The Annals of Statistics
2023, Vol. 51, No. 5, 2015–2039
https://doi.org/10.1214/23-AOS2319

ADAPTIVE AND ROBUST MULTI-TASK LEARNING

BY YAQI DUAN[1,a] AND KAIZHENG WANG[2,b]

[1]*Leonard N. Stern School of Business, New York University,* [a]yaqi.duan@stern.nyu.edu
[2]*Department of IEOR and Data Science Institute, Columbia University,* [b]kaizheng.wang@columbia.edu

We study the multitask learning problem that aims to simultaneously analyze multiple data sets collected from different sources and learn one model for each of them. We propose a family of adaptive methods that automatically utilize possible similarities among those tasks while carefully handling their differences. We derive sharp statistical guarantees for the methods and prove their robustness against outlier tasks. Numerical experiments on synthetic and real data sets demonstrate the efficacy of our new methods.

1. Introduction. Multitask learning (MTL) solves a number of learning tasks simultaneously. It has become increasingly popular in modern applications with data generated by multiple sources. When the tasks share certain common structures, a properly chosen MTL algorithm can leverage that to improve the performance. However, task relatedness is usually unknown and hard to quantify in practice; heterogeneity can even make multitask approaches perform worse than independent task learning, which trains models separately on their individual data sets. In this paper, we study MTL from a statistical perspective and develop a family of reliable approaches that adapt to the unknown task relatedness and are robust against outlier tasks with possibly contaminated data.

To set the stage, let $m \geq 1$ be the number of tasks and $\{\mathcal{X}_j\}_{j=1}^m$ be sample spaces. For every $j \in [m]$, let $\mathcal{P}_j$ be a probability distribution over $\mathcal{X}_j$, $\mathcal{D}_j = \{\boldsymbol{\xi}_{ji}\}_{i=1}^{n_j}$ be samples drawn from $\mathcal{P}_j$, and $\ell_j : \mathbb{R}^d \times \mathcal{X}_j \to \mathbb{R}$ be a loss function. The jth task is to estimate the population loss minimizer

$$\boldsymbol{\theta}_j^\star \in \operatorname*{arg\,min}_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{P}_j} \ell_j(\boldsymbol{\theta}, \boldsymbol{\xi})$$

from the data. For instance, in multitask linear regression, each sample $\boldsymbol{\xi}_{ji}$ can be written as $(\boldsymbol{x}_{ji}, y_{ji})$, where $\boldsymbol{x}_{ji} \in \mathbb{R}^d$ is a covariate vector and y_{ji} is a response. The loss function is $\ell_j(\boldsymbol{\theta}, (\boldsymbol{x}, y)) = (\boldsymbol{x}^\top \boldsymbol{\theta} - y)^2$.

Define the empirical loss function of the jth task as $f_j(\boldsymbol{\theta}) = \frac{1}{n_j} \sum_{i=1}^{n_j} \ell_j(\boldsymbol{\theta}, \boldsymbol{\xi}_{ji})$. Many MTL methods [13] are formulated as constrained minimization problems of the form

$$\min_{\boldsymbol{\Theta} \in \Omega} \left\{ \sum_{j=1}^m w_j f_j(\boldsymbol{\theta}_j) \right\}, \tag{1.1}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m) \in \mathbb{R}^{d \times m}$, $\{w_j\}_{j=1}^m$ are weight parameters (e.g., $w_j = n_j$), and $\Omega \subseteq \mathbb{R}^{d \times m}$ encodes the prior knowledge of task relatedness. Independent task learning corresponds to $\Omega = \mathbb{R}^{d \times m}$. Setting $\Omega = \{\boldsymbol{\beta} \mathbf{1}_m^\top : \boldsymbol{\beta} \in \mathbb{R}^d\}$ yields the data pooling strategy, where we simply merge all data sets to train a single model. It is also easy to construct parameter spaces so that the learned parameter vectors share part of their coordinates, cluster around a

Received May 2022; revised May 2023.
MSC2020 subject classifications. Primary 62R07; secondary 62F10.
Key words and phrases. Multitask learning, adaptivity, robustness, model misspecification, clustering, low-rank model.

few points, lie in a low-dimensional subspace, etc. In general, the hard constraint $\boldsymbol{\Theta} \in \Omega$ in (1.1) is overly rigid. When Ω fails to reflect the task structures, the model misspecification may have a huge negative impact on the performance.

To resolve the aforementioned issue, we propose to solve an augmented program

$$\min_{\boldsymbol{\Theta}\in\mathbb{R}^{d\times m},\boldsymbol{\Gamma}\in\Omega}\left\{\sum_{j=1}^{m} w_j\left[f_j(\boldsymbol{\theta}_j)+\lambda_j\|\boldsymbol{\theta}_j-\boldsymbol{\gamma}_j\|_2\right]\right\}, \tag{1.2}$$

obtain an optimal solution $(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\Gamma}})$ and then use $\widehat{\boldsymbol{\Theta}}$ as the final estimate. Here, $\{\lambda_j\}_{j=1}^m$ are regularization parameters and $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_m)$. Each task receives its own estimate $\widehat{\boldsymbol{\theta}}_j$, while the penalty terms shrink $\widehat{\boldsymbol{\Theta}}$ toward a prototype $\widehat{\boldsymbol{\Gamma}}$ in the prescribed model space Ω so as to promote relatedness among tasks. Our framework (1.2) can deal with different levels of task relatedness if we properly tune the regularization parameters $\{\lambda_j\}_{j=1}^m$. When Ω nicely captures the underlying structure, we can pick sufficiently large $\{\lambda_j\}_{j=1}^m$ so that the cusp of the ℓ_2 penalty at zero enforces the strict equality $\widehat{\boldsymbol{\Theta}} = \widehat{\boldsymbol{\Gamma}}$. The new procedure then reduces to the classical formulation (1.1). On the other hand, when Ω fails to reflect the structure, we take small λ_j's to guarantee each $\widehat{\boldsymbol{\theta}}_j$'s fidelity to its associated data. Observe that

$$\widehat{\boldsymbol{\theta}}_j \in \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min}\left\{f_j(\boldsymbol{\theta})+\lambda_j\|\boldsymbol{\theta}-\widehat{\boldsymbol{\gamma}}_j\|_2\right\} \quad \forall j \in [m].$$

In words, $\widehat{\boldsymbol{\theta}}_j$ minimizes a perturbed version of the loss function f_j associated to the jth task. When λ_j is not too large, the perturbation has limited influence and $\widehat{\boldsymbol{\theta}}_j$ stays close to the output of independent task learning $\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} f_j(\boldsymbol{\theta})$. This provides a safenet in case Ω is significantly misspecified. We see that strong regularization helps utilize task relatedness if that exists, while weak regularization better deals with heterogeneity.

Interestingly, there is a simple choice of $\{\lambda_j\}_{j=1}^m$ that provides the best of both worlds, regardless of whether the prescribed model space Ω captures the underlying structure or not. Roughly speaking, when $n_1 = \cdots = n_j = n$, our theory suggests choosing $w_j = 1$ and $\lambda_j = c\sqrt{\frac{d}{n}}$ for some constant c; when $\{n_j\}_{j=1}^m$ are different, our general results recommend $w_j = n_j$ and $\lambda_j = c\sqrt{\frac{d}{n_j}}$. In both cases, the factor c is shared by all of the m tasks. The estimator has a single tuning parameter rather than m different ones, which is practically appealing. Thanks to the *unsquared* ℓ_2 penalties in (1.2), the procedure automatically enforces an appropriate degree of relatedness among the learned models.

Moreover, the method can tolerate a reasonable fraction of exceptional tasks that are dissimilar to others or even have their data contaminated. Given the above merits, we name the framework as *A̲daptive and R̲obust M̲U̲ltitask L̲earning*, or ARMUL for short.

Main contributions. Our contributions are two-fold.

- (Methodology) We introduce a flexible framework for multitask learning. It works as a wrapper around any MTL method of the form (1.1), enhancing its ability to handle heterogeneous tasks.
- (Theory) We establish sharp guarantees for the framework on its adaptivity and robustness. Our analysis provides one customized statistical error bound for every single task.

Related work. Our work relates to a vast literature on integrative data analysis [43]. A classical example is simultaneous estimation of multiple Gaussian means. The specification of our method in this scenario is related to various shrinkage estimates [22, 26, 39]; see Section 2 for more discussions. An extension of multitask mean estimation is linear regression with multiple responses [11], which is a special form of multitask linear regression with shared covariates. Chen et al. [14] studied Stein-type shrinkage estimates for multitask linear regression with Gaussian data. Negahban and Wainwright [5, 46, 48, 54, 55, 61, 68] and [12] investigated high-dimensional (generalized) linear MTL where the tasks have similar sparsity patterns. There are also MTL approaches proposed to enforce other types of model similarities such as clustering structures [28, 29, 37, 51], low-rank structures [1, 4, 42], among others. The above list is far from being exhaustive.

Our study is largely motivated by the great empirical success in MTL with parameter augmentation [15, 29, 38]. Our idea of nonsmooth regularization originates from the seminal works by [22] and [21] on adaptive sparse estimation. Beyond the coordinatewise sparsity of vectors, recent studies have developed the sum of ℓ_2 penalties to promote columnwise sparsity in matrix estimation problems such as robust PCA [45, 50, 67] and robust low-rank MTL [15, 57]. Our design of the penalty is closely related to theirs. The ARMUL penalty also looks similar to the group lasso penalty $\sum_{\ell=1}^{d}(\sum_{j=1}^{m}|\theta_{j\ell}|^2)^{1/2}$ for variable selection in sparse MTL [46]. While the group lasso sums up the norms of rows (variables), ours does that to the columns (tasks).

Below we provide a selective overview of existing theories that are closely connected to our analysis of adaptivity and robustness. Wu et al. [66] and [53] analyzed the impact of task relatedness on linear models and one-hidden-layer neural networks when there are two tasks. Balcan et al. [6] and Denevi et al. [19] studied online MTL and showed the benefit of task relatedness. Konstantinov et al. [41] investigated multitask PAC learning with adversarial corruptions. They assumed homogeneous tasks and focused on robustness against different types of adversaries. Hanneke and Kpotufe [31] studied the adaptation in nonparametric MTL under the Bernstein class condition. Du et al. [23] and [65] considered representation learning from multiple data sets when the true statistical models share common latent structures. In the agnostic learning framework, [7] and [49] presented generalization bounds on the average risk across tasks, and [8] studied task-specific error bounds.

Outline. The rest of the paper is organized as follows. Section 2 studies multitask Gaussian mean estimation as a warm-up example. Section 3 presents the methodology. Section 4 conducts a sharp analysis of adaptivity and robustness. Section 5 verifies the theories and tests the methodology through numerical experiments. Finally, Section 6 concludes the paper and discusses possible future directions.

Notation. The constants $c_1, c_2, C_1, C_2, \dots$ may differ from line to line. Define $x_+ = \max\{x, 0\}$ for $x \in \mathbb{R}$. We use the symbol $[n]$ as a shorthand for $\{1, 2, \dots, n\}$ and $|\cdot|$ to denote the absolute value of a real number or cardinality of a set. For nonnegative sequences $\{a_n\}_{n=1}^{\infty}$ and $\{b_n\}_{n=1}^{\infty}$, we write $a_n \lesssim b_n$ or $a_n = O(b_n)$ or $b_n = \Omega(a_n)$ if there exists a positive constant C such that $a_n \leq Cb_n$. In addition, we write $a_n \asymp b_n$ if $a_n \lesssim b_n$ and $b_n \lesssim a_n$; $a_n = o(b_n)$ if $a_n = O(c_n b_n)$ for some $c_n \to 0$. Let $\mathbf{1}_d$ be the d-dimensional all-one vector and $\{\boldsymbol{e}_j\}_{j=1}^{d}$ canonical bases of $\mathbb{R}^d$. Define $\mathbb{S}^{d-1} = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 = 1\}$ and $B(\boldsymbol{x}, r) = \{\boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{y} - \boldsymbol{x}\|_2 \leq r\}$ for $\boldsymbol{x} \in \mathbb{R}^d$ and $r \geq 0$. For any matrix $\boldsymbol{A}$, we use $\boldsymbol{a}_j$ to refer to its jth column and let Range$(\boldsymbol{A})$ be its column space. $\|\boldsymbol{A}\|_2 = \sup_{\|\boldsymbol{x}\|_2=1}\|\boldsymbol{A}\boldsymbol{x}\|_2$ denotes the spectral norm and $\|\boldsymbol{A}\|_{\mathrm{F}}$ denotes the Frobenius norm. Define $\|X\|_{\psi_2} = \sup_{p\geq 1}\{p^{-1/2}\mathbb{E}^{1/p}|X|^p\}$ and $\|X\|_{\psi_1} = \sup_{p\geq 1}\{p^{-1}\mathbb{E}^{1/p}|X|^p\}$ for a random variable X; $\|\boldsymbol{X}\|_{\psi_2} = \sup_{\|\boldsymbol{u}\|_2=1}\|\langle\boldsymbol{u}, \boldsymbol{X}\rangle\|_{\psi_2}$ for a random vector $\boldsymbol{X}$.

