Task Modeling: Approximating Multitask Predictions for Cross-Task Transfer

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of learning a target task when data samples from several auxiliary source tasks are available. Examples of this problem appear in multitask learning, where several tasks are combined jointly, and weak supervision, where multiple programmatic labels are generated for each sample. Because of task data's heterogeneity, negative interference is a critical challenge for solving this problem. Previous works have measured first-order task affinity as an effective metric, yet it becomes less accurate for approximating higher-order transfers. We propose a procedure called task modeling to model first- and higher-order transfers. This procedure samples subsets of source tasks and estimates surrogate functions to approximate multitask predictions. We show theoretical and empirical results that task models can be estimated in nearly-linear time in the number of tasks and accurately approximate multitask predictions. Thus, the target task's performance can be optimized using task models to select source tasks. We validate this approach on various datasets and performance metrics. Our method increases accuracy up to 3.6% over existing methods on five text classification tasks with noisy supervision sources. Additionally, task modeling can be applied to group robustness and fairness metrics. Ablation studies show that task models can accurately predict whether or not a set of up to four source tasks transfer positively to the target task.

1 Introduction

Given a set of k auxiliary source tasks and a primary target task of interest, how can we select the beneficial ones for the target task? This question is motivated by a number of applications. In multitask learning [10, 13, 6], several tasks are learned simultaneously. The learned model can be further fine-tuned for a single task [29]. Depending on task relatedness, multitask learning may worsen performance compared to single task learning [7], a phenomenon known as negative transfer [31, 26]. Another example is weak supervision [35, 33]: each sample is annotated with multiple (possibly conflicting) labels, generated by labeling functions specified with domain knowledge. The labeling functions can be viewed as source tasks alongside the target task in a multitask model [34].

Early work shows that information sharing across tasks can be realized with explicit regularization in shallow linear and kernel models [17, 1, 58]. With deep neural networks, sharing information across tasks is more challenging [52]. A naive solution for finding the most beneficial source tasks is to search through all possible combinations of source tasks. However, this is prohibitively expensive as k grows. Another solution is to determine first-order task affinity by training one model for every source-target pair [42]. Such first-order task affinity can also be measured in the gradients during training [54, 14, 18]. These methods require training at most k models but ignore higher-order structures, such as the transfer from a set of source tasks to the target. Thus, higher-order approximations that

Submitted to the Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

average the first-order task affinity are used as a substitute [42]. In our experiments, we have observed that the accuracy of averaging deteriorates as the size of S grows (cf. Figure 2, Appendix B).

In this work, we propose an efficient method to model first- and higher-order transfer predictions. Let S be a subset of source tasks from $\{1, 2, \ldots, k\}$. Our approach estimates a surrogate function to approximate the prediction loss of combining S and a target task t, denoted as $f_t(S)$. If S is similar to t, $f_t(S)$ will be small; otherwise $f_t(S)$ will be large. Thus, extrapolating such multitask predictions provides a way to model higher-order task structures. Our method, called *task modeling*, fits the value of $f_t(S)$ of n random subsets S with linear regression. Figure 2 (in Appendix B) shows that task modeling remains highly correlated with $f_t(S)$ as $|S|$ grows. Additionally, task modeling accurately predicts whether a set of source tasks transfer positively to the target.

Results. We prove that the sample complexity of task modeling is $O(k\alpha^4 \log^2 k)$ for any $|S|$ up to order α (cf. Theorem 3.1). In particular, task modeling requires comparable runtime to compute first-order task affinity, but accelerates computing higher-order affinity from $O(k^\alpha)$ to a nearly linear time in k. With task modeling as a surrogate function of $f_t(S)$, finding the optimal S can be achieved with the task model, by selecting source tasks with negative model coefficients. The premise of this algorithm is that there exists one group of source tasks related to the target, while the rest are unrelated. In a linear parametric setting, we prove that our algorithm only selects related source tasks to the primary target task of interest (cf. Theorem 3.2).

We conduct a detailed empirical study of our methods on various datasets and performance metrics. First, we validate the benefit of modeling higher-order transfer for multitask learning and the efficiency of task modeling by detailing the computation costs. Second, we apply the task selection algorithm on five text classification tasks with noisy supervision sources [56], showing up to 3.6% accuracy improvement over all existing methods. Third, we show that task modeling can be used with group robustness and fairness metrics. On a tabular dataset where each task involves nine subpopulation groups [15], our approach consistently improves the worst-group accuracy over ten baselines.

2 Problem Setup

Consider a target task whose input features and class labels are drawn from an unknown distribution $\mathcal{D}_t$, supported on the product of a feature space $\mathcal{X}$ and a label space $\mathcal{Y}$. Suppose we have access to a training set $\hat{\mathcal{D}}_t$ and a validation set $\tilde{\mathcal{D}}_t$, both drawn from $\mathcal{D}_t$. Let N be the size of the validation set $\tilde{\mathcal{D}}_t$. Given a predictor $f : \mathcal{X} \to \mathbb{R}^k$ and a nonnegative function $\ell : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}^+$, the loss of a sample x, y is denoted as $\ell(f(x), y)$.

Suppose we have access to k related data distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$, called *source tasks*, which are supported on $\mathcal{X} \times \mathcal{Y}$. In cross-task transfer learning, we want to select a set of source tasks so that combining them with the target task optimizes the target task's performance. We assume that some of the source tasks are related to the target task, while many of them may negatively interfere (see Figure 3 in Appendix B). Thus, the problem is to select the related tasks out of the k source tasks.

A naive solution to this problem is to enumerate all combinations of source tasks. This requires training 2^k models, which is too costly. Another solution is to train k models, one for every source-target pair. Select all source tasks that provide a positive transfer to the target task. This idea trades off precision for efficiency and underlies several existing multitask learning approaches [42, 18]. Given that several source tasks will be combined with the target task, we consider higher-order transfer.

To capture higher-order transfer, we will consider a distribution $\mathcal{S}$ supported on subsets of a fixed size α. For instance, to capture how well five source tasks transfer to the target, $\mathcal{S}$ is a uniform distribution over subsets of $\{1, \ldots, k\}$ with size five. Later in Section 3.2, we argue that this distribution enjoys a certain covariance structure that preserves the gap between related and unrelated tasks.

3 Methodology

We present methods to model higher-order transfer and optimize cross-task transfer. Our approach estimates a surrogate function to approximate multitask prediction losses. We show that these functions can be estimated efficiently and predict the losses accurately. Thus, optimizing cross-task transfer can be done using the task models, leading to an algorithm for selecting source tasks.

3.1 Efficiently modeling higher-order transfer

We will estimate a surrogate function to approximate multitask predictions. Informally, this measures how well a set of source tasks transfer to the target task. Our method has two steps:

(i) Evaluate multitask predictions: For $i = 1, \ldots, n$, sample S_i from $\mathcal{S}$. Perform multitask training with the training samples in S_i. With a trained encode ϕ and the predictor ψ_t, evaluate the *multitask prediction* loss of S_i:

$$f_t(S_i) = \frac{1}{N} \sum_{(x,y) \in \tilde{\mathcal{D}}_t} \ell\big(\psi_t(\phi(x)), y\big). \tag{1}$$

(ii) Estimate surrogate functions: For $S \subseteq \{1, \ldots, k\}$, let $g(S) = \theta^\top \mathbb{1}_S$, parametrized by a k dimensional vector θ, where $\mathbb{1}_S \in \{0,1\}^k$ be the characteristic vector of whether or not a task is in S.. With n subsets and multitask predictions, estimate θ as:

$$\hat{\theta}_n \leftarrow \arg\min_{\theta \in \mathbb{R}^k} \hat{\mathcal{L}}_n(\theta) := \frac{1}{n} \sum_{i=1}^{n} \Big(\theta^\top \mathbb{1}_{S_i} - f_t(S_i)\Big)^2. \tag{2}$$

We analyze the sample complexity of estimating $\hat{\theta}_n$. To formulate the problem, notice that the population risk can be defined by taking the expectation over the randomness of f_t:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{f_t} \mathop{\mathbb{E}}_{T \sim \mathcal{S}} \left[\Big(\theta^\top \mathbb{1}_T - f_t(T)\Big)^2 \right]. \tag{3}$$

Let $\theta^\star$ be the population risk minimizer. Our result will depend on the Rademacher complexity of the function class. Additionally, we analyze the convergence of the empirical risk.

Theorem 3.1 (Proof in Appendix D.1)**.** *Suppose the functions in $\mathcal{F}$ are bounded by a fixed C. Suppose $\alpha \le k/2$. With probability at least $1 - \delta$, for any $\delta \ge 0$, $\hat{\theta}_n$ converges to $\theta^\star$:*

$$\left\| \hat{\theta}_n - \theta^\star \right\| \lesssim \mathcal{R}_N(\mathcal{F}) + \frac{\sqrt{\alpha \log(\delta^{-1} k)}}{\sqrt{N}} + \frac{C\alpha^2 \log(\delta^{-1}k)\sqrt{k}}{\sqrt{n}} + \frac{C\alpha\sqrt{\delta^{-1}k}}{\sqrt{n}}. \tag{4}$$

$\hat{\theta}_n$'s empirical risk converges to $\theta^\star$'s population risk:

$$\mathcal{L}(\theta^\star) - \hat{\mathcal{L}}_n\big(\hat{\theta}_n\big) \lesssim C\alpha\mathcal{R}_N(\mathcal{F}) + \frac{C\alpha^{3/2}\sqrt{\log(\delta^{-1}k)}}{\sqrt{N}} + \frac{C^2\alpha^{7/2}\log(\delta^{-1}k)\sqrt{k}}{\sqrt{n}} + \frac{C^2\alpha^{5/2}\sqrt{\delta^{-1}k}}{\sqrt{n}}. \tag{5}$$

This theorem implies that the sample complexity of estimating linear task models is only $O(k\alpha^4 \log^2 k)$—a nearly linear rate in the number of tasks. More broadly, the guarantee holds under mild conditions of the loss. It applies to group robustness and fairness measures in place of f_t.

