34, respectively. Besides, we compare the task selection using the three metrics in Table 6. We find that using the margin outperforms the other two by 0.37% on average over the six target tasks in terms of worst-group accuracy.

	HI	KS	LA	NJ	NV	SC
f uses zero-one accuracy	75.16±0.70	76.39±1.09	75.15±0.43	77.40±0.49	74.34±1.81	77.29±0.19
f uses cross-entropy loss	75.33±0.80	75.82±0.60	74.19±1.37	77.51±0.35	74.55±1.60	77.21±0.27
f uses classification margin	75.47±0.73	76.96±0.69	75.62±0.11	78.17±0.36	75.21±0.52	77.62±0.34

Table 6: Choosing different loss functions ℓ for six target tasks in the multi-group learning setting.

Subset size: Recall that we collect training results by sampling n subsets from a uniform distribution over subsets of a constant size. We evaluate the MSE of task models by varying $\alpha \in \{2, 5, 10, 20\}$. To control the computation budget the same, we scale the number of subsets n according to α. We train $n = 800, 400, 200, 100$ models with $\alpha = 2, 5, 10, 20$, respectively. We observe similar convergence results as in Figure 7. Among them, $\alpha = 5$ yields a highest Spearman's correlation of 0.89 between $f(\cdot)$ and $g(\cdot)$.



(a) $\alpha = 2$ (b) $\alpha = 5$ (c) $\alpha = 10$ (d) $\alpha = 20$

Figure 7: Fitting surrogate models using different α evaluated on a fixed target task.

Number of sampled subsets: Lastly, we show that task selection remains stable under different values of n. We measure the effect on two tasks (HI and LA) by comparing the 10 tasks with the smallest coefficients estimated from $n = 100, 200, 400$ subsets. We observe that using 100 subsets identifies 7/10 source tasks compared with $n = 400$. Increasing n to 200 further identifies 9/10 source tasks compared with $n = 400$.