2. Warm-up: Estimation of multiple Gaussian means.

In this section, we consider the multitask mean estimation problem as a warm-up example. We first introduce the setup and a simple estimation procedure. We relate the estimator to soft thresholding and Huber's location estimator. Then we show that it automatically adapts to the unknown task relatedness and is robust against a small fraction of tasks with contaminated data. Finally, we discuss the connection between our estimator and several fundamental topics in statistics and machine learning.

2.1. *Problem setup.* Suppose we want to simultaneously estimate the mean parameters of $m \geq 2$ Gaussian distributions $\{N(\theta_j^\star, 1)\}_{j=1}^m$. For each $j \in [m]$, we collect n i.i.d. samples $\{x_{ji}\}_{i=1}^n$ from $N(\theta_j^\star, 1)$. The m data sets $\{x_{1i}\}_{i=1}^n, \ldots, \{x_{mi}\}_{i=1}^n$ are independent. This is an extensively studied problem in statistics [25, 60] and a canonical example in multitask learning, where the jth learning task is to estimate $\theta_j^\star$.

- Without additional assumptions, it is natural to conduct maximum likelihood estimation (MLE). Due to the independence of data sets, MLE amounts to estimating each $\theta_j^\star$ by the sample mean $\bar{x}_j = \frac{1}{n}\sum_{i=1}^n x_{ji}$ of its associated data. The mean squared error is $\mathbb{E}(\bar{x}_j - \theta_j^\star)^2 = \frac{1}{n}$.
- If the parameters are very close, we may estimate them by the pooled sample mean $\bar{x} = \frac{1}{mn}\sum_{j=1}^m \sum_{i=1}^n x_{ji}$. In the ideal case $\theta_1^\star = \cdots = \theta_m^\star$, data pooling reduces the mean squared error to $\frac{1}{mn}$.
- We may use Bayesian procedures if $\{\theta_j^\star\}_{j=1}^m$ are independently drawn from some known prior distribution. When the prior itself has unknown parameters, empirical Bayes methods [27, 39] can be applied.

Since it is often hard to precisely quantify the prior knowledge in practice, we want an estimation procedure that automatically adapts to the unknown similarity among the tasks. Ideally, the procedure should also be robust against outlier tasks that are dissimilar to others or even contain corrupted data. To introduce our method, we first present optimization perspectives of MLE and its pooled version. Up to an affine transform, the negative log-likelihood function for the jth task is equal to

$$f_j(\theta) = \frac{1}{2n}\sum_{i=1}^n (x_{ji} - \theta)^2 \quad \forall \theta \in \mathbb{R}.$$

MLE returns one estimator $\bar{x}_j = \arg\min_{\theta_j \in \mathbb{R}} f_j(\theta_j)$ for each task, whereas data pooling outputs the same estimator $\bar{x} = \arg\min_{\theta \in \mathbb{R}} \sum_{j=1}^m f_j(\theta)$ for all tasks.

We propose to solve a convex optimization problem

$$(2.1) \qquad (\widehat{\theta}_1, \ldots, \widehat{\theta}_m, \widehat{\theta}) \in \underset{\theta_1, \ldots, \theta_m, \theta \in \mathbb{R}}{\arg\min} \left\{ \sum_{j=1}^m [f_j(\theta_j) + \lambda|\theta_j - \theta|] \right\}$$

and use $\{\widehat{\theta}_j\}_{j=1}^m$ to estimate $\{\theta_j^\star\}_{j=1}^m$. Here, $\lambda \geq 0$ is a penalty parameter and $\widehat{\theta}$ serves as a global coordinator. Similar to MLE, each task receives one individual estimator based on its loss function. Moreover, the penalty terms drive those estimators toward a common center. When $\lambda = 0$, $\widehat{\theta}_j = \bar{x}_j$. When $\lambda = \infty$, $\widehat{\theta}_j = \widehat{\theta} = \bar{x}$. Therefore, the method interpolates between MLE and its pooled version. We will derive a simple choice of λ with guaranteed quality outputs.

2.2. *Adaptivity and robustness.* It is easily seen from (2.1) that

$$(2.2) \qquad \widehat{\theta} \in \underset{\theta \in \mathbb{R}}{\arg\min} \sum_{j=1}^m \widetilde{f}_j(\theta) \quad \text{and} \quad \widehat{\theta}_j \in \underset{\theta \in \mathbb{R}}{\arg\min}\{f_j(\theta) + \lambda|\theta - \widehat{\theta}|\} \quad \forall j \in [m],$$

where $\widetilde{f}_j(\theta) = \min_{\xi \in \mathbb{R}}\{f_j(\xi) + \lambda|\theta - \xi|\}$ is the infimal convolution [32] of a quadratic loss $f_j(\cdot)$ and an absolute value penalty $\lambda|\cdot|$. It is well known that such infimal convolution is closely related to the Huber loss function [35] with parameter λ:

$$\rho_\lambda(x) = \begin{cases} x^2/2 & \text{if } |x| \leq \lambda, \\ \lambda(|x| - \lambda/2) & \text{if } |x| > \lambda. \end{cases}$$

See, for example, Section 6.1 of [20]. Based on that, we have the following elementary characterizations of $\widehat{\theta}$ and $\{\widehat{\theta}_j\}_{j=1}^m$. The proof is deferred to Appendix C.1 in the Supplementary Material [24].

LEMMA 2.1. *We have* $\widetilde{f}_j(\theta) = \rho_\lambda(\theta - \bar{x}_j) + \frac{1}{2n}\sum_{i=1}^n (x_{ji} - \bar{x}_j)^2$,

$$\widehat{\theta} \in \operatorname*{arg\,min}_{\theta \in \mathbb{R}} \sum_{j=1}^m \rho_\lambda(\theta - \bar{x}_j),$$

$$\widehat{\theta}_j = \widehat{\theta} + \operatorname{sgn}(\bar{x}_j - \widehat{\theta})(|\bar{x}_j - \widehat{\theta}| - \lambda)_+ = \bar{x}_j - \min\{\lambda, |\bar{x}_j - \widehat{\theta}|\} \operatorname{sgn}(\bar{x}_j - \widehat{\theta}) \quad \forall j \in [m].$$

According to Lemma 2.1, the global coordinator $\widehat{\theta}$ in (2.1) is a Huber estimator applied to sample means $\{\bar{x}_j\}_{j=1}^m$ of individual data sets. The estimators $\{\widehat{\theta}_j\}_{j=1}^m$ for $\{\theta_j^\star\}_{j=1}^m$ are shrunk toward $\widehat{\theta}$ by soft thresholding. Intuitively, we may view the procedure (2.1) as a combination of hypothesis testing and parameter estimation. The first step is to test the homogeneity hypothesis $H_0 : \theta_1^\star = \cdots = \theta_m^\star$, with λ controlling the significance level. When $\{\bar{x}_j\}_{j=1}^m$ are close enough, for example, $\max_{j \neq k} |\bar{x}_j - \bar{x}_k| \leq \lambda$, the parameters $\{\theta_j^\star\}_{j=1}^m$ do not seem to be significantly different. We apply data pooling and get $\widehat{\theta}_1 = \cdots = \widehat{\theta}_m = \widehat{\theta} = \bar{x}$. The exact equality $\widehat{\theta}_j = \widehat{\theta}$ is enforced by the cusp of the absolute value penalty $|\cdot|$ at zero. When all but a small fraction of $\{\bar{x}_j\}_{j=1}^m$ are close, the robustness property of the Huber loss makes $\widehat{\theta}$ a good summary of the majority; their corresponding $\widehat{\theta}_j$'s are equal to $\widehat{\theta}$. In general, the estimators $\{\widehat{\theta}_j\}_{j=1}^m$ can be different. It is worth pointing out that $|\widehat{\theta}_j - \bar{x}_j| \leq \lambda$ always holds, thanks to the Lipschitz smoothness of $|\cdot|$. This guarantees $\widehat{\theta}_j$'s fidelity to its associated data set $\{x_{ji}\}_{i=1}^n$. Hence, the proposed method easily handles heterogeneous tasks.

To analyze the statistical property of (2.1), we need to gauge the relatedness among tasks.

DEFINITION 2.1 (Parameter space). For any $\varepsilon \in [0, 1]$ and $\delta \geq 0$, define

$$\Omega(\varepsilon, \delta) = \Big\{\boldsymbol{\theta}^\star \in \mathbb{R}^m : \min_{\theta \in \mathbb{R}} \max_{j \in S} |\theta_j^\star - \theta| \leq \delta \text{ and } |S^c|/m \leq \varepsilon \text{ for some } S \subseteq [m]\Big\}.$$

We associate every $\boldsymbol{\theta}^\star \in \Omega(\varepsilon, \delta)$ with a subset $S = S(\boldsymbol{\theta}^\star)$ of $[m]$ that satisfies the above requirements.

ASSUMPTION 2.1 (Task relatedness). The m data sets $\{x_{1i}\}_{i=1}^n, \ldots, \{x_{mi}\}_{i=1}^n$ are statistically independent and there exists $\boldsymbol{\theta}^\star \in \Omega(\varepsilon, \delta)$ such that for any $j \in [m]$, $\{x_{ji}\}_{i=1}^n$ are i.i.d. $N(\theta_j^\star, 1)$.

We say the m tasks are (ε, δ)-related when Assumption 2.1 holds. In words, all but an ε fraction of the mean parameters $\{\theta_j^\star\}_{j=1}^m$ live in an interval with half-width δ; the others can be arbitrary. Smaller ε and δ imply more similarity among tasks. The extreme case $\varepsilon = \delta = 0$ corresponds to $\theta_1^\star = \cdots = \theta_m^\star$. Any m tasks of Gaussian mean estimation are $(0, \max_{j \in [m]} |\theta_j^\star|)$-related.

Theorem 2.1 below characterizes the estimation errors. The proof can be found in Appendix C.2 [24].

THEOREM 2.1 (Adaptivity and robustness). *Let Assumption 2.1 hold. Choose any $t \geq 2$ and $\lambda = 6\sqrt{\frac{2(\log m+t)}{n}}$. There is a universal constant $C > 0$ such that with probability at least $1 - e^{-t}$,*

$$\max_{j\in S}|\widehat{\theta}_j - \theta_j^\star| < C\biggl(\sqrt{\frac{t}{mn}} + \min\biggl\{\delta, \sqrt{\frac{\log m + t}{n}}\biggr\} + \varepsilon\sqrt{\frac{\log m + t}{n}}\biggr),$$

$$\max_{j\in[m]\backslash S}|\widehat{\theta}_j - \theta_j^\star| < C\sqrt{\frac{\log m + t}{n}},$$

$$\frac{1}{m}\sum_{j=1}^{m}|\widehat{\theta}_j - \theta_j^\star|^2 \leq C\biggl(\frac{t}{mn} + \min\biggl\{\delta^2, \frac{\log m + t}{n}\biggr\} + \varepsilon \cdot \frac{\log m + t}{n}\biggr).$$

REMARK 1 (Data contamination). We can further relax the assumption on task relatedness to allow the data sets $\{\mathcal{D}_j\}_{j\notin S}$ to be arbitrarily contaminated. In that case, the results for $\max_{j\in S}|\widehat{\theta}_j - \theta_j^\star|$ in Theorem 2.1 continue to hold.