Empirical examples. We verify that task modeling estimates an accurate approximation of f_t. We consider a tabular data with 50 source tasks and a text dataset with 24 source tasks. We evaluate the task model g on a holdout set. In both cases, we consider five-way multitask relations, i.e., $\mathcal{S}$ is a uniform distribution over all combinations of source tasks with size $\alpha = 5$. For tabular datasets,

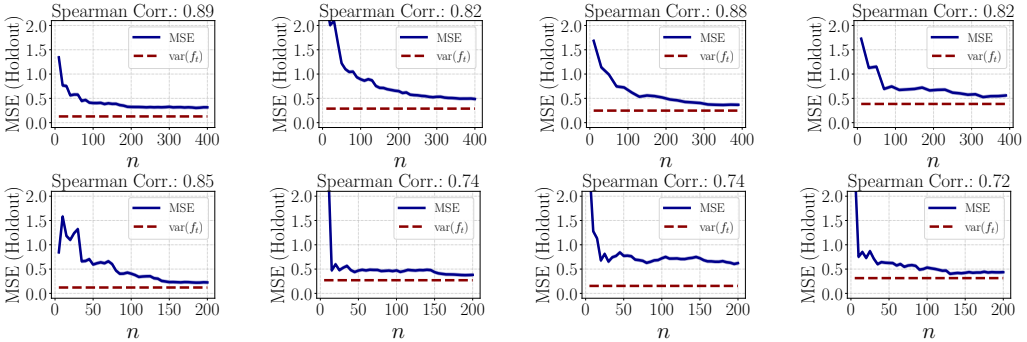


Figure 1: **(a)** The MSE of task modeling converges with less than $8k$ samples. **(b)** Task modeling approximates f_t accurately with a Spearman correlation of 0.8 on average. **Top**: Training one tabular target task along with subsets of **50** source tasks. **Bottom**: Training one text classification target task along with subsets of **24** source tasks. Appendix E.2 contains similar results with more target tasks.

Algorithm 1 Selecting source tasks using task modeling

Input: Training examples from source tasks $\hat{\mathcal{D}}_1, \ldots, \hat{\mathcal{D}}_k$; Training and validation sets of target task $\hat{\mathcal{D}}_t$ and $\tilde{\mathcal{D}}_t$.
Require: A multitask prediction loss function $f_t : 2^{\{1,2,\ldots,k\}} \to \mathbb{R}^+$; A distribution over subsets of source tasks $\mathcal{S}$; Number of subsets n; A threshold γ.

1: For $i = 1, \ldots, n$, sample a set S_i from $\mathcal{S}$, perform multitask training and evaluate $f_t(S_i)$.
2: Estimate the task model coefficients $\hat{\theta}_n$ following equation 2.
3: **Output**: Select source tasks $S^\star = \left\{i : \hat{\theta}_n(i) < \gamma, \text{ for any } 1 \le i \le k\right\}$.

we use a fully-connected layer as the encoder. For text datasets, we use BERT-mini as the encoder. Figure 1 plots the convergence of task modeling for eight target tasks. With $n \le 8k$, the MSE of $\hat{\theta}_n$ on a holdout set converges comparably to the variance of f_t, defined as follows:

$$\text{var}(f_t) = \frac{1}{n}\sum_{i=1}^{n}\left(f_t(S_i) - \mathop{\mathbb{E}}_{f_t}\left[f_t(S_i)\right]\right)^2. \tag{6}$$

We estimate the empirical mean of f_t from ten random seeds. A smaller gap between the empirical risk and the variance of f_t implies the linear model fits the expected f_t values more accurately.

3.2 Optimizing cross-task transfer learning

Optimize cross-task transfer performance requires finding an S that minimizes $f_t(S)$. With a task model, we can select S using the approximated model: $S^\star = \arg\min_S g(S)$. Thus, the minimum can be achieved by choosing all source tasks with a negative coefficient in $\hat{\theta}_n$. Due to the randomness of $\hat{\theta}_n$, we set a threshold γ. To illustrate the intuition, we present a case study in a linear model.

Assume the feature covariate of every task is drawn from an isotropic normal distribution $\mathcal{N}(0, \text{Id}_{p\times p})$. Each task i follows a linear model specified by a parameter vector $\theta^{(i)}$. Given a p dimensional feature vector x, the label of task i satisfies $y = x^\top\theta^{(i)} + \epsilon$, where ϵ is a random variable with mean 0 and variance σ^2. Let a and b be two fixed values so that $b > a > 0$, a task is: (i) *related* if $\theta^{(i)} = \theta^{(t)} + z$, where $z \sim \mathcal{N}(0, a^2\,\text{Id}_{p\times p})$; (ii) *unrelated* if $\theta^{(i)} = \theta^{(t)} + z$, where $z \sim \mathcal{N}(0, b^2\,\text{Id}_{p\times p})$. We prove that with enough samples, Algorithm 1 only selects related source tasks.

Theorem 3.2 (Proof in Appendix D.2)**.** *In the setting described within this subsection, suppose the loss function ℓ is bounded from above by $C > 0$. There are d samples from each task. Let $n \gtrsim C^2k^2/((a^2-b^2)^2))$, $d \gtrsim a^4k^4/(a^2-b^2)^2 + k\log k + p$, and $N \gtrsim p\log p$. There exists a threshold γ such that with probability at least 0.99, (i) $\hat{\theta}_n(i) < \gamma$ for any related task $i \in \{1, 2, \ldots, k\}$; (ii) $\hat{\theta}_n(j) > \gamma$ for any unrelated task $j \in \{1, 2 \ldots, k\}$.*

The analysis uses the fact that $\mathcal{S}$ is a uniform distribution over subsets of a fixed size. We show that under this distribution, the covariance structure in the task indices of $\hat{\theta}_n$ is approximately an identity matrix plus a constant term for every task (cf. Lemma D.3). This covariance structure allows the task model coefficients to separate the related tasks from the unrelated tasks.

4 Empirical Evaluation

Our experiments seek to address the following questions: (i) Does modeling higher-order transfers in task modeling bring some benefit compared to prior works using first-order task affinity? (ii) Does our approach select tasks that transfer positively to the target task? (iii) How well does our approach extend to performance metrics beyond the average prediction loss?

We investigate these questions on various datasets and performance metrics, showing positive results to the three questions. First, we present a detailed analysis of task modeling to validate the benefit of higher-order transfers over first-order transfer metrics and report the computational cost of our approach. Second, we apply our approach to five text classification tasks with noisy supervision sources. Our approach increases the test performance over combining all tasks by **6.4%** and prior methods up to **3.6%**. Third, we apply our approach to optimize group robustness and fairness measures on datasets with multiple subgroups. Our approach consistently improves performance over previous multitask learning approaches. The rest of our experiments can be found in Appendix C. Our work highlights the benefit of modeling higher-order transfers in multitask learning.

References

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. "Convex multi-task feature learning". In: *Machine learning* 73.3 (2008), pp. 243–272.

[2] Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, et al. "Snorkel drybell: A case study in deploying weak supervision at industrial scale". In: *ICMD*. 2019, pp. 362–375.

[3] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. "TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification". In: *EMNLP*. 2020, pp. 1644–1650.

[4] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. "Spectrally-normalized margin bounds for neural networks". In: *NeurIPS* 30 (2017).

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

[6] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. "A theory of learning from different domains". In: *Machine learning* 79.1 (2010), pp. 151–175.

[7] Shai Ben-David and Reba Schuller Borbely. "A notion of task relatedness yielding provable multiple-task learning guarantees". In: *Machine learning* 73.3 (2008), pp. 273–287.

[8] Shai Ben-David and Reba Schuller. "Exploiting task relatedness for multiple task learning". In: *Learning theory and kernel machines*. Springer, 2003, pp. 567–580.

[9] Jonathon Byrd and Zachary Lipton. "What is the effect of importance weighting in deep learning?" In: *ICML*. 2019, pp. 872–881.

[10] Rich Caruana. "Multitask learning". In: *Machine learning* (1997).

[11] Jianhui Chen, Jiayu Zhou, and Jieping Ye. "Integrating low-rank and group-sparse structures for robust multi-task learning". In: *KDD*. 2011, pp. 42–50.

[12] Shuxiao Chen, Koby Crammer, Hangfeng He, Dan Roth, and Weijie J Su. "Weighted Training for Cross-Task Learning". In: *ICLR* (2022).

[13] Koby Crammer, Michael Kearns, and Jennifer Wortman. "Learning from Multiple Sources." In: *Journal of Machine Learning Research* 9.8 (2008).

[14] Lucio M Dery, Yann Dauphin, and David Grangier. "Auxiliary task update decomposition: The good, the bad and the neutral". In: *ICLR* (2021).

[15] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. "Retiring adult: New datasets for fair machine learning". In: *NeurIPS* 34 (2021).

[16] Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. "Few-shot learning via learning the representation, provably". In: *ICML* (2020).

[17] Theodoros Evgeniou and Massimiliano Pontil. "Regularized multi–task learning". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 109–117.

[18] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. "Efficiently identifying task groupings for multi-task learning". In: *NeurIPS* 34 (2021).

[19] Han Guo, Ramakanth Pasunuru, and Mohit Bansal. "AutoSeM: Automatic Task Selection and Mixing in Multi-Task Learning". In: *NAACL* (2019).

[20] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. "Co-teaching: Robust training of deep neural networks with extremely noisy labels". In: *NeurIPS* 31 (2018).

[21] Steve Hanneke and Samory Kpotufe. "On the value of target data in transfer learning". In: *NeurIPS* 32 (2019).

[22] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers". In: *ICASSP*. IEEE. 2013, pp. 7304–7308.

[23] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. "Datamodels: Predicting predictions from training data". In: *ICML* (2022).

[24] Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. "Learning from noisy singly-labeled data". In: *ICLR* (2018).

[25] Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. "Meta-learning for mixed linear regression". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5394–5404.

[26] Abhishek Kumar and Hal Daume III. "Learning task grouping and overlap in multi-task learning". In: *ICML* (2012).

[27] Hunter Lang and Hoifung Poon. "Self-supervised self-supervision by combining deep learning and probabilistic logic". In: *AAAI*. Vol. 35. 6. 2021, pp. 4978–4986.

[28] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. "Representation learning using multi-task deep neural networks for semantic classification and information retrieval". In: *NAACL* (2015).

[29] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. "Multi-Task Deep Neural Networks for Natural Language Understanding". In: *ACL*. 2019, pp. 4487–4496.