Theorem 2.1 provides maximum error bounds for the "good" tasks in S and "bad" tasks in $[m]\backslash S$, as well as the mean squared error (MSE) over all tasks. A crude error bound $\max_{j\in[m]}|\widehat{\theta}_j - \theta_j^\star| \lesssim \sqrt{\frac{\log m}{n}}$ always holds regardless of (ε, δ). On the other hand, elementary calculation shows that $\max_{j\in[m]}|\bar{x}_j - \theta_j^\star| \gtrsim \sqrt{\frac{\log m}{n}}$ with constant probability. Therefore, the new method is always comparable to MLE. That provides a safe net.

Moreover, the suggested penalty parameter $\lambda = 6\sqrt{\frac{2(\log m+t)}{n}}$ in Theorem 2.1 does not depend on ε or δ at all. The estimator automatically adapts to the unknown task relatedness, achieving higher accuracy when ε and δ are small. Up to logarithmic factors, the MSE bound reads

$$\frac{1}{m}\sum_{j=1}^{m}|\widehat{\theta}_j - \theta_j^\star|^2 \lesssim \frac{1}{mn} + \min\biggl\{\delta^2, \frac{1}{n}\biggr\} + \frac{\varepsilon}{n}. \tag{2.3}$$

The first term $\frac{1}{mn}$ is the MSE of the pooled sample mean $\bar{x}$ in the most homogeneous case $\theta_1^\star = \cdots = \theta_m^\star$. When $\varepsilon = \delta = 0$, only this term exists and our procedure reduces to data pooling. The second term $\min\{\delta^2, \frac{1}{n}\}$ is nondecreasing in the discrepancy δ among $\{\theta_j^\star\}_{j\in S}$. It increases first and then flattens out, never exceeding the error rate of MLE. When $\varepsilon = 0$, we have

$$\frac{1}{m}\sum_{j=1}^{m}|\widehat{\theta}_j - \theta_j^\star|^2 \lesssim \frac{1}{mn} + \min\biggl\{\delta^2, \frac{1}{n}\biggr\} \asymp \min\biggl\{\frac{1}{mn} + \delta^2, \frac{1}{n}\biggr\}.$$

Here, $\frac{1}{n}$ and $\frac{1}{mn} + \delta^2$ are the MSEs of the MLE and its pooled version, respectively. Therefore, the new method achieves the smaller error between the two. It is closely related to robust inference procedures considered by [10, 33] and others that (i) perform well when the parameter of interest $\boldsymbol{\theta}^\star$ truly lives in a small set (e.g., $\theta_1^\star = \cdots = \theta_m^\star$), and (ii) are nearly minimax optimal over a larger parameter space (e.g., $\mathbb{R}^m$). Our analysis covers a continuum of parameter spaces $\{\Omega(\varepsilon, \delta) : 0 \leq \varepsilon \leq 1, \delta \geq 0\}$ while those studies mostly look at the two extremes.

When $\varepsilon > 0$, the third term $\frac{\varepsilon}{n}$ in (2.3) is the price we pay for not knowing the index set S^c of tasks that may be very different from the others. As an illustration, suppose that $\delta = 0$ and $\{\theta_j^\star\}_{j\in S}$ are all equal to some $\theta^\star$. Then $\{\bar{x}_j\}_{j\in S}$ are i.i.d. $N(\theta^\star, 1/n)$ and $\{\bar{x}_j\}_{j\in S^c}$ can be arbitrary. $\widehat{\theta}$ is a Huber estimator of $\theta^\star$ based on ε-contaminated data $\{\bar{x}_j\}_{j=1}^m$. Our error bound has optimal dependence on ε up to a logarithmic factor [36], whereas the pooled MLE can be ruined by a single outlier task.

We now present a minimax lower bound for an idealized problem with known ε and δ. It is a special case ($d = 1$) of Theorem 4.3 for multivariate Gaussians.

THEOREM 2.2 (Minimax lower bound). *There exist universal constants $C, c > 0$ such that for any $\varepsilon \in [0, 1]$ and $\delta \geq 0$,*

$$\inf_{\widehat{\boldsymbol{\theta}}} \sup_{\boldsymbol{\theta}^\star \in \Omega(\varepsilon,\delta)} \mathbb{P}_{\boldsymbol{\theta}^\star}\left[\frac{1}{m}\sum_{j=1}^{m}|\widehat{\theta}_j - \theta_j^\star|^2 \geq C\left(\frac{1}{mn} + \min\left\{\delta^2, \frac{1}{n}\right\} + \frac{\varepsilon}{n}\right)\right] \geq c.$$

The ARMUL estimator achieves the oracle error up to a $\log m$ factor without knowing ε and δ. It would be interesting to investigate whether the logarithmic term is a fundamental price of adaptation, as is the case with sparse Gaussian mean estimation [21].

For any given $\boldsymbol{\theta}^\star \in \mathbb{R}^m$, there exist infinitely many pairs of (ε, δ) that make Assumption 2.1 hold. For instance, when $\boldsymbol{\theta}^\star = \boldsymbol{e}_1$, we can take any (ε, δ) in the set

$$\{(\varepsilon, \delta) \in [0, 1] \times [0, +\infty) : \varepsilon \geq 1/m \text{ or } \delta \geq 1\}.$$

The MSE bound in Theorem 2.1 holds simultaneously for all of those (ε, δ). Unfortunately, the bound is not directly computable from data. On the one hand, ε and δ are not uniquely defined. On the other hand, even if we set $\varepsilon = 0$, the estimation error of δ will be of order $1/\sqrt{n}$. This results in an error up to $O(1/n)$ in the estimated MSE bound and makes it meaningless, because $O(1/n)$ is the largest possible value of our MSE bound (up to a $\log m$ factor). A similar phenomenon arises in nonparametric estimation. As [47] pointed out, "although an estimate may be adaptive for squared error loss it may be impossible to make a data dependent claim on how well you have done."

2.3. *Discussions.* The estimation procedure and theory in this section have deep connections to several fundamental topics in statistics and machine learning.

2.3.1. *James–Stein estimators.* For the Gaussian mean estimation problem in Section 2.1, a sufficient statistic is $\sqrt{n}(\bar{x}_1, \dots, \bar{x}_m)^\top \sim N(\boldsymbol{\theta}^\star, \boldsymbol{I}_m)$. Therefore, we may assume $n = 1$ in the original problem without loss of generality. The goal then becomes estimating $\boldsymbol{\theta}^\star \in \mathbb{R}^m$ from a single sample $\boldsymbol{x} \sim N(\boldsymbol{\theta}^\star, \boldsymbol{I}_m)$. The MLE is $\widehat{\boldsymbol{\theta}}^{\mathrm{MLE}} = \boldsymbol{x}$. In a seminal paper, [39] proposed to shrink the MLE toward zero and introduce a new estimator $\widehat{\boldsymbol{\theta}}^{\mathrm{JS},0} = (1 - \frac{m-2}{\|\boldsymbol{x}\|_2^2})\boldsymbol{x}$. Surprisingly, when $m \geq 3$, the ℓ_2 risk of $\widehat{\boldsymbol{\theta}}^{\mathrm{JS},0}$ is always strictly smaller than that of $\widehat{\boldsymbol{\theta}}^{\mathrm{MLE}}$:

$$\mathbb{E}_{\boldsymbol{\theta}^\star}\|\widehat{\boldsymbol{\theta}}^{\mathrm{JS},0} - \boldsymbol{\theta}^\star\|_2^2 < \mathbb{E}_{\boldsymbol{\theta}^\star}\|\widehat{\boldsymbol{\theta}}^{\mathrm{MLE}} - \boldsymbol{\theta}^\star\|_2^2 \quad \forall \boldsymbol{\theta}^\star \in \mathbb{R}^d. \tag{2.4}$$

The shrinking point does not have to be $\mathbf{0}$. They also introduced another estimator

$$\widehat{\theta}_j^{\mathrm{JS}} = \bar{x} + \left(1 - \frac{m-3}{\sum_{j=1}^m (x_j - \bar{x})^2}\right)(x_j - \bar{x}) \quad \forall j \in [m],$$

whose entries are shrunk toward the pooled sample mean $\bar{x}$. They proved the same dominance as (2.4) for $\widehat{\boldsymbol{\theta}}^{\mathrm{JS}}$ when $m \geq 4$. The gain is the most significant when $\{\theta_j^\star\}_{j=1}^m$ are close and m is large. In the ideal case $\theta_1^\star = \cdots = \theta_m^\star$, we derive from equation (7.14) in [25] that $\mathbb{E}_{\boldsymbol{\theta}^\star}\|\widehat{\boldsymbol{\theta}}^{\mathrm{JS}} - \boldsymbol{\theta}^\star\|_2^2 = 3$, which is within a constant factor (3) times the ℓ_2 risk of the pooled sample mean. The MLE has risk $\mathbb{E}_{\boldsymbol{\theta}^\star}\|\widehat{\boldsymbol{\theta}}^{\mathrm{MLE}} - \boldsymbol{\theta}^\star\|_2^2 = m$.

Efron and Morris [27] adopted an empirical Bayes approach to the simultaneous estimation problem and derived class of estimators that dominate the MLE. The positive part version of the James–Stein estimator

$$\widehat{\theta}_j^{\mathrm{JS}+} = \bar{x} + \left(1 - \frac{m-3}{\sum_{j=1}^m (x_j - \bar{x})^2}\right)_+ (x_j - \bar{x}) \quad \forall j \in [m]$$

is one example, which avoids negative shrinkage factor. The lemma below connects $\widehat{\boldsymbol{\theta}}^{\mathrm{JS}}$ and $\widehat{\boldsymbol{\theta}}^{\mathrm{JS}+}$ to multitask learning with ridge regularization [29]; see the proof in Appendix C.3 [24].

LEMMA 2.2. *Let* $\lambda > 0$ *and*

$$(\widetilde{\theta}_1, \ldots, \widetilde{\theta}_m, \widetilde{\theta}) \in \underset{\theta_1,\ldots,\theta_m,\theta\in\mathbb{R}}{\arg\min} \left\{ \sum_{j=1}^m [(\theta_j - x_j)^2 + \lambda(\theta_j - \theta)^2] \right\}. \tag{2.5}$$

We have $\widetilde{\theta} = \bar{x}$ *and* $\widetilde{\theta}_j = \bar{x} + \frac{1}{1+\lambda}(x_j - \bar{x})$, $\forall j \in [m]$. *If we define* $S = \sum_{j=1}^m (x_j - \bar{x})^2$, *then*

- $\widetilde{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}^{\mathrm{JS}}$ *when* $S > m-3$ *and* $\lambda = \frac{m-3}{S-(m-3)}$;
- $\widetilde{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}^{\mathrm{JS}+}$ *when* $\lambda = \frac{\min\{S,m-3\}}{S-\min\{S,m-3\}}$, *with the convention that* $c/0 = +\infty$ *for any* $c > 0$.

Our estimator $\widehat{\boldsymbol{\theta}}$ is defined by the ℓ_1-regularized program (2.1) that differs from (2.5) in the penalty function. As a result, the entries $\{\widehat{\theta}_j\}_{j=1}^m$ are shrunk toward a Huber estimator $\widehat{\theta}$ instead of the pooled sample mean used by James–Stein estimators; see Lemma 2.1. The nonsmooth ℓ_1 penalty can shrink the difference $\widehat{\theta}_j - \widehat{\theta}$ to exact zero. The relation between Huber loss, quadratic loss and ℓ_1 penalty function has also been used by [3, 30, 58] and [18] in wavelet thresholding and robust statistics.

The James–Stein estimators $\widehat{\boldsymbol{\theta}}^{\mathrm{JS},0}$, $\widehat{\boldsymbol{\theta}}^{\mathrm{JS}}$ and $\widehat{\boldsymbol{\theta}}^{\mathrm{JS}+}$ are tailored for the Gaussian mean problem. Their strong theoretical guarantees such as (2.4) are built upon analytical calculations of the ℓ_2 risk under the Gaussianity assumption. In contrast, our estimator $\widehat{\boldsymbol{\theta}}$ is constructed from penalized MLE framework (2.1), which easily extends to general multivariate M-estimation problems. We want the estimator to benefit from possible similarity among tasks while still being reliable in unfavorable circumstances; see Theorem 2.1. In the worst case, the price of generality is an extra logarithm factor in the risk.

2.3.2. *Limited translation estimators.* James–Stein estimators improve over the MLE in terms of ℓ_2 risk, which measures the average performance over parameters $\{\theta_j^\star\}_{j=1}^m$. There is no guarantee on the individuals. It is well known that the estimators underperform MLE by a large margin for $\theta_j^\star$'s far from the bulk. To make matters worse, such exceptional cases also significantly reduce the overall ℓ_2 efficacy. Efron and Morris [26] and [60] proposed limited translation estimators that restrict the amount of shrinkage. Hence, those estimators cannot deviate far from the MLE. By carefully setting the restrictions, they are able to control the maximum (ℓ_∞) error over all parameters. According to Lemma 2.1, our estimator $\widehat{\boldsymbol{\theta}}$ also has limited translation bounded by λ. Theorem 2.1 presents a sharp bound on the ℓ_∞ error.

2.3.3. *Soft-thresholding for sparse estimation.* When the mean vector $\boldsymbol{\theta}^\star$ is assumed to be sparse, it is natural to shrink many entries of the estimator to exact zero. Donoho [22] studied the ℓ_1-regularized estimator

$$\widehat{\boldsymbol{\theta}}^{\ell_1} \in \underset{\boldsymbol{\theta}\in\mathbb{R}^m}{\arg\min} \left\{ \sum_{j=1}^m [(\theta_j - x_j)^2 + \lambda|\theta_j|] \right\}$$

and its minimax optimality. For each $j \in [m]$, $\widehat{\theta}_j^{\ell_1} = x_j - \min\{\lambda, |x_j|\}\operatorname{sgn}(x_j)$ is a soft-thresholded version of x_j. If $|x_j| \leq \lambda$, then $\widehat{\theta}_j^{\ell_1} = 0$. Soft-thresholding and ℓ_1 regularization have wide applications in statistics, including parameter estimation subject to good risk properties at zero [9], ideal spatial adaptation [21], variable selection [63], etc. Our use of the ℓ_1 penalty in (2.1) is inspired by this line of research. By Lemma 2.1, the difference $\widehat{\theta}_j - \widehat{\theta}$ between individual estimator and the global coordinator is soft-thresholded. Merging some $\widehat{\theta}_j$'s to $\widehat{\theta}$ pools the information across similar tasks. Soft-thresholding has been used for combining the information in a small, high-quality data set and a less costly one with a possibly different distribution; see [14] and [16]. Our formulation (2.1) handles multiple data sets.

2.3.4. *Homogeneity of parameters.* An extension of sparsity is homogeneity, which refers to the phenomenon that parameters in similar subgroups are close to each other. Various methods are developed to exploit such structure in high-dimensional regression, including fused lasso [64], grouping pursuit [59] and CARDS [40]. Our method (2.1) uses one global coordinator to utilize the homogeneity when a majority of parameters live within the same small region. In Section 3, we will incorporate more than one coordinators to deal with multiple clusters of parameters.

2.3.5. *Minimax lower bounds.* For sparse Gaussian mean estimation, [22] derived the minimax lower bound on the ℓ_2 risk over $\{\boldsymbol{\theta}^\star \in \mathbb{R}^m : \|\boldsymbol{\theta}^\star\|_0 \leq \varepsilon m\}$ for $\varepsilon \in (0, 1)$, with precise constant factors. Here, $\|\boldsymbol{x}\|_0 = |\{i : x_i \neq 0\}|$ is the ℓ_0 pseudo-norm. Their parameter space is a subset of ours with $\delta = 0$. We aim to cover broader regimes but make no endeavor to optimize the constants. In a recent work, [17] studied fundamental limits of multitask and federated learning. Their definition of task relatedness is similar to ours in Assumption 2.1 with $\varepsilon = 0$. They construct m logistic models whose discrepancies are quantified by some parameter δ, and derive a minimax lower bound on the estimation error of the form $\min\{\frac{1}{\sqrt{mn}} + \delta, \frac{1}{\sqrt{n}}\}$. From there, they show that the optimal rate is achieved by either MLE or its pooled version. Our lower bound in Theorem 2.2 is proved for the canonical Gaussian mean problem and allows an ε fraction of the tasks to be arbitrarily different from the others. In that case, neither MLE nor pooled MLE is optimal.

3. Methodologies.

In this section, we present our framework for Adaptive and Robust MUltitask Learning (ARMUL). We focus on three important cases and provide algorithms for their efficient implementations.

3.1. *Adaptive and robust multitask learning.* Let $m \in \mathbb{Z}_+$. For every $j \in [m]$, let $\mathcal{P}_j$ be a probability distribution over a sample space $\mathcal{X}_j$ and $\ell_j : \mathbb{R}^d \times \mathcal{X}_j \to \mathbb{R}$ be a loss function. Suppose that we collect n_j i.i.d. samples $\mathcal{D}_j = \{\boldsymbol{\xi}_{ji}\}_{i=1}^{n_j}$ from $\mathcal{P}_j$ for every j, and the m data sets $\{\mathcal{D}_j\}_{j=1}^m$ are independent. The jth learning task is to minimize the population risk $\mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{P}_j} \ell_j(\boldsymbol{\theta}; \boldsymbol{\xi})$ by estimating the population risk minimizer $\boldsymbol{\theta}_j^\star \in \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{P}_j} \ell_j(\boldsymbol{\theta}; \boldsymbol{\xi})$ based on $\mathcal{D}_j$. For statistical estimation in well-specified models, $\boldsymbol{\theta}_j^\star$ is the true parameter and ℓ_j can be the negative log-likelihood function. Multitask learning (MTL) targets all of the m tasks simultaneously. The difficulty comes from the unknown task relatedness. It is often unclear whether and how a task can be better resolved by incorporating the information in other tasks.

Define the jth empirical loss function $f_j(\boldsymbol{\theta}) = \frac{1}{n_j} \sum_{i=1}^{n_j} \ell(\boldsymbol{\theta}; \boldsymbol{\xi}_{ji})$. Many MTL algorithms can be formulated as constrained loss minimization problems of the form

$$\min_{\boldsymbol{\Theta} \in \Omega} \left\{ \sum_{j=1}^m w_j f_j(\boldsymbol{\theta}_j) \right\}, \tag{3.1}$$

where w_j and $\boldsymbol{\theta}_j$ are the weight and the model parameter of the jth task; $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m) \in \mathbb{R}^{d\times m}$; $\Omega \subseteq \mathbb{R}^{d\times m}$ encodes the prior knowledge of task relatedness. Below are several examples.

EXAMPLE 3.1 (Independent task learning). A naïve approach is independent task learning which minimizes the m empirical loss functions separately. That is equivalent to (3.1) with $\Omega = \mathbb{R}^{d\times m}$.

EXAMPLE 3.2 (Data pooling). In the other extreme, one may pool all the data together, solve the consensus program $\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\{\sum_{j=1}^m w_j f_j(\boldsymbol{\beta})\}$ and output one estimate for all tasks. We have $\Omega = \{\boldsymbol{\beta}\mathbf{1}_m^\top : \boldsymbol{\beta} \in \mathbb{R}^d\}$.

EXAMPLE 3.3 (Clustered MTL). The one-size-fits-all strategy above can be extended to clustered MTL, which handles multiples clusters of similar tasks. One may solve the program

$$\min_{\substack{\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_K\in\mathbb{R}^d\\ z_1,\dots,z_m\in[K]}} \left\{\sum_{j=1}^m w_j f_j(\boldsymbol{\beta}_{z_j})\right\} \tag{3.2}$$

to get the estimated labels $\{\widehat{z}_j\}_{j=1}^m$ and cluster centers $\{\widehat{\boldsymbol{\beta}}_j\}_{j=1}^K$. The estimated model parameters for the m tasks are $\{\widehat{\boldsymbol{\beta}}_{\widehat{z}_j}\}_{j=1}^m$. This method corresponds to $\Omega = \{\boldsymbol{B}\boldsymbol{Z} : \boldsymbol{B} \in \mathbb{R}^{d\times K}, \boldsymbol{Z} \in \{0,1\}^{K\times m}, \boldsymbol{Z}^\top\mathbf{1}_K = \mathbf{1}_m\}$.

EXAMPLE 3.4 (Low-rank MTL). By further relaxing the discrete class indicators in (3.2) to continuous latent variables, one gets a formulation for low-rank MTL

$$\min_{\boldsymbol{B}\in\mathbb{R}^{d\times K},\boldsymbol{Z}\in\mathbb{R}^{K\times m}} \left\{\sum_{j=1}^m w_j f_j(\boldsymbol{B}\boldsymbol{z}_j)\right\}. \tag{3.3}$$

An optimal solution $(\widehat{\boldsymbol{B}}, \widehat{\boldsymbol{Z}})$ yields estimated model parameters $\{\widehat{\boldsymbol{B}}\widehat{\boldsymbol{z}}_j\}_{j=1}^m$ that lie in the range of $\widehat{\boldsymbol{B}}$. We have $\Omega = \{\boldsymbol{B}\boldsymbol{Z} : \boldsymbol{B} \in \mathbb{R}^{d\times K}, \boldsymbol{Z} \in \mathbb{R}^{K\times m}\}$.

EXAMPLE 3.5 (Hard parameter sharing). A popular approach of MTL with neural networks is to learn a network shared by all tasks for feature extraction, plus task-specific linear functions that map features to final predictions [13]. Thus, models of the m tasks share part of their parameters. It can be viewed as (3.1) with Ω of the form

$$\left\{\begin{pmatrix}\boldsymbol{\beta}\mathbf{1}_m^\top\\ \boldsymbol{\Gamma}\end{pmatrix} : \boldsymbol{\beta} \in \mathbb{R}^{d-K}, \boldsymbol{\Gamma} \in \mathbb{R}^{K\times m}\right\},$$

where K is the number of features, $\boldsymbol{\beta}$ consists of weight parameters of the neural network, and the columns of $\boldsymbol{\Gamma}$ are parameters of task-specific linear functions. This is a combination of independent task learning and data pooling. When the neural network is replaced with a linear transform, it is equivalent to low-rank MTL.

We propose a framework named *Adaptive and Robust MUltitask Learning*, or ARMUL for short: solve an augmented program

$$(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\Gamma}}) \in \mathop{\arg\min}_{\boldsymbol{\Theta}\in\mathbb{R}^{d\times m},\boldsymbol{\Gamma}\in\Omega} \left\{\sum_{j=1}^m w_j\big[f_j(\boldsymbol{\theta}_j) + \lambda_j\|\boldsymbol{\theta}_j - \boldsymbol{\gamma}_j\|_2\big]\right\}, \tag{3.4}$$

and use the columns $\{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^m$ of $\widehat{\boldsymbol{\Theta}}$ as the estimated model parameters for m tasks. Here, $\{\lambda_j\}_{j=1}^m$ are nonnegative regularization parameters. Setting all of λ_j's to zero or infinity result in independent task learning or the constrained program (3.1), respectively. The framework (3.4) is a relaxation of (3.1) so that the estimated models better fit their associated data. The method (2.1) for multitask mean estimation is a special case, with $d = 1$, $\Omega = \{\theta \mathbf{1}_m^\top : \theta \in \mathbb{R}\}$ and f_j being the square loss.