[30] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. "Adversarial NLI: A New Benchmark for Natural Language Understanding". In: *ACL*. 2020.

[31] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* (2009).

[32] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning". In: *arXiv preprint arXiv:1711.05225* (2017).

[33] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. "Snorkel: Rapid training data creation with weak supervision". In: *VLDB*. Vol. 11. 3. NIH Public Access. 2017, p. 269.

[34] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. "Training complex models with multi-task weak supervision". In: *AAAI*. Vol. 33. 01. 2019, pp. 4763–4771.

[35] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. "Data programming: Creating large training sets, quickly". In: *NeurIPS* 29 (2016).

[36] Alexander J Ratner, Braden Hancock, and Christopher Ré. "The Role of Massively Multi-Task and Weak Supervision in Software 2.0." In: *CIDR*. 2019.

[37] Sebastian Ruder. "An overview of multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1706.05098* (2017).

[38] Khaled Saab, Sarah M Hooper, Nimit S Sohoni, Jupinder Parmar, Brian Pogatchnik, Sen Wu, Jared A Dunnmon, Hongyang R Zhang, Daniel Rubin, and Christopher Ré. "Observational supervision for medical image classification using gaze data". In: *MICCAI*. Springer. 2021.

[39] Esteban Safranchik, Shiying Luo, and Stephen Bach. "Weakly supervised sequence tagging from noisy rules". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 5570–5578.

[40] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization". In: *ICLR* (2020).

[41] Changho Shin, Winfred Li, Harit Vishwakarma, Nicholas Roberts, and Frederic Sala. "Universalizing weak supervision". In: *ICLR* (2022).

[42] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. "Which tasks should be learned together in multi-task learning?" In: *ICML*. PMLR. 2020, pp. 9120–9132.

[43] Joel A Tropp. "An Introduction to Matrix Concentration Inequalities". In: *Foundations and Trends in Machine Learning* 8.1-2 (2015), pp. 1–230.

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

[45] Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. "Exploring and predicting transferability across NLP tasks". In: *EMNLP* (2020).

[46] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press, 2019.

[47] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. "Superglue: A stickier benchmark for general-purpose language understanding systems". In: *NeurIPS* 32 (2019).

[48] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *ICLR*. 2019, pp. 353–355.

[49] Kishan Wimalawarne, Masashi Sugiyama, and Ryota Tomioka. "Multitask learning meets tensor factorization: task imputation via convex optimization". In: *NeurIPS* 27 (2014).

[50] Sen Wu, Hongyang R Zhang, and Christopher Ré. "Understanding and improving information transfer in multi-task learning". In: *ICLR* (2020).

[51] Fan Yang, Hongyang R Zhang, Sen Wu, Weijie J Su, and Christopher Ré. "Analysis of information transfer from heterogeneous sources via precise high-dimensional asymptotics". In: *arXiv preprint arXiv:2010.11750* (2021).

[52] Yongxin Yang and Timothy Hospedales. "Deep multi-task representation learning: A tensor factorisation approach". In: *ICLR* (2017).

[53] Shipeng Yu, Volker Tresp, and Kai Yu. "Robust multi-task learning with t-processes". In: *ICML*. 2007, pp. 1103–1110.

[54] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. "Gradient surgery for multi-task learning". In: *NeurIPS* 33 (2020), pp. 5824–5836.

[55] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. "Taskonomy: Disentangling task transfer learning". In: *CVPR*. 2018, pp. 3712–3722.

[56] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. "WRENCH: A Comprehensive Benchmark for Weak Supervision". In: *NeurIPS Datasets and Benchmarks Track*. 2021.

[57] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. "Correct-N-Contrast: A Contrastive Approach for Improving Robustness to Spurious Correlations". In: *ICML* (2022).

[58] Yu Zhang and Dit-Yan Yeung. "A convex formulation for learning task relationships in multi-task learning". In: *UAI* (2010).

[59] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. "Facial landmark detection by deep multi-task learning". In: *ECCV*. Springer. 2014, pp. 94–108.

A Related Work

Our work builds on and extends various settings studied in multitask learning (MTL) and transfer learning.

Multitask Learning: We build on existing MTL approaches using an architecture that shares the encoder for all tasks and assigns a separate predictor layer for each task [37]. These approaches have shown great progress in both language [29] and vision domains [32]. Meanwhile, many studies have observed negative results, where MTL performs worse than single task learning [45, 50]. This raises the question of identifying the negative interference and finding the task structures [55]. This is further complicated by the nonlinearity of neural networks. One approach is to measure gradient similarity during training [54, 18]. Since gradients are noisy, directly precomputing multitask predictions is considered [42], which computes first-order task affinity for all pairs of tasks and uses them to approximate higher-order transfer predictions. Our work offers an efficient and principled approach to model higher-order task structures via sampling. Some studies design neural net architectures to encourage information sharing across multiple tasks. Depending on the semantics, layers may be shared or separated across the network [22, 59, 28]. However, this approach requires specifying one architecture for each application. Low-rank tensor factorization can be used to constrain several task model parameters [49, 52]. Complementary to these works, we fix the network encoder and examine the relations of task data structures.

Our approach is inspired by a recent work of Ilyas et al. [23], which predicts the prediction of a set of training samples on another sample drawn from the same unknown distribution. The idea of Ilyas et al. [23] is to use simple surrogate functions such as linear models to approximate a complex function such as the validation loss of a model trained with a subset of samples. Notice that the

multitask learning setting crucially differs from the above work, since we estimate the prediction of combining a subset of source tasks with a target task, then test on the target task.

Task Grouping: Our setting is related to the task grouping problem [26, 42, 18], which is defined as assigning tasks into several groups with each learned in one network for optimizing the overall loss. Different from this problem, we are concerned with a primary target task of interest. This is also studied in several recent works [19, 14, 38, 12], and is loosely related to a robust multitask learning problem [53, 11] studied earlier within linear and kernel models.

Generalization Theory: Some of the earliest works in multitask learning study task relatedness from a learning theoretic perspective [8, 7]. Ben-David et al. [6] introduce a discrepancy notion called H-divergence, which leads to a generalization bound for minimizing the empirical risk of combining source and target training data. Transfer exponents are another measure of discrepancy between two distributions [21], leading to minimax convergence rates. Notice that our sample complexity bound only requires the Rademacher complexity of the encoder. Thus, they can also be combined with spectral norm bounds of deep networks [4], leading to a generalization bound for multitask learning with deep networks. Variants of the linear parametric model for task selection has also been considered in few-shot learning [16] and meta-learning [25]. Extending our approach to these cases is a promising research direction.

Weak Supervision: We draw motivation from recent work which models and integrates weak supervision for rapidly training deep models [33, 20, 24, 2, 39, 41]. In particular, our work is inspired by previous multitask weak supervision approaches [34, 36]. These approaches, however, do not handle the negative interference between multiple labeling functions. Our approach is related to probabilistic models of the noisy sources [27], but differ in that each label is treated as a source task rather than aggregated together.

B Figures for Illustrating Task Modeling and Negative Transfer

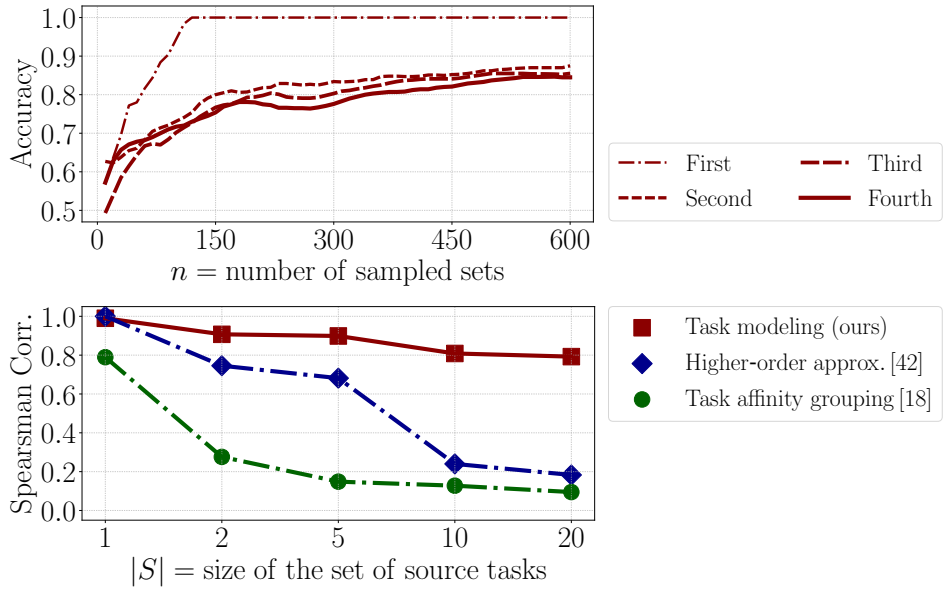


Figure 2: Will combining a set of source tasks S with a primary target of interest help or hurt? We approach this question by sampling source tasks and estimating the loss of the target. This leads to a new way to efficiently approximate higher-order task structures, called *task modeling* in this work. **Top**: Task modeling answers the above question with 80% accuracy with up to four tasks in S. **Bottom:** Compared with existing higher-order approximations that average first-order task affinity [42, 18], task modeling consistently captures higher-order predictions more accurately.

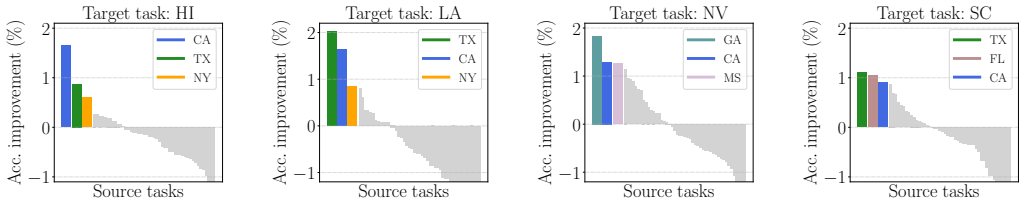


Figure 3: Mixed outcomes are commonly observed due to negative transfer in multitask learning. In some cases, combining a source task with a target task helps; in other cases, it hurts. $\boldsymbol{x}$**-axis**: Each bar represents one source task, for a total of fifty of them. $\boldsymbol{y}$**-axis**: Difference between test accuracy of combining a source and target task and single task learning.