REMARK 2 (Relaxation). One could also consider the following relaxation of (3.1):

$$\min_{\boldsymbol{\Theta}\in\Omega_r} \left\{ \sum_{j=1}^m w_j f_j(\boldsymbol{\theta}_j) \right\}, \tag{3.5}$$

where

$$\Omega_r = \left\{ \boldsymbol{\Theta} \in \mathbb{R}^{d\times m} : \exists \boldsymbol{\Gamma} \in \Omega \text{ s.t. } \sum_{j=1}^m w_j \lambda_j \|\boldsymbol{\theta}_j - \boldsymbol{\gamma}_j\|_2 \le r \right\}.$$

The programs (3.5) and (3.1) share the same form. Choosing a positive r helps deal with possible misspecification of the space Ω for the true parameter $\boldsymbol{\Theta}^\star = (\boldsymbol{\theta}_1^\star, \dots, \boldsymbol{\theta}_m^\star)$. Also, there exists some $r \ge 0$ such that the constrained program (3.5) is equivalent to the penalized program (3.4). Selecting r and $\{\lambda_j\}_{j=1}^m$ for (3.5) can be difficult when the amount of misspecification is unknown. On the other hand, our theory shows that (3.4) enjoys strong guarantees while being agnostic to the misspecification.

We see from (3.4) that $\widehat{\boldsymbol{\Gamma}}$ solves a constrained problem $\min_{\boldsymbol{\Gamma}\in\Omega}\{\sum_{j=1}^m w_j \widetilde{f}_j(\boldsymbol{\gamma}_j)\}$ similar to (3.1), where $\widetilde{f}_j(\boldsymbol{\gamma}) = \min_{\boldsymbol{\xi}\in\mathbb{R}^d}\{f_j(\boldsymbol{\xi}) + \lambda_j\|\boldsymbol{\gamma} - \boldsymbol{\xi}\|_2\}$ is the infimal convolution of the loss function $f_j(\cdot)$ and the ℓ_2 penalty $\lambda_j\|\cdot\|_2$. Since the latter is λ_j-Lipschitz, as long as f_j is convex, the infimal convolution $\widetilde{f}_j$ is always convex and λ_j-Lipschitz (Lemma F.4 in the Supplementary Material [24]) just like the Huber loss function in Lemma 2.1. This makes our method robust against a small fraction of tasks which are dissimilar to others or even contain contaminated data. Meanwhile, the fact

$$\widehat{\boldsymbol{\theta}}_j \in \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d}\{f_j(\boldsymbol{\theta}) + \lambda_j\|\boldsymbol{\theta} - \widehat{\boldsymbol{\gamma}}_j\|_2\} \quad \forall j \in [m] \tag{3.6}$$

shows that $\widehat{\boldsymbol{\Theta}}$ is shrunk toward $\widehat{\boldsymbol{\Gamma}} \in \Omega$. When the set Ω accurately reflects the relations among m underlying models and λ_j is not too small, the cusp of the ℓ_2 norm penalty at zero forces $\widehat{\boldsymbol{\Theta}} = \widehat{\boldsymbol{\Gamma}} \in \Omega$. When λ_j is not too large and f_j is strongly convex near its minimizer, the Lipschitz smoothness of the ℓ_2 penalty ensures the closeness between $\widehat{\boldsymbol{\theta}}_j$ and $\operatorname{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d} f_j(\boldsymbol{\theta})$. Hence, the new method will at least be comparable to independent task learning. In Section 4 and Appendix D, we will conduct a formal analysis of the adaptivity and robustness. The theory suggests choosing $w_j = n_j$ and $\lambda_j \asymp \sqrt{\frac{d+\log m}{n_j}}$ to achieve the goal.

3.2. *Implementations.* For efficient implementation of ARMUL, we define $\boldsymbol{V} = \boldsymbol{\Theta} - \boldsymbol{\Gamma}$ and transform the program (3.4) to a more convenient form

$$\min_{\boldsymbol{V}\in\mathbb{R}^{d\times m}, \boldsymbol{\Gamma}\in\Omega} \left\{ \sum_{j=1}^m w_j \big[f_j(\boldsymbol{\gamma}_j + \boldsymbol{v}_j) + \lambda_j\|\boldsymbol{v}_j\|_2\big] \right\}. \tag{3.7}$$

We will optimize the two blocks of variables $\boldsymbol{V}$ and $\boldsymbol{\Gamma}$ in an alternating manner. Assume that $\{f_j\}_{j=1}^m$ are differentiable. If $\boldsymbol{\Gamma}$ is fixed, (3.7) decomposes into m independent programs

$$\min_{\boldsymbol{v}_j\in\mathbb{R}^d} \{f_j(\boldsymbol{\gamma}_j + \boldsymbol{v}_j) + \lambda_j\|\boldsymbol{v}_j\|_2\}, \quad j \in [m]. \tag{3.8}$$

```
Algorithm 1: Adaptive and robust multitask learning (ARMUL)
  Input: loss functions $\{f_j\}_{j=1}^m$, weights $\{w_j\}_{j=1}^m$, penalty parameters $\{\lambda_j\}_{j=1}^m$, step-size
  $\eta_v$, number of iterations $T$, initial guesses $\boldsymbol{V}^0 \in \mathbb{R}^{d\times m}$ and $\boldsymbol{\Gamma}^0 \in \Omega$.
  For $t = 0, 1, \dots, T-1$
      Compute $\boldsymbol{V}^{t+1}$ by
      $\boldsymbol{v}_j^{t+1} = \left(1 - \frac{\eta_v \lambda_j}{\|\boldsymbol{v}_j^t - \eta_v \nabla f_j(\boldsymbol{\gamma}_j + \boldsymbol{v}_j^t)\|_2}\right)_+ (\boldsymbol{v}_j^t - \eta_v \nabla f_j(\boldsymbol{\gamma}_j^t + \boldsymbol{v}_j^t)), \quad j \in [m].$
      Compute $\boldsymbol{\Gamma}^{t+1}$.
  Return: $\widehat{\boldsymbol{\Theta}} = \boldsymbol{\Gamma}^T + \boldsymbol{V}^T$.
```

A natural algorithm for handling nonsmooth convex regularizers such as $\|\cdot\|_2$ is proximal gradient descent [56]. The iteration for solving (3.8) is

$$\boldsymbol{v}_j^{t+1} = \operatorname{prox}_{\eta\lambda_j}(\boldsymbol{v}_j^t - \eta \nabla f_j(\boldsymbol{\gamma}_j + \boldsymbol{v}_j^t)), \quad t = 0, 1, \dots, \tag{3.9}$$

where η is the step-size and we define $\operatorname{prox}_c(\boldsymbol{x}) = (1 - \frac{c}{\|\boldsymbol{x}\|_2})_+ \boldsymbol{x}$. If $\boldsymbol{V}$ is fixed, (3.7) reduces to a constrained program

$$\min_{\boldsymbol{\Gamma}\in\Omega} \left\{ \sum_{j=1}^m w_j f_j(\boldsymbol{\gamma}_j + \boldsymbol{v}_j) \right\} \tag{3.10}$$

of the form (3.1) with shifted loss functions. We will choose algorithms according to Ω. The whole procedure above is summarized in Algorithm 1. For simplicity, we only perform a single iteration of proximal gradient descent. Numerical experiments show that this already gives satisfactory results.

Having introduced the general procedure, we now focus on three important cases of ARMUL (3.4) and derive the updating rules for their $\boldsymbol{\Gamma}$'s. Their Python implementations are available at https://github.com/kw2934/ARMUL/.

1. Vanilla ARMUL: $\Omega = \{\boldsymbol{\beta}\mathbf{1}_m^\top : \boldsymbol{\beta} \in \mathbb{R}^d\}$. The original program (3.4) is equivalent to

$$\min_{\boldsymbol{\Theta}\in\mathbb{R}^{d\times m}, \boldsymbol{\beta}\in\mathbb{R}^d} \left\{ \sum_{j=1}^m w_j [f_j(\boldsymbol{\theta}_j) + \lambda_j \|\boldsymbol{\theta}_j - \boldsymbol{\beta}\|_2] \right\}. \tag{3.11}$$

   It is jointly convex in $(\boldsymbol{\Theta}, \boldsymbol{\beta})$ as long as $\{f_j\}_{j=1}^m$ are convex functions. The intermediate program (3.10) is equivalent to an unconstrained one

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \left\{ \sum_{j=1}^m w_j f_j(\boldsymbol{\beta} + \boldsymbol{v}_j) \right\}.$$

   We can update $\boldsymbol{\beta}$ by gradient descent.
2. Clustered ARMUL: $\Omega = \{\boldsymbol{B}\boldsymbol{Z} : \boldsymbol{B} \in \mathbb{R}^{d\times K}, \boldsymbol{Z} \in \{0,1\}^{K\times m}, \boldsymbol{Z}^\top \mathbf{1}_K = \mathbf{1}_m\}$. The original program (3.4) is equivalent to

$$\min_{\boldsymbol{\Theta}\in\mathbb{R}^{d\times m}, \boldsymbol{B}\in\mathbb{R}^{d\times K}, \boldsymbol{z}\in[K]^m} \left\{ \sum_{j=1}^m w_j [f_j(\boldsymbol{\theta}_j) + \lambda_j \|\boldsymbol{\theta}_j - \boldsymbol{\beta}_{z_j}\|_2] \right\}. \tag{3.12}$$

   The intermediate program (3.10) is equivalent to

$$\min_{\boldsymbol{B}\in\mathbb{R}^{d\times K}, \boldsymbol{z}\in[K]^m} \left\{ \sum_{j=1}^m w_j f_j(\boldsymbol{\beta}_{z_j} + \boldsymbol{v}_j) \right\}.$$

When z is fixed, we update $\boldsymbol{B}$ by gradient descent; when $\boldsymbol{B}$ is fixed, we update z with its optimal value

$$\bigl(\operatorname*{arg\,min}_{z\in[K]} f_1(\boldsymbol{\beta}_z+\boldsymbol{v}_1),\dots,\operatorname*{arg\,min}_{z\in[K]} f_m(\boldsymbol{\beta}_z+\boldsymbol{v}_m)\bigr).$$

We can repeat the above steps multiple times.

3. Low-rank ARMUL: $\Omega=\{\boldsymbol{BZ}:\boldsymbol{B}\in\mathbb{R}^{d\times K},\boldsymbol{Z}\in\mathbb{R}^{K\times m}\}$. The original program (3.4) is equivalent to

$$\min_{\boldsymbol{\Theta}\in\mathbb{R}^{d\times m},\boldsymbol{B}\in\mathbb{R}^{d\times K},\boldsymbol{Z}\in\mathbb{R}^{K\times m}}\left\{\sum_{j=1}^m w_j\bigl[f_j(\boldsymbol{\theta}_j)+\lambda_j\|\boldsymbol{\theta}_j-\boldsymbol{B}\boldsymbol{z}_j\|_2\bigr]\right\}. \tag{3.13}$$

The intermediate program (3.10) is equivalent to

$$\min_{\boldsymbol{B}\in\mathbb{R}^{d\times K},\boldsymbol{Z}\in\mathbb{R}^{K\times m}}\left\{\sum_{j=1}^m w_j f_j(\boldsymbol{B}\boldsymbol{z}_j+\boldsymbol{v}_j)\right\}.$$

When $\boldsymbol{B}$ or $\boldsymbol{Z}$ is fixed, we update the other by gradient descent. Again, the procedure can be repeated.

Algorithm 1 returns the estimated model parameters $\{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^m\subseteq\mathbb{R}^d$ for m tasks. As a byproduct, vanilla ARMUL yields a center $\widehat{\boldsymbol{\beta}}\in\mathbb{R}^d$; clustered ARMUL yields K centers $\{\widehat{\boldsymbol{\beta}}_k\}_{k=1}^K\subseteq\mathbb{R}^d$ together with m cluster labels $\{\widehat{z}_j\}_{j=1}^m\subseteq[K]$; low-rank ARMUL yields a K-dimensional subspace Range$(\widehat{\boldsymbol{B}})\subseteq\mathbb{R}^d$ and m coefficient vectors $\{\widehat{\boldsymbol{z}}_j\}_{j=1}^m\subseteq\mathbb{R}^K$. These quantities reveal intrinsic structures of the task population: the model parameters concentrate around one point, multiple points or a low-dimensional linear subspace. Such knowledge is valuable for dealing with new tasks of similar types.

4. Theoretical analysis. In this section, we conduct a nonasymptotic analysis of vanilla, clustered and low-rank ARMUL algorithms. Our theoretical investigation shows that the proposed estimators automatically adapt to the unknown task relatedness. The study under statistical settings is built upon the deterministic results in Appendix A, which could be of independent interest.

4.1. *Problem setup.* Recall the setup in Section 3.1 where $\{\mathcal{P}_j\}_{j=1}^m$ are probability distributions over sample spaces $\{\mathcal{X}_j\}_{j=1}^m$ and $\{\ell_j\}_{j=1}^m$ are loss functions. We draw m independent datasets $\{\mathcal{D}_j\}_{j=1}^m$, where $\mathcal{D}_j=\{\boldsymbol{\xi}_{ji}\}_{i=1}^{n_j}$ are i.i.d. from $\mathcal{P}_j$. For each j, define the population loss function and its minimizer

$$F_j(\boldsymbol{\theta})=\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{P}_j}\ell_j(\boldsymbol{\theta},\boldsymbol{\xi})\quad\text{and}\quad\boldsymbol{\theta}_j^\star\in\arg\min\nolimits_{\boldsymbol{\theta}\in\mathbb{R}^d}F_j(\boldsymbol{\theta}).$$

Define the jth empirical loss function $f_j(\boldsymbol{\theta})=\frac{1}{n_j}\sum_{i=1}^{n_j}\ell_j(\boldsymbol{\theta},\boldsymbol{\xi}_{ji})$. To facilitate illustration, throughout this section we focus on the case where $n_1=\cdots=n_m=n$. We estimate $\{\boldsymbol{\theta}_j^\star\}_{j=1}^m$ by the solutions $\{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^m$ computed from the program (3.4) with $\lambda_1=\cdots=\lambda_m=\lambda$ and $w_1=\cdots=w_m=1$. We defer discussions on general sample sizes $\{n_j\}_{j=1}^m$ to Appendix D [24].

To analyze the estimation error, we make the following standard assumptions.

ASSUMPTION 4.1 (Regularity). For any $j\in[m]$ and $\boldsymbol{\xi}\in\mathcal{X}_j$, $\ell_j(\cdot,\boldsymbol{\xi}):\mathbb{R}^d\to\mathbb{R}$ is convex and twice differentiable. Also, there exist absolute constants $c_1,c_2>0$ and $c_1<\rho,L,M<c_2$ such that $\rho\boldsymbol{I}\preceq\nabla^2F_j(\boldsymbol{\theta})\preceq L\boldsymbol{I}$ holds for all $\boldsymbol{\theta}\in B(\boldsymbol{\theta}_j^\star,M)$ and $j\in[m]$.

ASSUMPTION 4.2 (Concentration). There exist $0 \le \sigma, \tau, p < c$ for an absolute constant c such that for any $j \in [m]$, we have

$$\|\nabla \ell_j(\boldsymbol{\theta}_j^\star, \boldsymbol{\xi}_{j1})\|_{\psi_2} \le \sigma,$$
$$\|\langle(\nabla^2 \ell_j(\boldsymbol{\theta}, \boldsymbol{\xi}_{j1}) - \mathbb{E}[\nabla^2 \ell_j(\boldsymbol{\theta}, \boldsymbol{\xi}_{j1})])\boldsymbol{v}, \boldsymbol{v}\rangle\|_{\psi_1} \le \tau^2 \quad \forall \boldsymbol{\theta} \in B(\boldsymbol{\theta}_j^\star, M), \boldsymbol{v} \in \mathbb{S}^{d-1},$$
$$\mathbb{E} Q_j(\boldsymbol{\xi}_{j1}) \le \tau^3 d^p,$$

where we define

$$Q_j(\boldsymbol{\xi}) = \sup_{\substack{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in B(\boldsymbol{\theta}_j^\star, M) \\ \boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2}} \frac{\|\nabla^2 \ell_j(\boldsymbol{\theta}_2, \boldsymbol{\xi}) - \nabla^2 \ell_j(\boldsymbol{\theta}_1, \boldsymbol{\xi})\|_2}{\|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\|_2} \quad \forall \boldsymbol{\xi} \in \mathcal{X}_j.$$

The gradients of ℓ_j are taken with respect to its first argument.

The regularity assumption requires the Hessian of the population loss function F_j to be bounded from below and above near its minimizer $\boldsymbol{\theta}_j^\star$. The concentration assumption implies light tails and smoothness of the empirical gradient and Hessian. They are commonly used in statistical machine learning; see [52] and the references therein. Below we present several examples for illustration.

EXAMPLE 4.1 (Gaussian mean estimation). Let $\mathcal{X}_j = \mathbb{R}^d$, $\mathcal{P}_j = N(\boldsymbol{\theta}_j^\star, \boldsymbol{I}_d)$ and $\ell_j(\boldsymbol{\theta}, \boldsymbol{\xi}) = \|\boldsymbol{\xi} - \boldsymbol{\theta}\|_2^2$. Then $\nabla \ell_j(\boldsymbol{\theta}, \boldsymbol{\xi}) = 2(\boldsymbol{\theta} - \boldsymbol{\xi})$ and $\nabla^2 \ell_j(\boldsymbol{\theta}, \boldsymbol{\xi}) = 2\boldsymbol{I}_d$. Assumptions 4.1 and 4.2 clearly hold.

EXAMPLE 4.2 (Linear regression). Let $\boldsymbol{\xi}_{ji} = (\boldsymbol{x}_{ji}, y_{ji}) \in \mathbb{R}^d \times \mathbb{R}$, where $\boldsymbol{x}_{ji}$ is the covariate vector and y_{ji} is the response. Consider the square loss $\ell_j(\boldsymbol{\theta}, (\boldsymbol{x}, y)) = (y - \boldsymbol{x}^\top \boldsymbol{\theta})^2$ and let $\varepsilon_{ji} = y_{ji} - \boldsymbol{x}_{ji}^\top \boldsymbol{\theta}_j^\star$ be the residual of the best linear prediction. Then $\nabla \ell_j(\boldsymbol{\theta}, (\boldsymbol{x}, y)) = 2\boldsymbol{x}(\boldsymbol{x}^\top \boldsymbol{\theta} - y)$ and $\nabla^2 \ell_j(\boldsymbol{\theta}, (\boldsymbol{x}, y)) = 2\boldsymbol{x}\boldsymbol{x}^\top$. Assumption 4.1 holds when the eigenvalues of $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_j}(\boldsymbol{x}\boldsymbol{x}^\top)$ are bounded from above and below. Note that $\nabla \ell_j(\boldsymbol{\theta}_j^\star, (\boldsymbol{x}_{ji}, y_{ji})) = -2\boldsymbol{x}_{ji}\varepsilon_{ji}$. If $\boldsymbol{x}_{ji}$ is sub-Gaussian and ε_{ji} is bounded, then Assumption 4.2 holds. It is worth pointing out that most of our results continue to hold up to logarithmic factors when ε_{ji} is unbounded but light-tailed.

EXAMPLE 4.3 (Logistic regression). Let $\boldsymbol{\xi}_{ji} = (\boldsymbol{x}_{ji}, y_{ji}) \in \mathbb{R}^d \times \{0, 1\}$, where $\boldsymbol{x}_{ji}$ is the covariate vector and y_{ji} is the binary label. Define the logistic loss $\ell_j(\boldsymbol{\theta}, (\boldsymbol{x}, y)) = b(\boldsymbol{x}^\top \boldsymbol{\theta}) - y\boldsymbol{x}^\top \boldsymbol{\theta}$ where $b(t) = \log(1 + e^t)$. We have $\nabla \ell_j(\boldsymbol{\theta}, (\boldsymbol{x}, y)) = \boldsymbol{x}[b'(\boldsymbol{x}^\top \boldsymbol{\theta}) - y]$, $\nabla^2 \ell_j(\boldsymbol{\theta}, (\boldsymbol{x}, y)) = b''(\boldsymbol{x}^\top \boldsymbol{\theta})\boldsymbol{x}\boldsymbol{x}^\top$, $b'(t) = 1/(1 + e^{-t}) \in [0, 1]$ and $b''(t) = e^t/(1 + e^t)^2 = 1/(2 + e^t + e^{-t}) \in (0, 1/4]$. Hence, $0 \prec \nabla^2 F_j(\boldsymbol{\theta}) \preceq (1/4)\boldsymbol{I}$ for all $\boldsymbol{\theta}$, and Assumption 4.1 easily holds for bounded $\|\boldsymbol{\theta}_j^\star\|_2$ and M. When $\boldsymbol{x}_{ji}$ is sub-Gaussian, so is $\nabla \ell_j(\boldsymbol{\theta}, (\boldsymbol{x}_{ji}, y_{ji}))$; for any $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\boldsymbol{v} \in \mathbb{S}^{d-1}$, $\langle \boldsymbol{v}, \nabla^2 \ell_j(\boldsymbol{\theta}, (\boldsymbol{x}_{ji}, y_{ji}))\boldsymbol{v}\rangle = b''(\boldsymbol{x}_{ji}^\top \boldsymbol{\theta})(\boldsymbol{x}_{ji}^\top \boldsymbol{v})^2$ is subexponential. From $\sup_{t\in\mathbb{R}} |b'''(t)| < \infty$ and

$$\begin{aligned}\|\nabla^2 \ell_j(\boldsymbol{\theta}_2, (\boldsymbol{x}, y)) - \nabla^2 \ell_j(\boldsymbol{\theta}_1, (\boldsymbol{x}, y))\|_2 &= |b''(\boldsymbol{x}^\top \boldsymbol{\theta}_2) - b''(\boldsymbol{x}^\top \boldsymbol{\theta}_1)| \cdot \|\boldsymbol{x}\|_2^2 \\ &\lesssim \|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\|_2 \|\boldsymbol{x}\|_2^3\end{aligned}$$

we obtain that $Q_j(\boldsymbol{x}, y) \le \|\boldsymbol{x}\|_2^3$. According to Remark 2.3 in [34], if $\|\boldsymbol{x}_{ji}\|_{\psi_2} \lesssim 1$, then $\mathbb{E} Q_j(\boldsymbol{x}_{ji}, y_{ji}) \le \mathbb{E}\|\boldsymbol{x}_{ji}\|_2^3 \lesssim d^{3/2}$. Based on the above, Assumption 4.2 holds.

4.2. *Personalization.* Independent task learning estimates each $\boldsymbol{\theta}_j^\star$ by the minimizer $\widetilde{\boldsymbol{\theta}}_j = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} f_j(\boldsymbol{\theta})$ of its associated empirical loss, without referring to other tasks. ARMUL (3.4) with $\lambda_1 = \cdots = \lambda_m = \lambda$ and $w_1 = \cdots = w_m = 1$ also yields one personalized model for each task. Below we show their closeness and provide a way of choosing λ so that ARMUL is at least comparable to independent task learning.

For any constraint set $\Omega \subseteq \mathbb{R}^{d\times m}$, the output $\widehat{\boldsymbol{\theta}}_j$ of ARMUL (3.4) always satisfies (3.6). Therefore, $\widehat{\boldsymbol{\theta}}_j$ and $\widetilde{\boldsymbol{\theta}}_j$ minimize similar functions. The penalty term $\lambda\|\boldsymbol{\theta} - \widehat{\boldsymbol{\gamma}}_j\|_2$ in (3.6) can be viewed as a perturbation added to the objective function f_j. According the following theorem, it can only perturb the minimizer by a limited amount; see Appendix D.1 for stronger results for general $\{n_j\}_{j=1}^m$ and their proof.

THEOREM 4.1 (Personalization). *Let Assumptions 4.1 and 4.2 hold. There exist constants C, C_1 and C_2 such that under the conditions $\lambda < \rho M/4$, $n > C_1 d(\log n)(\log m)$ and $0 \le t < C_2 n/(d\log n)$, the following holds with probability at least $1 - e^{-t}$:*

$$\|\widetilde{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2 \le C\sigma\sqrt{\frac{d+\log m + t}{n}} \quad \textit{and} \quad \|\widehat{\boldsymbol{\theta}}_j - \widetilde{\boldsymbol{\theta}}_j\|_2 \le \frac{2\lambda}{\rho} \quad \forall j \in [m].$$

The distance between the estimates $\widehat{\boldsymbol{\theta}}_j$ and $\widetilde{\boldsymbol{\theta}}_j$ returned by ARMUL and independent task learning is bounded using the penalty level λ and the strong convexity parameter ρ. Intuitively, when the empirical loss function f_j is strongly convex in a neighborhood of its minimizer $\widetilde{\boldsymbol{\theta}}_j$, the Lipschitz penalty function does make much difference. The unsquared ℓ_2 penalty is crucial. In Lemma 2.1, we showed this phenomenon for mean estimation in one dimension, where the ℓ_2 penalty becomes the absolute value. Theorem 4.1 guarantees the fidelity of ARMUL outputs to their associated data sets for general M-estimation.