C Experiments

C.1 Experimental setup

Datasets. First, we consider text classification tasks with noisy supervision sources from a weak supervision dataset [56]. Each weak supervision source generates noisy labels for a subset of training samples. We view noisy sources as source tasks. The task with true labels can be viewed as the target task and is not available during training. A validation set of true labels is used for task selection and parameter tuning. Table 1 describes the statistics of five text classification tasks along with the number of source tasks.

Table 1: Dataset statistics of five text classification tasks.

Tasks	Youtube	TREC	CDR	Chemprot	Semeval
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

Second, we consider binary classification tasks which involve multiple groups of subpopulations. We consider the Folktables dataset derived from the US census [15], in particular an income prediction task spanning all states. In this task, each record indicates whether an individual's income is above $50,000 or not, using ten tabular features including education level, age, sex, etc. We view each state as one task. We use the racial attribute of an individual to split each state dataset into nine groups that exhibit group shifts. We evaluate the robustness of a predictor with its worst-group accuracy, defined as the predictor's accuracy in the worst performing group among all nine groups. Table 2 describes the statistics of six states/target tasks. In each case, there will be fifty source tasks.

Table 2: Dataset statistics of six binary classification tasks.

Tasks	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	67	75	58	52	61	203

Baselines. We first compare our approach with training on all tasks using hard parameter sharing. Then, we consider approaches that model first-order task affinity, including approximating higher order task relations using two-task network performance [42], estimating task relations using cosine similarity between task gradients [18] and computing lookahead losses with task gradients [18]. Additionally, we also consider approaches that alter the optimization using pairwise task relations, including auxiliary task gradient update decomposition [14] and target-aware weighted training [12].

For tasks with weak supervision sources, we incorporate previous weak supervision methods to aggregate noisy labels and train an end model on the labels inferred by the methods. The methods include Majority Vote, Data Programming [35], and MeTaL [34].

For the binary prediction tasks, we also consider empirical risk minimization and approaches that aim to improve the worst-group performance, including importance weighting [9], group distribu-

Table 3: Test performance on five text classification tasks with multiple noisy supervisions, averaged over five random seeds.

Method/Dataset (Metrics)	Youtube (Acc.)	TREC (Acc.)	CDR (F1)	Chemprot (Acc.)	Semeval (Acc.)	Avg. Rank
Majority Vote	95.36±1.71	66.56±2.31	58.89±0.50	57.32±0.98	85.03±0.83	4.6
Data Programming [35]	93.84±1.61	68.64±3.57	58.48±0.73	57.00±1.20	83.93±0.83	6.6
MeTaL [34]	92.32±1.44	58.28±1.95	58.48±0.90	56.17±0.66	71.74±0.57	8.4
Hard parameter sharing	94.72±0.85	64.10±0.50	58.20±0.55	53.43±0.53	89.00±1.06	7.8
High-order approx. [42]	94.93±1.80	74.67±4.66	59.76±0.97	45.57±0.41	79.94±4.42	6.2
Gradient similarity [18]	95.33±0.68	78.25±3.71	59.21±0.80	53.67±1.89	89.89±2.17	4.0
Task affinity grouping [18]	95.20±0.65	77.50±3.62	59.31±0.15	53.67±2.74	89.06±1.47	4.2
Weighted training [12]	94.53±1.05	72.40±2.36	59.85±0.30	53.76±2.96	86.83±1.78	5.0
Gradient decomposition [14]	95.28±0.16	65.80±1.81	58.81±0.36	54.76±0.67	78.57±0.13	7.0
Task modeling (Alg. 1)	**97.47±0.82**	**81.80±1.14**	**61.22±0.39**	**57.54±0.55**	**93.50±0.24**	**1.0**

tionally robust optimization [40], and supervised contrastive learning [57]. More details concerning hyperparameters are described in Section E.1.

Implementation. We use BERT-Base on the text classification tasks. For the income prediction task from the Folktables dataset, we use a two-layer perceptron model with a hidden size 32. We adopt the hard parameter sharing architecture for conducting multitask learning on the datasets.

To estimate a task model, we collect n task subsets along with the multitask training result of each subset. We consider a uniform sampling distribution over the task subsets of a constant size. For each income prediction task, we obtain $n = 400$ results on $|S| = 5$ source tasks. We also construct as a holdout set of size 100. For the text classification datasets, since the number of source tasks (c.f. Table 1) varies among datasets. We obtain $n = \{50, 200, 200, 400, 800\}$ results with each on $|S| = \{3, 5, 5, 10, 15\}$ source tasks from Youtube, Chemprot, CDR, TREC, Semeval datasets, respectively. We set $f_t(S)$ as the negative classification margin — the difference between the logit of the correct class and the highest incorrect logit.

C.2 Task modeling results

How much does modeling higher-order structures gain? We validate the benefit of using higher-order task affinity over first-order and second-order task affinities.

For first-order task affinity, we select source tasks by training every source task with the target task, following HOA [42]. The results in Table 3 and 4 confirm that by sampling subsets of S with size up to five, task modeling outperforms HOA by **3.9%** averaged over eleven tasks.

For second-order task affinity, we conduct an exhaustive search over the space of source tasks to show that going beyond first-order task affinity is necessary. We search through all possible choices of $|S| = 2$ on one binary classification task, which amounts to training 1225 models, each with two source tasks and one target task. The results show that our approach outperforms the best S by **1.21%** accuracy. See Table 5 for the results.

How long does constructing task models take? We detail the computation costs of our approach. As shown in Section 3.1, for each target task, using $n \leq 8k$ subsets suffices for task models to converge. We validate similar convergence for other tasks in Figure 5 of Appendix E.2. We also report the GPU hours of collecting training results for each target task in Appendix E.3. Across all eleven cases, constructing task models until convergence takes at most 85.9 GPU hours, evaluated on an NVIDIA TITAN RTX instance.

Next, we compare the computation costs of task modeling and prior methods. We use one binary classification task as an example. To only precompute first-order task affinity, our approach takes the same amount of time as HOA, which takes 1.24 GPU hours to train on all source-target pairs, and comparable time to TAG, which takes 0.87 GPU hours.

Notice that both HOA and TAG are not designed to predict higher-order transfers. Thus, we compare our approach with exhaustive search for $|S| > 2$. Recall that our approach requires sampling $n = O(k\alpha^4 \log^2 k)$ in theory. In practice, we notice that $n = 8k$ suffices for training task models until convergence in all of our use cases. We also notice that the required n decreases as $|S|$ increases, as shown in Figure 4 of Section C.2. As a result, our approach takes the same amount of time for

Table 4: Worst-group test accuracy on six binary classification tasks with tabular features, averaged over ten random seeds.

Method/Dataset	HI	KS	LA	NJ	NV	Avg. Rank
Empirical risk minimization	74.46±0.48	73.73±1.19	72.39±1.96	76.34±0.64	72.89±1.42	9.7
Importance weighting [9]	74.53±0.81	72.84±1.74	74.82±0.94	76.43±0.50	71.25±1.73	7.8
Correct-N-Contrast [57]	74.37±0.27	75.52±1.19	74.25±0.15	77.60±0.10	73.22±0.40	5.0
Group robust optimization [40]	74.56±0.58	75.50±0.59	74.90±0.38	76.95±0.20	73.06±0.66	5.5
Hard parameter sharing	73.63±0.46	75.22±0.73	73.24±1.01	77.28±0.25	73.22±1.12	8.0
High-order approx. [42]	74.67±0.32	75.22±1.48	73.69±0.86	77.49±0.25	73.88±0.66	3.6
Gradient similarity [18]	74.53±0.52	75.22±2.02	73.66±1.22	77.44±0.38	74.38±0.91	3.8
Task affinity grouping [18]	74.48±0.41	75.97±1.18	73.24±1.01	77.41±0.48	74.05±0.84	5.1
Weighted training [12]	73.53±0.44	75.14±1.39	73.51±1.38	76.47±1.31	72.89±0.81	7.8
Gradient decomposition [14]	73.20±0.57	72.24±1.19	73.51±0.66	76.38±0.69	73.71±0.84	8.0
Task modeling (Alg. 1)	**75.47±0.73**	**76.96±0.69**	**75.62±0.11**	**78.17±0.36**	**75.21±0.52**	**1.0**

Table 5: Comparison of different loss functions and exhaustive search over all subsets of at most two source tasks.

	HI	KS	LA	NJ	NV	SC
Exhaustive search of $\|S\| \leq 2$	75.10±0.37	77.03±0.76	73.60±1.02	77.40±0.24	73.21±1.10	77.16±0.21
f_t 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_t 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_t 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

different sizes of S up to 20, which is less than 52 GPU hours. By contrast, the runtime of the exhaustive search increases exponentially as the size of S grows.

C.3 Task selection results

Cross-task transfer learning. Our result in Section 3.2 shows that task modeling provides signals to identify beneficial source tasks. We validate the result with the text classification tasks with several noisy supervision sources. We apply Algorithm 1 to select the noisy sources and evaluate the test performance on the classification task with true labels. Table 3 shows the results.

Compared with hard parameter sharing which trains all tasks in the same network, our algorithm improves the test performance by **6.4%** on average. This shows that our algorithm excludes tasks with negative interference, thus performing better than training on all tasks. Compared with existing multitask learning approaches that either reweight the source tasks [14, 12] or select with first-order task affinity [42, 18], our algorithm increases up to **3.6%** accuracy.

Group robustness and fairness metrics. Next, we show that task modeling also captures task affinity with various performance metrics of the primary target task. We consider the binary classification tasks with multiple subpopulation groups. We apply Algorithm 1 to select source tasks as an augmentation of the target classification task. Table 4 presents the comparison.

Compared with single task learning, including ERM, GroupDRO, and CNC, we find that task modeling improves the worst-group accuracy by 1.17% on average, confirming the benefit of data augmentation. Compared with existing multitask learning approaches, our approach shows a favorable gain of up to **1.9%** absolute accuracy. On two fairness measures, our algorithm outperforms all methods by **1.8%** on average. Due to the space limit, this result is described in Appendix E.3. Hence, we conclude that task modeling is a general approach that approximates multitask predictions for various performance measures.

C.4 Ablation study of model parameters

We ablate the parameters used in our algorithm, providing further insights into its working.

Subset size $|S|$: 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 $|S| \in \{2, 5, 10, 20\}$. To control the computation budget the same, we scale the number of subsets n according to $|S|$. We train $n = 800, 400, 200, 100$ models with $|S| = 2, 5, 10, 20$, respectively. We observe similar convergence results as in Figure 1. Among them, $|S| = 5$ yields a highest Spearman

correlation of 0.89 between $f_t(\cdot)$ and $g(\cdot)$. The reason why higher values of $|S|$ do not help is that the number of beneficial tasks is limited in this setting.

Number of samples n: Next, we explore how n affects the estimated task models. 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 (out of 10) same source tasks as using 400. Increasing n to 200 further identifies 9 (out of 10) same source tasks as using 400.

Loss function ℓ: 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 5. 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.

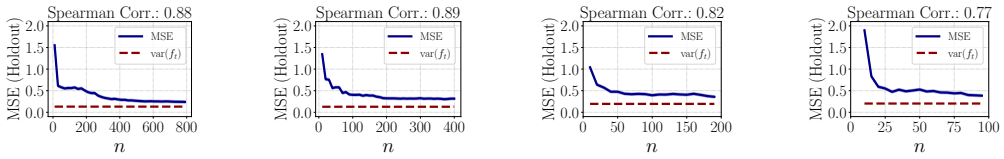


Figure 4: Ablation study of choosing different subset sizes on the same target task. From left to right: $|S| = 2, 5, 10, 20$.

D Proofs of Theorems 3.1 and 3.2

Notations: Let $\mathrm{Id}_{p\times p}$ denote the identity matrix with dimension p by p. Let $\|\cdot\|$ denote the Euclidean norm of a vector. For two functions $f(n)$ and $g(n)$, we write $g(n) \lesssim f(n)$ if there exists a fixed value c that does not grow with n such that $g(n) \leq c \cdot f(n)$ when n is large enough. Let $\mathcal{F} = \{\ell(\psi_t(\phi(x)), y) \,|\, \forall\, \psi_t, \phi\}$ be a function class of the target task. Let $\mathcal{R}_N(\mathcal{F})$ be the Rademacher complexity of $\mathcal{F}$ on N samples of the target task distribution.

We follow the convention of big-O notations in the proof. Given two functions $f(n)$ and $g(n)$, we use $f(n) = O(g(n))$ to indicate that $f(n) \leq C \cdot g(n)$ for some fixed constant C when n is large enough. The notation $f(n) \lesssim g(n)$ indicates that $f(n) = O(g(n))$. 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 X, denote the spectral norm (or the largest singular value) of X as $\|X\|_2$. Denote the Frobenius norm of X as $\|X\|_F$. For a vector v, denote the Euclidean norm of v as $\|v\|$.

Let $\tilde{\mathcal{D}}_t = \{x_1^{(t)}, x_2^{(t)}, \ldots, x_N^{(t)}\}$ be N i.i.d. samples of $\mathcal{D}_t$. Let $\sigma_1, \sigma_2, \ldots, \sigma_N$ be independent Rademacher random variables. Denote the Rademacher complexity of task t with N samples from $\mathcal{D}_t$ as:

$$\mathcal{R}_N(\mathcal{F}) = \mathop{\mathbb{E}}_{\tilde{\mathcal{D}}_t, \sigma}\left[\sup_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \sigma_i f\left(x_i^{(t)}\right)\right]. \tag{7}$$

D.1 Proof of Theorem 3.1

We prove the convergence rate of task modeling as a function of n—the number of subsets that one needs to sample in order to learn a task model, and N—the size of the target task's validation set used to evaluate f_t. Let $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{|\mathcal{S}|\times k}$ be a zero-one matrix including $\mathbb{1}_T$ as its row vectors, for all $T \in \mathcal{S}$. Let $\boldsymbol{f}$ be an $|\mathcal{S}|$ dimensional vector such that $\boldsymbol{f}_T = f_t(T)$ for every $T \in \mathcal{S}$. Let $\mathcal{I}_n \in \mathbb{R}^{n\times k}$ be a zero-one matrix including $\mathbb{1}_{S_1}, \ldots, \mathbb{1}_{S_n}$ as its row vectors. Let $\hat{f}$ be an n dimensional vector such that $\hat{f}_i = f_t(S_i)$.

Recall from equation (29) that the minimizer of the empirical loss $\hat{\mathcal{L}}_n(\theta)$ is equal to:[1]

$$\hat{\theta}_n = \left(\mathcal{I}_n^\top \mathcal{I}_n\right)^{-1} v_n,$$

where the i-th entry of v_n is defined as

$$\sum_{1 \le j \le n:\, i \in S_j} f_t(S_j).$$

In a similar vein, denote the minimizer of the population loss $\mathcal{L}(\theta)$ as

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

Lemma D.1. *In the setting of Theorem 3.1, let $\hat{\theta}_{|\mathcal{S}|}$ be defined as $\left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{S}|}\right)^{-1} \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{S}|}$. Conditional on f_t and $\hat{\mathcal{D}}_1, \ldots, \hat{\mathcal{D}}_k$, with probability $1 - 2\delta$ over the randomness of the sampled subsets $S_1, S_2, \ldots, S_n$, for any $\delta \ge 0$, $\hat{\theta}_n$ converges to $\hat{\theta}_{|\mathcal{S}|}$ in probability:*

$$\left\|\hat{\theta}_n - \hat{\theta}_{|\mathcal{S}|}\right\| \le Z\sqrt{\frac{k}{n}}. \tag{8}$$

where $Z = 4C\alpha^2 \log(2k\delta^{-1}) + (1 - \alpha/k)^{-3} C\alpha\delta^{-1/2}$.

Proof. We will use the triangle inequality to attribute the error between $\hat{\theta}_n$ and $\hat{\theta}_{|\mathcal{S}|}$ to two parts.

$$\begin{aligned}
\left\|\hat{\theta}_n - \hat{\theta}_{|\mathcal{S}|}\right\| &= \left\| \left(\left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} - \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{S}|}\right)^{-1} \right) \frac{v_n}{n} + \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{S}|}\right)^{-1} \left(\frac{v_n}{n} - \frac{\boldsymbol{\mathcal{I}}^\top \mathbb{E}\left[\boldsymbol{f}\right]}{|\mathcal{S}|} \right) \right\| \\
&\le \left\| \left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} - \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{S}|}\right)^{-1} \right\|_2 \cdot \left\| \frac{v_n}{n} \right\| \qquad (9) \\
&\quad + \left\| \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{S}|}\right)^{-1} \right\|_2 \cdot \left\| \frac{v_n}{n} - \frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{S}|} \right\|. \qquad (10)
\end{aligned}$$

We compare the sampled score vector $\frac{v_n}{n}$ and the population score vector $\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{S}|}$. 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 $\frac{v_n}{n}$ and the i-th coordinate of $\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{f}}{|\mathcal{S}|}$:

$$\mathcal{E}_i = \frac{1}{n} \sum_{1 \le j \le n:\, i \in S_j} f_t(S_j) - \frac{1}{|\mathcal{S}|} \sum_{T \in \mathcal{S}:\, i \in T} f_t(T). \tag{11}$$

Notice that the sampling of $S_1, S_2, \ldots, S_n$ is independent of the randomness in $f_{\mathcal{A}}$. Therefore, we have that the expectation of $\mathcal{E}_i$ is zero: $\mathbb{E}\left[\mathcal{E}_i\right] = 0$. Next, we apply the Chebyshev's inequality to analyze the deviation of $\mathcal{E}_i$ from its expectation. We consider the variance of $\mathcal{E}_i$, which is the expectation of $\mathcal{E}_i^2$:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{E}_i^2\right] &= \mathbb{E}\left[\left(\frac{1}{n} \sum_{1 \le j \le n : i \in S_j} f_t(S_j) - \frac{1}{|\mathcal{S}|} \sum_{T \in \mathcal{S} : i \in T} f_t(T) \right)^2 \right] \qquad (12) \\
&= \mathbb{E}\left[\left(\sum_{1 \le j \le n : i \in S_j} f_t(S_j) \right)^2 - \frac{2}{n\,|\mathcal{S}|} \sum_{1 \le j \le n : i \in S_j} f_t(S_j) \sum_{T \in \mathcal{S} : i \in T} f_t(T) + \frac{1}{|\mathcal{S}|^2} \Big(\sum_{T \in \mathcal{S} : i \in T} f_t(T) \Big)^2 \right]
\end{aligned}$$

Notice that for any $T \in \mathcal{S}$ such that $i \in T$, the probability that T is sampled in a size n (training) set is equal to

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

[1] With a similar analysis one could also prove the convergence from $\hat{\mathcal{L}}_n(\cdot)$ to $\mathcal{L}(\cdot)$ with the minimizer of the ridge regression, which includes λ times an identity matrix in the inverted sample covariance of $\hat{\theta}_n$.