By Assumptions 4.1 and 4.2, we have $\sigma, \rho^{-1} \lesssim 1$. Theorem 4.1 implies that when $\lambda \lesssim \sqrt{\frac{d+\log m}{n}}$, the bound $\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2 \lesssim \sqrt{\frac{d+\log m}{n}}$ simultaneously holds for all $j \in [m]$ with high probability. In that case, the ARMUL achieves the same parametric error rate $O(\sqrt{\frac{d+\log m}{n}})$ of independent task learning. The $\log m$ term results from the simultaneous control over m tasks.

The above results on personalization hold for general ARMUL with arbitrary constraint set Ω. In the subsections to follow, we will investigate three important cases of ARMUL (vanilla, clustered and low-rank) to study the adaptivity and robustness.

4.3. *Vanilla ARMUL.* In this subsection, we analyze the vanilla ARMUL estimators $\{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^m$ returned by

$$(4.1) \qquad (\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\beta}}) \in \mathop{\arg\min}_{\boldsymbol{\Theta}\in\mathbb{R}^{d\times m}, \boldsymbol{\beta}\in\mathbb{R}^d} \left\{ \sum_{j=1}^m \left[f_j(\boldsymbol{\theta}_j) + \lambda\|\boldsymbol{\theta}_j - \boldsymbol{\beta}\|_2 \right] \right\}.$$

We introduce an assumption on task relatedness. It is a multivariate extension of Assumption 2.1.

ASSUMPTION 4.3 (Task relatedness). For any $\varepsilon \in [0,1]$ and $\delta \ge 0$, define

$$\Omega(\varepsilon, \delta) = \left\{ \boldsymbol{\Theta} \in \mathbb{R}^{d\times m} : \min_{\theta\in\mathbb{R}^d} \max_{j\in S} |\boldsymbol{\theta}_j - \boldsymbol{\theta}| \le \delta \text{ and } |S^c|/m \le \varepsilon \text{ for some } S \subseteq [m] \right\}.$$

Assume that $\boldsymbol{\Theta}^\star \in \Omega(\varepsilon, \delta)$ holds for some $\varepsilon, \delta \ge 0$. Let S be a subset of $[m]$ that satisfies the requirements in the definition.

When Assumption 4.3 holds, we say the m tasks are (ε, δ)-related. It is worth pointing out that any m tasks are $(0, \max_{j\in[m]} \|\boldsymbol{\theta}_j^\star\|_2)$-related. Smaller ε and δ imply stronger similarity among the tasks. The theorem below presents upper bounds on estimation errors of vanilla ARMUL (4.1); see Appendix D.2 for stronger results for general $\{n_j\}_{j=1}^m$ and their proof.

THEOREM 4.2 (Vanilla ARMUL). *Let Assumptions 4.1, 4.2 and 4.3 hold. There exist positive constants $\{C_i\}_{i=0}^5$ such that under the conditions $n > C_1 d(\log n)(\log m)$, $0 \le t < C_2 n/(d\log n)$, $C_3\sigma\sqrt{\frac{d+\log m+t}{n}} < \lambda < C_4\sigma$ and $0 \le \varepsilon < C_5$, the following bounds hold with probability at least $1 - e^{-t}$:*

$$\max_{j\in S}\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2 \le C_0\left(\sigma\sqrt{\frac{d+t}{mn}} + \min\{\delta, \lambda\} + \varepsilon\lambda\right),$$

$$\max_{j\in S^c}\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2 \le C_0\lambda,$$

$$\frac{1}{m}\sum_{j=1}^m[F_j(\widehat{\boldsymbol{\theta}}_j) - F_j(\boldsymbol{\theta}_j^\star)] \le \frac{L}{m}\sum_{j=1}^m\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2^2 \le C_0 L\left(\sigma^2\frac{d+t}{mn} + \min\{\delta^2, \lambda^2\} + \varepsilon\lambda^2\right).$$

Moreover, there exists a constant C_6 such that under the conditions $\varepsilon = 0$ and $\delta < C_6\sigma\sqrt{\frac{d+\log m}{n}}$, we have $\widehat{\boldsymbol{\theta}}_1 = \cdots = \widehat{\boldsymbol{\theta}}_m = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\{\sum_{j=1}^m f_j(\boldsymbol{\theta})\}$ with probability at least $1 - e^{-t}$.

Theorem 4.2 simultaneously controls the estimation errors for all individual tasks. This implies the bounds on the MSE $\frac{1}{m}\sum_{j=1}^m \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2^2$ and the average excess risk $\frac{1}{m}\sum_{j=1}^m[F_j(\widehat{\boldsymbol{\theta}}_j) - F_j(\boldsymbol{\theta}_j^\star)]$. The results suggest choosing $\lambda = C\sqrt{\frac{d+\log m}{n}}$ for some constant C. In practice, C can be selected by cross-validation to optimize the performance. When $\lambda \asymp \sqrt{\frac{d+\log m}{n}}$, the MSE bound reads

$$\frac{1}{m}\sum_{j=1}^m\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2^2 \lesssim \frac{d}{mn} + \min\left\{\delta^2, \frac{d}{n}\right\} + \frac{\varepsilon d}{n}, \tag{4.2}$$

where $\lesssim$ hides logarithmic factors.

For any ε and δ, a simple bound $\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2 \lesssim \lambda$ always holds for all $j \in [m]$, which echoes Theorem 4.1. Theorem 4.2 implies more refined results.

- (Reduction to data pooling) When $\varepsilon = \delta = 0$, all target parameters are the same. The parameter space becomes $\Omega(0, 0) = \{\boldsymbol{\beta}\mathbf{1}_m^\top : \boldsymbol{\beta} \in \mathbb{R}^d\}$. Data pooling is a natural approach, whose MSE is $O(d/mn)$. According to (4.2), the vanilla ARMUL has the same error rate. In fact, it coincides with data pooling with high probability, thanks to the cusp of the unsquared ℓ_2 penalty at zero.
- (Adaptivity) The relatedness parameters ε and δ quantify the amount of model misspecification incurred in data pooling. As ε and δ increase, the MSE upper bound (4.2) smoothly transits from that for data pooling to that for independent task learning. We will see in Theorem 4.3 below that for every (ε, δ), the error bound is minimax optimal over $\Omega(\varepsilon, \delta)$. Therefore, vanilla ARMUL automatically adapts to the unknown relatedness (ε, δ) of the tasks. Meanwhile, we need an estimate on the noise level σ. Since σ is determined by individual tasks rather than their relatedness, it is easy to estimate using traditional independent task learning methods. We also note that knowledge about the noise level is commonly assumed in adaptive statistical estimation, including adaptation to smoothness in nonparametric regression [44] and adaptation to sparsity in high-dimensional estimation [22].

- (Robustness) Vanilla ARMUL only pays a limited price $\frac{\varepsilon d}{n}$ for the outlier tasks with unknown index set S^c and arbitrary difference from the others. For the Gaussian mean problem (Example 4.1) with $\delta = 0$, our bounds on $\max_{j \in S} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$ and $\frac{1}{m}\sum_{j=1}^m \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2^2$ recover Theorem 6 in [18]. In addition, we can allow the data sets $\{\mathcal{D}_j\}_{S^c}$ to be arbitrarily contaminated, in which case the bound on $\max_{j \in S} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$ in Theorem 4.2 continues to hold.

To close this subsection, we use multitask Gaussian mean estimation to get minimax lower bounds on the MSE. The proof can be found in Appendix E.1 [24].

THEOREM 4.3 (Minimax lower bound). *Consider the setup in Example 4.1 and let Assumption 4.3 hold. There exist universal constants $C, c > 0$ such that for any (ε, δ),*

$$\inf_{\widehat{\boldsymbol{\Theta}}} \sup_{\boldsymbol{\Theta}^\star \in \Omega(\varepsilon,\delta)} \mathbb{P}_{\boldsymbol{\Theta}^\star}\left[\frac{1}{m}\sum_{j=1}^m \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2^2 \geq C\left(\frac{d}{mn} + \min\left\{\delta^2, \frac{d}{n}\right\} + \frac{\varepsilon d}{n}\right)\right] \geq c.$$

4.4. *Clustered ARMUL.* In this subsection, we study clustered ARMUL

$$(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{B}}, \widehat{z}) \in \mathop{\arg\min}_{\boldsymbol{\Theta} \in \mathbb{R}^{d \times m}, \boldsymbol{B} \in \mathbb{R}^{d \times K}, z \in [K]^m} \left\{\sum_{j=1}^m [f_j(\boldsymbol{\theta}_j) + \lambda \|\boldsymbol{\theta}_j - \boldsymbol{\beta}_{z_j}\|_2]\right\}. \tag{4.3}$$

Here, $K \geq 2$ is the target number of clusters. Clustered multitask learning works the best when $\{\boldsymbol{\theta}_j^\star\}_{j=1}^m$ concentrate around K well-separated centers. Yet, such regularity conditions are difficult to verify and may not hold in practice. We introduce a relaxed version of that as our technical assumption.

ASSUMPTION 4.4 (Task relatedness). There exist $\varepsilon, \delta \geq 0$, $K \geq 2$, $\{\boldsymbol{\beta}_k^\star\}_{k=1}^K \subseteq \mathbb{R}^d$, $\{z_j^\star\}_{j=1}^m \subseteq [K]$, $S \subseteq [m]$ and absolute constants $c_1, c_2 > 0$ such that the following hold:

- (Similarity) $\max_{j \in S} \|\boldsymbol{\theta}_j^\star - \boldsymbol{\beta}_{z_j^\star}^\star\|_2 \leq \delta$ and $|S^c| \leq \varepsilon m$;
- (Separation) $\min_{k \neq \ell} \|\boldsymbol{\beta}_k^\star - \boldsymbol{\beta}_\ell^\star\|_2 \geq c_1$;
- (Balancedness) $\min_{k \in [K]} |\{j \in [m] : z_j^\star = k\}| \geq c_2 m/K$.

When $\varepsilon = \delta = 0$, the target parameters $\{\boldsymbol{\theta}_j^\star\}_{j=1}^m$ consist of only K distinct points $\{\boldsymbol{\beta}_k^\star\}_{k=1}^K$ with constant separations. Also, there is no vanishingly small cluster. Assumption 4.4 allow for any possible tasks as long as we use large enough δ. For instance, we can take $\boldsymbol{\beta}_k^\star = k\boldsymbol{e}_1$ for all k, $z_j^\star = (j \mod K) + 1$ for all j, $\varepsilon = 0$ and $\delta = K + \max_{j \in [m]} \|\boldsymbol{\theta}_j^\star\|_2$ to make Assumption 4.4 hold.

The theorem below presents upper bounds on estimation errors of clustered ARMUL (4.3) when $\delta = 0$, whose proof is in Appendix E.2 [24].