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

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

Thus, by taking the expectation over the randomness of the sampled subsets in equation (12) conditional on f_t, we get that

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

since we have assumed that the loss function $\ell(\cdot, \cdot)$ is bounded from above by an absolute constant C and f_t is the average loss. 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 $a > 0$,

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

Therefore, with probability $1 - \delta$, for any $\delta > 0$, we have that

$$\left\| \frac{v_n}{n} - \frac{\boldsymbol{\mathcal{I}}^\top \mathbb{E}\left[\boldsymbol{f}\right]}{|\mathcal{S}|} \right\| \leq C\delta^{-1/2}\sqrt{\frac{k}{n}}. \tag{13}$$

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 $\mathcal{S}$. Denote by

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

By the Sherman-Morrison formula (for matrix inversion), we get

$$\begin{aligned}
\left\| \left(\frac{\mathcal{I}_n^\top \mathcal{I}_n}{n}\right)^{-1} - \left(\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{S}|}\right)^{-1} \right\|_2 &= \left\|(E + A)^{-1} - A^{-1}\right\|_2 \\
&= A^{-1}\Big(AE^{-1} + \mathrm{Id}_{k\times k}\Big)^{-1} \\
&= A^{-1}E\Big(A + E\Big)^{-1} \\
&\leq \big(\lambda_{\min}(A)\big)^{-1} \cdot \|E\|_2 \cdot \big(\lambda_{\min}(A + E)^{-1}\big) \\
&\leq \frac{\|E\|_2}{\lambda_{\min}(A)(\lambda_{\min}(A) - \|E\|_2)}.
\end{aligned} \tag{14}$$

We now use the matrix Bernstein inequality (cf. Theorem 6.1.1 in Tropp [43]) to deal with the spectral norm of E. Let

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

Let $\mathcal{D}$ denote the uniform distribution over $\mathcal{S}$. In expectation over $\mathcal{D}$, 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 ones. Therefore, for all $t \geq 0$,

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

This implies (with some calculation) that for any $\delta \geq 0$, with probability at least $1-\delta$,

$$\|E\|_2 \leq \frac{4\alpha \cdot \log\left(2k\delta^{-1}\right)}{\sqrt{n}}. \tag{15}$$

By applying equation (13) into equation (9) and equation (15) into equation (10), we have shown that with probability at least $1-2\delta$, for any $\delta \geq 0$,

$$\left\|\hat{\theta}_n - \hat{\theta}_{|\mathcal{S}|}\right\| \leq \left\|\frac{v_n}{n}\right\|_2 \cdot \frac{4\alpha \cdot \log\left(2k\delta^{-1}\right)}{\sqrt{n}} + \frac{1}{\left(\lambda_{\min}(A)\right)^2\left(\lambda_{\min}(A) - \|E\|_2\right)} \cdot C\delta^{-1/2}\sqrt{\frac{k}{n}}. \tag{16}$$

For the first part, 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 loss $\ell(\cdot,\cdot)$ is bounded from above by an absolute constant C. Thus,

$$\left\|\frac{v_n}{n}\right\| \leq \frac{1}{n}\sqrt{C^2 \cdot \sum_{i=1}^{k} z_i^2} \leq \frac{C}{n}\Big(\sum_{i=1}^{k} z_i\Big) = C\alpha. \tag{17}$$

Regarding the minimum eigenvalue of A, notice that the diagonal entry of $\frac{\boldsymbol{\mathcal{I}}^\top \boldsymbol{\mathcal{I}}}{|\mathcal{S}|}$ is equal to $\binom{k-1}{\alpha-1}$. The off diagonal entries of this matrix is equal to $\binom{k-2}{\alpha-2}$. Thus,

$$\lambda_{\min}(A) \geq 1 - \frac{\binom{k-2}{\alpha-2}}{\binom{k-1}{\alpha-1}} = 1 - \frac{\alpha-1}{k-1} \geq 1 - \frac{\alpha}{k}. \tag{18}$$

Applying equations (17) and (18) back into equation (16), we conclude that with probability at least $1-2\delta$, $\hat{\theta}_n$ deviates from $\hat{\theta}_{|\mathcal{S}|}$ by a rate of $\sqrt{k/n}$:

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

□

Next, we show the uniform convergence of $\hat{\theta}_{|\mathcal{S}|}$. The key observation is that the size of $\mathcal{S}$ only depends on the number of tasks k. Therefore, one can afford to apply a union bound over $\mathcal{S}$.

Lemma D.2. *In the setting of Theorem 3.1, for any $\delta > 0$, with probability at least $1-\delta$, the deviation between $\hat{\theta}_{|\mathcal{S}|}$ and $\theta^\star$ satisfies:*

$$\left\|\hat{\theta}_{|\mathcal{S}|} - \theta^\star\right\| \leq (1-\alpha/k)^{-1}\frac{\mathcal{R}_N(\mathcal{F})}{2} + (1-\alpha/k)^{-1}\sqrt{\frac{\alpha\log\left(k/\delta\right)}{2N}}. \tag{19}$$

Proof. Based on the definitions of $\hat{\theta}_{|\mathcal{S}|}$ and $\theta^\star$, we analyze their difference as follows:

$$\begin{aligned}
\left\|\hat{\theta}_{|\mathcal{S}|} - \theta^\star\right\| &= \left\|\left(\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}\right)^{-1}\boldsymbol{\mathcal{I}}^\top\left(\boldsymbol{f} - \mathbb{E}\left[\boldsymbol{f}\right]\right)\right\| \\
&\leq \left\|\sqrt{|\mathcal{S}|}\left(\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}\right)^{-1}\boldsymbol{\mathcal{I}}^\top\right\|_2 \cdot \frac{\|\boldsymbol{f} - \mathbb{E}\left[\boldsymbol{f}\right]\|}{\sqrt{|\mathcal{S}|}} \\
&= \left(\lambda_{\min}(\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}})\right)^{-1/2} \cdot \frac{\|\boldsymbol{f} - \mathbb{E}\left[\boldsymbol{f}\right]\|}{\sqrt{|\mathcal{S}|}} \\
&\leq (1-\alpha)^{-1}\frac{\|\boldsymbol{f} - \mathbb{E}\left[\boldsymbol{f}\right]\|}{\sqrt{|\mathcal{S}|}}. \qquad \text{(by equation (18))}
\end{aligned} \tag{20}$$

For each subset $T \in \mathcal{S}$, we will apply a Rademacher complexity based generalization bound to analyze the generalization error $\boldsymbol{f}_T - \mathbb{E}\left[\boldsymbol{f}_T\right]$. Recall the Rademacher complexity of $\mathcal{F}$ with N_t samples from $\mathcal{D}_t$ is defined as

$$\mathcal{R}_N(\mathcal{F}) = \mathop{\mathbb{E}}_{\tilde{\mathcal{D}}_t,\sigma}\left[\sup_{f\in\mathcal{F}}\frac{1}{N}\sum_{j=1}^{N}\sigma_j f(x_j^{(t)})\right].$$

By Theorem 5 of Bartlett and Mendelson [5], with probability at least $1-\delta$, we can get:

$$\boldsymbol{f}_T \leq \mathbb{E}\left[\boldsymbol{f}_T\right] + \frac{\mathcal{R}_N(\mathcal{F})}{2} + \sqrt{\frac{\log\left(1/\delta\right)}{2N}}. \tag{21}$$

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

$$\boldsymbol{f}_T \leq \mathbb{E}\left[\boldsymbol{f}_T\right] + \frac{\mathcal{R}_N(\mathcal{F})}{2} + \sqrt{\frac{\alpha \log\left(k/\delta\right)}{2N}}, \text{ for all } T \in \mathcal{S}, \tag{22}$$

since $\log\left(\binom{k}{\alpha}/\delta\right) \leq \alpha \log(k\delta^{-1})$. Let $z = \sqrt{\alpha \log\left(k\delta^{-1}\right)/(2N)}$. Applying equation (22) back into equation (20), we have shown

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

Thus, the proof is complete. □

Based on the result from Lemma D.1 and Lemma D.2, we are now ready to prove our main result.

Proof of Theorem 3.1. Notice that equation (4) follows by combining equation (8) (from Lemma D.1) and equation (19) (from Lemma D.2), together with the condition that $\alpha \leq 1/2$.

To analyze the generalization error of $\hat{\mathcal{L}}_n(\hat{\theta}_n)$, we first expand it out as

$$\begin{aligned}\hat{\mathcal{L}}_n(\hat{\theta}_n) &= \left\|\mathcal{I}_n\hat{\theta}_n - \hat{f}\right\|^2 \\ &= \frac{1}{n}\left\|\mathcal{I}_n\hat{\theta}_n - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] + \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] - \hat{f}\right\|^2 \\ &= \frac{1}{n}\left\|\mathcal{I}_n\hat{\theta}_n - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\|^2 + \frac{1}{n}\langle \mathcal{I}_n\hat{\theta}_n - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right], \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] - \hat{f}\rangle + \frac{1}{n}\left\|\mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right] - \hat{f}\right\|^2.\end{aligned} \tag{23}$$

Based on Lemma D.1, the distance between $\hat{\theta}_n$ and $\theta^\star$ is at the order of $O\left(n^{-1/2}\right)$ with high probability. We will use this result to deal with the first term in equation (23):

$$\begin{aligned}&\frac{1}{n}\left\|\mathcal{I}_n\hat{\theta}_n - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\|^2 - \frac{1}{n}\left\|\mathcal{I}_n\theta^\star - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\| && (24)\\ =&\left|\frac{1}{n}\langle \mathcal{I}_n^\top\mathcal{I}_n, \hat{\theta}_n(\hat{\theta}_n)^\top - \theta^\star(\theta^\star)^\top\rangle - \frac{2}{n}\langle \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right], \hat{\theta}_n - \theta^\star\rangle\right| \\ \leq&\left\|\frac{1}{n}\mathcal{I}_n^\top\mathcal{I}_n\right\|_2 \cdot \left\|\theta^\star(\theta^\star)^\top - \hat{\theta}_n(\hat{\theta}_n)^\top\right\|_F + \frac{2}{n}\left\|\mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\| \cdot \left\|\theta^\star - \hat{\theta}_n\right\| && \text{(by triangle inequality)} \\ \leq&\alpha\left\|\theta^\star(\theta^\star)^\top - \hat{\theta}_n(\hat{\theta}_n)^\top\right\|_F + 2C\alpha \cdot e_1, && \text{(by equations (17) and (4))}\end{aligned}$$

where e_1 denotes the right hand side of equation (4). In the last step, we used 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. Next,

$$\begin{aligned}\left\|\theta^\star(\theta^\star)^\top - \hat{\theta}_n(\hat{\theta}_n)^\top\right\|_F &= \left\|\theta^\star(\theta^\star - \hat{\theta}_n)^\top + (\theta^\star - \hat{\theta}_n)(\hat{\theta}_n)^\top\right\|_F \\ &\leq \left\|\theta^\star(\theta^\star - \hat{\theta}_n)^\top\right\|_F + \left\|(\theta^\star - \hat{\theta}_n)(\hat{\theta}_n)^\top\right\|_F && \text{(by triangle inequality)} \\ &\leq \left(\|\theta^\star\| + \left\|\hat{\theta}_n\right\|\right)e_1. && \text{(by equation (4))}\end{aligned}$$

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,

$$\begin{aligned}\|\theta^\star\| &= \left\|(\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}})^{-1}\boldsymbol{\mathcal{I}}^\top \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}]\right\| \\ &\le \left\|\left(\frac{\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}}{|\mathcal{S}|}\right)^{-1}\right\|_2 \cdot \left\|\frac{\boldsymbol{\mathcal{I}}^\top \mathbb{E}_{\boldsymbol{f}}[\boldsymbol{f}]}{|\mathcal{S}|}\right\| \\ &\le (1-\alpha/k)^{-1}\cdot C\sqrt{\alpha} \qquad \text{(by equation (18) and } \ell(\cdot,\cdot)\le C)\end{aligned}$$

Notice that the spectral norm between $\boldsymbol{\mathcal{I}}^\top\boldsymbol{\mathcal{I}}/|\mathcal{S}|$ and $\mathcal{I}_n^\top\mathcal{I}_n/n$ is bounded by equation (15). Thus, with similar steps as above, we can show

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

To wrap up our analysis above, we have shown that equation (24) 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 (23). Let e_2 be the deviation error indicated in equation (22) Thus, every entry of $\hat{f} - \mathbb{E}_{\hat{f}}\left[\hat{f}\right]$ is at most e_2. Besides, each entry of $\mathcal{I}_n\hat{\theta}_n - \mathbb{E}_{\hat{f}}\left[\hat{f}\right]$ is less than

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

Thus, the second term in equation (23) is less than

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

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

$$\begin{aligned}\mathcal{L}(\theta^\star) &= \mathop{\mathbb{E}}_{\boldsymbol{f}}\left[\frac{1}{|\mathcal{S}|}\|\boldsymbol{\mathcal{I}}\theta^\star - \boldsymbol{f}\|^2\right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{f}}\left[\frac{1}{|\mathcal{S}|}\left\|\boldsymbol{\mathcal{I}}\theta^\star - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}] + \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}] - \boldsymbol{f}\right\|^2\right] \\ &= \frac{1}{|\mathcal{S}|}\left\|\boldsymbol{\mathcal{I}}\theta^\star - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}]\right\|^2 + \frac{1}{|\mathcal{S}|}\left(\mathop{\mathbb{E}}_{\boldsymbol{f}}\left[\left\|\boldsymbol{f} - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}]\right\|^2\right]\right) \qquad (25)\end{aligned}$$

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

$$\left|\frac{1}{n}\left\|\mathcal{I}_n\theta^\star - \mathop{\mathbb{E}}_{\hat{f}}\left[\hat{f}\right]\right\| - \frac{1}{|\mathcal{S}|}\left\|\boldsymbol{\mathcal{I}}\theta^\star - \mathop{\mathbb{E}}_{\boldsymbol{f}}[\boldsymbol{f}]\right\|\right| \le \left((1-\alpha/k)^{-1}\sqrt{\alpha} + C\right)\sqrt{\frac{\log\left(\delta^{-1}\right)}{n}}. \qquad (26)$$

Lastly, we consider the third term in equation (23), compared with the second term in equation (25). For every $T\in\mathcal{S}$, let $e_T = \boldsymbol{f}_T - \mathbb{E}[\boldsymbol{f}_T]$. By equation (22), we know that e_T is of order $O(N^{-1/2})$, for every $T\in\mathcal{S}$. Therefore

$$\left|\frac{1}{n}\sum_{i=1}^n e_{S_i}^2\right| \le \left(\frac{\mathcal{R}_N(\mathcal{F})}{2} + \sqrt{\frac{\alpha\log(k/\delta)}{2N}}\right)^2, \qquad (27)$$

which is of order $O(N^{-1})$. Similarly, the same holds for variance of $\boldsymbol{f}$ in the second term of equation (25). Comparing equations (26) and (23), we have shown that

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

The above follows by incorporating the definitions of the error terms. Thus, we have completed the proof of equation (5). The proof is now finished. □

D.2 Proof of Theorem 3.2

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 covariate matrices and label vectors as $(X_1, Y_1), (X_2, Y_2), \ldots, (X_\alpha, Y_\alpha)$. With hard parameter sharing [51], we minimize

$$\ell(B) = \sum_{i=1}^{\alpha} \|X_i B - Y_i\|^2 . \tag{28}$$

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, the loss of using B on the validation set of the target task is equal to

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

By solving equation (2), the estimated task model $\hat{\theta}_n$ is equal to

$$\hat{\theta}_n = \left(\mathcal{I}_n^\top \mathcal{I}_n\right)^{-1} v_n, \tag{29}$$

where $v_n = \mathcal{I}_n^\top \hat{f} \in \mathbb{R}^k$ is a vector that satisfies:

$$v_n(i) = \sum_{j:\, i\in\mathcal{S}_j} f_t(S_j), \text{ for any } 1 \le i \le k.$$

First we show that $\hat{\theta}_n$ is approximately a scaling of the vector v_n. The key observation is that $\mathcal{I}_n^\top \mathcal{I}_n$ is approximately an identity matrix plus a constant shift for every task.

Lemma D.3. *In the setting of Theorem 3.2, with probability $1-\delta$, for any $\delta > 0$, the following holds:*

$$\left|\frac{\hat{\theta}_n(i) - \hat{\theta}_n(j)}{n} - \frac{k}{\alpha}\cdot\frac{v_n(i) - v_n(j)}{n}\right| \lesssim \frac{\log(\delta^{-1}k)}{\sqrt{n}}, \text{ for any } 1 \le i < j \le k. \tag{30}$$

Proof. 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

$$f_t(S) = \left\|\left(\sum_{i=1}^{\alpha} X_i^\top X_i\right)^{-1} \sum_{i=1}^{\alpha} X_i^\top \epsilon^{(i)}\right\|^2 . \tag{31}$$

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 $Z = \mathcal{I}_n^\top \mathcal{I}_n / n$. The expectation of Z over the randomness of $\mathcal{I}_n$ satisfies

$$\mathbb{E}[Z] = \frac{\alpha}{k}\,\mathrm{Id}_{k\times k} + \frac{\alpha(\alpha-1)}{k(k-1)} ee^\top,$$

where $e \in \mathbb{R}^k$ is the all ones vector. Thus, by the Woodbury matrix identity,

$$\mathop{\mathbb{E}}_{Z}[Z]^{-1} = \frac{k}{\alpha}\left(\mathrm{Id}_{k\times k} - \frac{k(\alpha-1)}{\alpha(k\alpha-1)} ee^\top\right). \tag{32}$$

Thus, for any $i \neq j$, we observe that

$$\begin{aligned}
\left|\frac{\hat{\theta}_n(i) - \hat{\theta}_n(j)}{n} - \frac{k}{\alpha} \cdot \frac{v_n(i) - v_n(j)}{n}\right| &= \left|(e_i - e_j)^\top \left(Z^{-1} - \mathbb{E}\left[Z\right]^{-1}\right) \frac{v_n}{n}\right| \\
&\leq \|e_i - e_j\| \cdot \left\|Z^{-1} - \mathbb{E}\left[Z\right]^{-1}\right\|_2 \cdot \left\|\frac{v_n}{n}\right\| \\
&\leq 2C\alpha \cdot \left\|Z^{-1} - \mathbb{E}\left[Z\right]^{-1}\right\|_2 \qquad \text{(by equation (17))} \\
&\leq \frac{4\alpha \log\left(2k\delta^{-1}\right)}{\sqrt{n}} \frac{2}{(1 - \alpha/k)^2}. \\
& \qquad \text{(by equations (14), (15), (18))}
\end{aligned}$$

The last step follows by applying equations (15) and (18) into equation (14). Thus, we have finished the proof of equation (30). □

Second we show that provided n and d are sufficiently large, a separation exists in v_n between related and unrelated tasks.

Proof of Theorem 3.2. We calculate $v_n(i)/n$ for all $i = 1, \ldots, k$ and compare it between a related task and an unrelated task. We first compare their expectations over the randomly sampled subsets. By equation (13), we get

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

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

$$\left|\frac{v_n(i) - v_n(j)}{n} - \frac{\sum_{T \in \mathcal{S}: i \in T} f_t(T) - \sum_{T \in \mathcal{S}: j \in T} f_t(T)}{|\mathcal{S}|}\right| \leq \frac{2Ck\delta^{-1/2}}{\sqrt{n}}. \tag{33}$$

To deal with equation (33), we shall apply a union bound over the sample covariance of every subset T in $\mathcal{S}$ to show that they are close to their expectation. By Gaussian covariance estimation results (e.g., equation (6.12) in Wainwright [46]), for a fixed $T \in \mathcal{S}$, we get

$$\left|\frac{1}{md} \sum_{j=1}^{m} X_{i_j}^\top X_{i_j} - \mathrm{Id}_{p \times p}\right| \leq 2\sqrt{\frac{p}{md}} + 2\epsilon + \left(\sqrt{\frac{p}{md}} + \epsilon\right)^2,$$

with probability at least $1 - 2\exp(-md\epsilon^2/2)$. With a union bound over all $T \in \mathcal{S}$, we have that the above holds with probability at least $1 - \delta$ for all $T \in \mathcal{S}$, with $\epsilon = \sqrt{2\alpha k \log(2k\delta^{-1})/(md)}$. Let ε_1 denote the error term above:

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

Let $u_T = \frac{1}{md} \sum_{j \in T} X_j^\top \epsilon^{(j)}$, for any $T \in \mathcal{S}$. Therefore, , one can verify that

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

Notice that

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

If j is a related task, then the expectation over $\varepsilon^{(j)}$ is equal to $a^2\,\mathrm{Id}$ by our assumption. If j is an unrelated task, on the other hand, then the expectation over $\varepsilon^{(t)}$ is equal to $b^2\,\mathrm{Id}$. Let $s(T)$ be equal to the number of similar tasks in T. Thus,

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

To argue about the deviation error of $\|u_T\|^2$, we use the following two estimates (see, e.g., Vershynin [44]), which holds with high probability:

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

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

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

To finish the proof, consider a related task i versus an unrelated task j. Provided that

$$(1 - 3\varepsilon_1)\frac{p(a^2 - b^2)}{m^2 d} \ge (1 + 3\varepsilon_1)\frac{p\sqrt{d}a^2}{d^2} + \frac{2Ck\delta^{-1/2}}{\sqrt{n}}, \tag{34}$$

there must exist a threshold that separates all the related tasks from the unrelated tasks. One can verify that condition (34) is satisfied when

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

Set the threshold γ as k/α times any value between the left and right hand side of equation (34). Thus, when n and d satisfy the condition above, combined with Lemma D.3, with high probability, for any i such that $\hat{\theta}_n(i) < \gamma$, i must be a related task. When $\hat{\theta}_n(j) > \gamma$, i much be a unrelated task. Thus, we have finished the proof. □

E Experiment Details

We describe details that were left out from Section C. 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 omitted from Section C.

E.1 Additional experimental setup

Experimental setup for predicting higher-order transfers. Figure 2 (Top) measures the accuracy of task modeling in predicting whether combining a set of source tasks S with a primary target task leads to a positive transfer to the target task. We measure the positive transfer as whether training with the set of tasks S improves single task learning of the target task. Figure 2 (Bottom) measures the correlation between multitask prediction losses $f_t(S)$ and task model predictions $g(S)$ as the size of subset $|S|$ varies. For previous approaches [42, 18], we use the higher-order approximation that averages first-order task affinity in a set S as the prediction score $g(S)$. Both figures are studied on the target task HI with fifty source tasks.

Figure 3 measures the transferability from a source task to a target task in multitask learning. For each figure, we fix a target task and vary the source task. We measure the transferability as the difference between: (i) the multitask prediction result from a source and the target task averaged over multiple subsets containing the source task; (ii) single task learning with the target task alone. For both, we measure the worst-group accuracy of the target task.

Figure 1 (Top) provides the convergence of task modeling on binary prediction tasks. We use four target tasks, including HI, LA, MN, and NM. Figure 1 (Bottom) provides the convergence of task modeling on text datasets. We collect twenty-five tasks from several natural language processing benchmarks, including GLUE [48], SuperGLUE [47], TweetEval [3], and ANLI [30]. The collection

Table 6: 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.

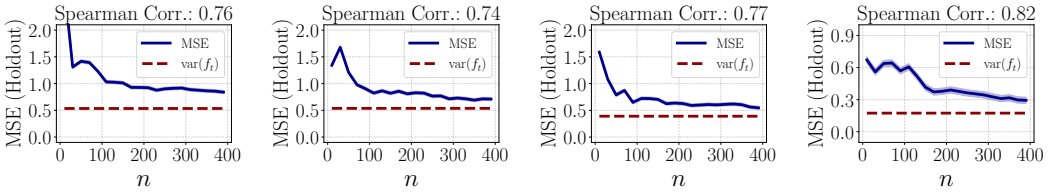
spans numerous categories of tasks, including sentence classification, natural language inference, and question answering. Table 6 shows the statistics of the twenty-five tasks. We choose four target tasks, including STS-B, RTE, WNLI, and Emotion. We use BERT-Mini as the encoder. The encoder module is shared for all tasks, and a separate predictor is assigned for each task. We construct the task models using $n = 200$ sampled sets with $|S| = 5$ out of $k = 24$ source tasks. We construct a holdout set of size 50. We set the prediction loss f_t as the loss of task t.

The abbreviation of each US state follows the convention. We include the ones we have referred to for reference: California (CA), Hawaii (HI), Kansas (KS), Louisiana (LA), Minnesota (MN), Nevada (NV), New Jersey (NJ), New Mexico (NM), Rhode Island (RI), and South Carolina (SC).

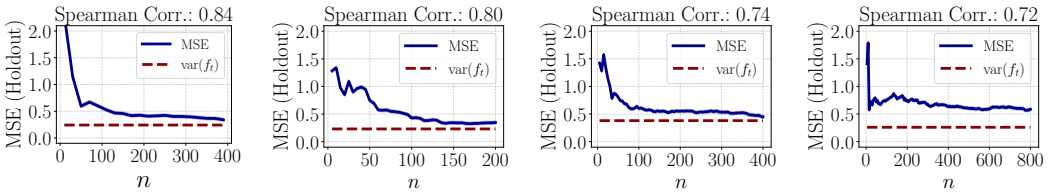
Implementation details. We report the results for baselines by running the official open-sourced implementations. We describe the hyperparameters for baselines as follows. For higher-order approximation [42] and task affinity grouping [18], we compute the task affinity scores between source tasks and target tasks. Then, we select m source tasks as the tasks with the largest task affinity scores for each target task. m is searched between 0 and the number of the source tasks. For gradient decomposition [14], we search the number of decomposition basis and auxiliary task gradient direction parameters, following the search space in [14]. For target-aware weighted training [12], we search the task weight learning rate in $[10^{-2}, 10^2]$. For our approach (cf. Algorithm 1), we use the threshold γ in the range of $[-0.2, 0.2]$. The hyperparameters are tuned on the validation dataset by grid search. For each target task, we search 10 times over the hyperparameter space. We use the same number of trials in tuning hyperparameters for baselines.

E.2 Results on the convergence of task modeling

Section 3.1 presents that the sample complexity for task models to converge is nearly linear to the number of tasks. We further validate the convergence of task modeling on ten more target tasks, including five binary classification tasks and five text classification tasks with noisy supervision sources. We measure the MSE between task model predictions and empirically training results on the holdout set. The experimental setup of is described in Section C.1. Figure 5 shows the results. We observe similar results as in Figure 1 that the MSE of task models consistently converges to the variance of the prediction loss. Hence, we conclude that the convergence of task modeling generally holds for various datasets.



(a) Target task: KS (b) Target task: NJ (c) Target task: NV (d) Target task: SC



(e) Target task: RI (f) Target task: Chemprot (g) Target task: TREC (h) Target task: Semeval

Figure 5: The MSE of task modeling consistently converges close to the variance of f_t for various tasks. **(a-e)** Binary classification tasks. **(f-h)** Text classification tasks with noisy supervision sources.

Demographic parity	HI	KS	LA	NJ	NV	Avg. Rank
Empirical risk minimization	12.95±1.76	4.09±1.15	26.30±1.21	26.06±0.53	12.62±1.99	6.3
Hard parameter sharing	8.25±1.31	4.06±1.17	21.24±0.66	27.73±0.94	13.35±0.51	5.0
Higher-order approx. [42]	8.63±2.95	6.15±3.00	22.83±0.53	26.14±0.29	13.15±0.64	5.6
Gradient similarity [18]	9.39±1.45	3.26±1.21	20.61±0.55	25.51±1.17	12.50±1.10	3.5
Task affinity grouping [18]	8.93±2.35	3.97±0.61	20.72±0.86	25.21±0.68	12.24±0.82	2.8
Weighted training [12]	18.12±1.80	4.84±0.71	25.77±0.94	25.66±0.38	12.40±0.74	6.0
Gradient decomposition [14]	11.98±2.55	2.40±0.91	27.38±0.93	26.10±0.55	13.29±0.40	5.6
Task modeling (Alg. 1)	**7.63±2.12**	**1.06±0.62**	**17.25±1.13**	**24.96±0.63**	**11.34±1.31**	**1.0**
Equality of opportunity	HI	KS	LA	NJ	NV	
Empirical risk minimization	9.86±1.29	1.43±3.62	29.64±3.24	22.43±1.02	13.61±3.67	6.0
Hard parameter sharing	3.86±0.84	2.03±2.11	21.26±1.35	24.43±1.49	12.14±2.21	5.0
Higher-order approx. [42]	3.55±2.85	4.34±3.18	22.88±1.72	22.98±1.18	12.92±2.23	5.3
Gradient similarity [18]	3.96±0.60	1.72±1.94	20.89±0.92	**21.48±1.79**	12.78±2.92	3.6
Task affinity grouping [18]	4.27±0.25	1.18±0.97	20.66±1.43	21.89±0.69	11.66±1.58	3.0
Weighted training [12]	4.21±2.25	1.40±2.14	30.38±2.17	23.26±0.30	11.77±1.01	5.6
Gradient decomposition [14]	3.18±4.92	6.01±2.47	32.31±0.86	22.83±1.01	15.48±1.17	6.1
Task modeling (Alg. 1)	**0.24±1.32**	**0.21±1.34**	**14.14±2.32**	**21.48±0.90**	**9.65±3.49**	**1.0**

Table 7: Violation of two fairness measures (demographic parity and equality of opportunity) on six binary prediction tasks with tabular features, averaged over ten random seeds.

E.3 Omitted results from Section C

Results for improving fairness measures. 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 C.3, we consider two fairness measures: demographic parity and equal opportunity [15]. The demographic parity measure is defined as: $|\Pr[\hat{y} = 1 \mid g = \text{black}] - \Pr[\hat{y} = 1 \mid g = \text{white}]|$. This measures the difference of the positive rates between the white and African American demographic groups. The equality of opportunity measure is defined as: $|\Pr[\hat{y} = 1 \mid y = 1, g = \text{black}] - \Pr[\hat{y} = 1 \mid y = 1, g = \text{white}]|$. This measures the difference of the true positive rates between the two groups. We consider the binary classification tasks with multiple subpopulation groups. Table 7 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 data augmentation. Second, our approach consistently reduces both fairness measure violations more by **1.26%** and **2.31%** on average than previous multitask learning approaches, respectively.

Varying the number of sampled sets n. We study how the number of sampled sets affect the selected task in the constructed task models. We measure the effect by comparing the 10 tasks with the smallest coefficients estimated from $n = 100, 200, 400$ subsets on two target tasks. We observe that using 100 and 200 subsets identifies 7 and 9 the same source tasks as using 400, respectively. Thus, we conclude that task selection results remain stable to the number of sampled sets.

We report the selected tasks with different numbers of sampled sets in the following. On target task HI, with 400 subsets, the ten tasks with the smallest task model coefficients are {CA NY TX FL PA IL OH NJ MI MA}. Using 200 subsets selects {CA NY TX FL PA IL OH NJ MI MA}. Using 100 subsets selects {CA TX NY PA OH FL NJ IL IN CO}. On target task LA, with 400 subsets, the ten tasks with the smallest task model coefficients are {CA TX NY FL IL GA PA MI NJ VA}. Using 200 subsets selects {CA TX NY FL IL PA NJ GA MI NC}. Using 100 subsets selects {CA NY TX FL IL NC GA IN CO PA}.

Runtime results. We report the GPU hours of contructing task models for each of the eleven target tasks in Table 3 and 4. Dataset (GPU hours) are listed in the following: Youtube (4.0), TREC (37.0), CDR (55.4), Chemprot (68.2), Semeval (85.9), HI (42.4), KS (44.0), LA (49.9), NJ (47.6), NV (43.7), SC (50.2).