THEOREM 4.4 (Clustered ARMUL). *Let Assumptions 4.1, 4.2 and 4.4 hold with $\varepsilon = 0$. There exist positive constants $\{C_i\}_{i=0}^5$ such that under the conditions $n > C_1 K d (\log n)(\log m)$, $0 \leq t < C_2 n/(d \log n)$, $C_3 K \sigma \sqrt{\frac{d + \log m + t}{n}} < \lambda < C_4 \sigma$ and $0 \leq \varepsilon < C_5/K^2$, the following bound holds for the estimator $\widehat{\boldsymbol{\Theta}}$ in (4.3) with probability at least $1 - e^{-t}$:*

$$\frac{1}{m}\sum_{j=1}^m [F_j(\widehat{\boldsymbol{\theta}}_j) - F_j(\boldsymbol{\theta}_j^\star)] \leq L \cdot \max_{j \in [m]} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2^2 \leq C_0 L\left(\frac{\sigma^2 K(d+t)}{mn} + \min\{K^2\delta^2, \lambda^2\}\right).$$

In addition, there exists a positive constant C_6 that makes the following holds: when $\delta \leq \frac{C_6 \sigma}{K}\sqrt{\frac{d + \log m}{n}}$, with probability at least $1 - e^{-t}$ there is a permutation τ of $[K]$ such that

- $\widehat{\boldsymbol{\theta}}_j = \widehat{\boldsymbol{\beta}}_{\widehat{z}_j}$ *and* $\widehat{z}_j = \tau(z_j^\star)$ *hold for all* $j \in [m]$;
- $\widehat{\boldsymbol{\beta}}_k = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\{\sum_{j:z_j^\star=\tau^{-1}(k)} f_j(\boldsymbol{\beta})\}$ *hold for all* $k \in [K]$.

Take $\lambda = CK\sigma\sqrt{\frac{d+\log m}{n}}$ for some large constant C. By Theorem 4.4, clustered ARMUL satisfies

$$\max_{j\in[m]} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2 \lesssim \min\left\{\sigma\sqrt{\frac{Kd}{mn}} + K\delta, K\sigma\sqrt{\frac{d+\log m}{n}}\right\}.$$

If $\delta = 0$ and $\{z_j^\star\}_{j=1}^m$ are known, then we should pool all the data in each cluster. Since each cluster has $O(m/K)$ tasks and $O(mn/K)$ samples, the estimation error has order $O(\sigma\sqrt{\frac{Kd}{mn}})$. Clustered ARMUL achieves the same rate without knowing $\{z_j^\star\}_{j=1}^m$. As δ grows from 0 to $+\infty$, the error bound gradually become $O(K\sigma\sqrt{\frac{d+\log m}{n}})$. This is the error rate of independent task learning up to a factor of K and an additive term $\log m$. The theorem also states that when the discrepancy δ is small, all cluster labels $\{z_j^\star\}_{j=1}^m$ are perfectly recovered up to a global permutation. The estimated centers $\{\widehat{\boldsymbol{\beta}}_k\}_{k=1}^K$ minimize empirical losses on pooled data in the corresponding clusters. The final estimates $\{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^m$ coincide with their cluster centers.

When $\varepsilon > 0$, there can be tasks that are arbitrarily different from the others. We can prove that clustered ARMUL with cardinality constraints manages to utilize the task relatedness in a robust way; see Appendix E.3 for formal results including a minimax lower bound.

4.5. *Low-rank ARMUL.* In this subsection, we study the estimators $\{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^m$ returned by low-rank ARMUL

$$(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{B}}, \widehat{z}) \in \mathop{\arg\min}_{\boldsymbol{\Theta}\in\mathbb{R}^{d\times m}, \boldsymbol{B}\in\mathbb{R}^{d\times K}, \boldsymbol{Z}\in\mathbb{R}^{K\times m}} \left\{\sum_{j=1}^m [f_j(\boldsymbol{\theta}_j) + \lambda\|\boldsymbol{\theta}_j - \boldsymbol{B}\boldsymbol{z}_j\|_2]\right\}. \tag{4.4}$$

Here, $K \geq 1$ is the target rank. Ideally, we would adopt low-rank multitask learning when $\{\boldsymbol{\theta}_j^\star\}_{j=1}^m$ span a K-dimensional linear subspace and K is much less than d. In other words, $\boldsymbol{\Theta}^\star = \boldsymbol{B}^\star\boldsymbol{Z}^\star$ holds for some $\boldsymbol{B}^\star \in \mathbb{R}^{d\times K}$ and $\boldsymbol{Z}^\star \in \mathbb{R}^{K\times m}$. To handle possible misspecification of the low-rank model, we introduce the following notion of task relatedness. Here, we denote by $\mathcal{O}_{d,K}$ the set of all $d \times K$ matrices with orthonormal columns.

ASSUMPTION 4.5 (Task relatedness). There exist $\varepsilon, \delta \geq 0$, $K \in \mathbb{Z}_+$, $\boldsymbol{B}^\star \in \mathcal{O}_{d,K}$, $\{\boldsymbol{z}_j^\star\}_{j=1}^m \subseteq \mathbb{R}^K$, $S \subseteq [m]$ and absolute constants $c_1, c_2 > 0$ such that the followings hold:

- (Similarity) $\max_{j\in S} \|\boldsymbol{\theta}_j^\star - \boldsymbol{B}^\star\boldsymbol{z}_j^\star\|_2 \leq \delta$ and $|S^c| \leq \varepsilon m$;
- (Balancedness and signal strength) $\max_{j\in[m]} \|\boldsymbol{z}_j^\star\|_2 \leq c_1$ and $\frac{K}{m}\sum_{j=1}^m \boldsymbol{z}_j^\star\boldsymbol{z}_j^{\star\top} \succeq c_2\boldsymbol{I}_K$.

Note that $\boldsymbol{B}^\star\boldsymbol{Z}^\star = (\boldsymbol{B}^\star\boldsymbol{R})(\boldsymbol{R}^{-1}\boldsymbol{Z}^\star)$ holds for any nonsingular $\boldsymbol{R} \in \mathbb{R}^{K\times K}$. Without loss of generality, in Assumption 4.5 we let $\boldsymbol{B}^\star$ have orthonormal columns. The parameters $\{\boldsymbol{\theta}_j^\star\}_{j\in S}$ are approximated by vectors $\{\boldsymbol{B}^\star\boldsymbol{z}_j^\star\}_{j\in S}$ living in a K-dimensional linear subspace Range$(\boldsymbol{B}^\star)$. The approximation errors are bounded by $\delta/\sqrt{n}$, which can be arbitrarily large. The coefficient vectors $\{\boldsymbol{z}_j^\star\}_{j=1}^m$ are assumed to be uniformly bounded and spread out in all directions. The upper bound $\max_{j\in[m]} \|\boldsymbol{z}_j^\star\|_2 \leq c_1$ and the lower bound $\frac{K}{m}\sum_{j=1}^m \boldsymbol{z}_j^\star\boldsymbol{z}_j^{\star\top} \succeq c_2\boldsymbol{I}_K$ imply that at least a constant fraction of $\boldsymbol{z}_j^\star$'s are bounded away from $\mathbf{0}$.

The following theorem depicts the adaptivity of low-rank ARMUL to the unknown task relatedness; see Appendix E.4 for its proof. Here, we only consider the case $\varepsilon = 0$ and focus on the impact of dissimilarity δ. The general case ($\varepsilon > 0$) is left for future work.

THEOREM 4.5 (Low-rank ARMUL). *Let Assumptions 4.1, 4.2 and 4.5 hold, with $\varepsilon = 0$. There exist positive constants $\{C_i\}_{i=0}^5$ such that under the conditions $n > C_1 K d(\log n)(\log m)$, $0 \le t < C_2 n/(d \log n)$ and $C_3 K\sigma\sqrt{\frac{d+\log m+t}{n}} < \lambda < C_4\sigma$, the following bound holds for the estimator $\widehat{\mathbf{\Theta}}$ in (4.3) with probability at least $1 - e^{-t}$:*

$$\frac{1}{m}\sum_{j=1}^{m}\big[F_j(\widehat{\boldsymbol{\theta}}_j) - F_j(\boldsymbol{\theta}_j^\star)\big] \le L \max_{j\in[m]}\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2^2$$
$$\le C_0 K^2\bigg(\frac{\sigma^2 d}{mn} + \frac{\sigma^2(1+\log m + t)}{n} + \min\{\delta^2, \lambda^2/K^2\}\bigg).$$

In addition, there exists a positive constant C_6 such that when $\delta \le \frac{C_6\sigma}{K}\sqrt{\frac{d+\log m}{n}}$, $\widehat{\mathbf{\Theta}} = \widehat{\boldsymbol{B}}\widehat{\boldsymbol{Z}}$ holds with probability at least $1 - e^{-t}$.

Suppose that K is bounded and take $\lambda = C\sigma\sqrt{\frac{d+\log m}{n}}$ for some large constant C. By Theorem 4.5, low-rank ARMUL satisfies

$$\max_{j\in[m]}\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2 \lesssim \min\bigg\{\sigma\sqrt{\frac{d}{mn}} + \sigma\sqrt{\frac{1}{n}} + \delta, \sigma\sqrt{\frac{d}{n}}\bigg\} \tag{4.5}$$

with high probability. We provide a matching minimax lower bound in Appendix E.4 [24]. Again, the error rate never exceeds that for independent task learning. When δ is small, low-rank ARMUL adapts to the task relatedness. Note that $O(\sigma\sqrt{\frac{d}{mn}} + \frac{\sigma}{\sqrt{n}})$ is the best rate one can achieve when the low-rank model is true ($\delta = 0$). In that case, the unknown matrix $\mathbf{\Theta}^\star = \boldsymbol{B}^\star\boldsymbol{Z}^\star$ has $O(d+m)$ unknown parameters. The mn samples imply an error bound $O(\sigma\sqrt{\frac{d+m}{mn}}) = O(\sigma\sqrt{\frac{d}{mn}} + \frac{\sigma}{\sqrt{n}})$. The two terms can be viewed as estimation errors of bases $\boldsymbol{B}^\star$ and coefficients $\boldsymbol{Z}^\star$, respectively.

5. Numerical experiments.

We conduct simulations to verify our theories and real data experiments to test the efficacy our proposed approaches. Our implementations of ARMUL follow the description in Section 3.2. The code and all numerical results are available at `https://github.com/kw2934/ARMUL/`.

5.1. *Simulations.*

We generate synthetic data for multitask linear regression. Throughout our simulations, the number of tasks is $m = 30$. For any $j \in [m]$, the data set $\mathcal{D}_j$ consists of $n = 200$ samples $\{(\boldsymbol{x}_{ji}, y_{ji})\}_{i=1}^n$. The covariate vectors $\{\boldsymbol{x}_{ji}\}_{(i,j)\in[n]\times[m]}$ are i.i.d. $N(\mathbf{0}, \boldsymbol{I}_d)$ with $d = 50$, given which we sample each response $y_{ji} = \boldsymbol{x}_{ji}^\top\boldsymbol{\theta}_j^\star + \varepsilon_{ji}$ from a linear model with noise term $\varepsilon_{ji} \sim N(0,1)$ being independent of the covariates. To study vanilla, clustered and low-rank ARMUL, we determine the coefficient vectors $\{\boldsymbol{\theta}_j^\star\}_{j=1}^m$ in three different ways. The parameters ε and δ below characterize task relatedness, similar to those in Assumptions 4.3, 4.4 and 4.5. Below we write $r\mathbb{S}^{d-1}$ as a shorthand notation for the sphere $\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 = r\}$:

1. Vanilla case

   - Data generation: Set $\boldsymbol{\beta}^\star = 2\boldsymbol{e}_1$, sample i.i.d. random vectors $\{\boldsymbol{\delta}_j\}_{j=1}^m$ uniformly from the sphere $\delta\mathbb{S}^{d-1}$ and set $\boldsymbol{\theta}_j^\star = \boldsymbol{\beta}^\star + \boldsymbol{\delta}_j$ for all $j \in [m]$. Next, draw $\lceil\varepsilon m\rceil$ elements $\{j_s\}_{s=1}^{\lceil\varepsilon m\rceil}$ uniformly at random from $[m]$ without replacement. Replace $\{\boldsymbol{\theta}_{j_s}^\star\}_{s=1}^{\lceil\varepsilon m\rceil}$ with i.i.d. random vectors from $2\mathbb{S}^{d-1}$. Denote by $S = [m]\backslash\{j_1, \dots, j_{\lceil\varepsilon m\rceil}\}$.

- Methods for comparison: vanilla ARMUL (4.1), independent task learning (Example 3.1) and data pooling (Example 3.2).

2. Clustered case

- Set $K = 3$, $\boldsymbol{\beta}_k^\star = 2\boldsymbol{e}_k$ for $k \in [K]$ and $z_j^\star = (j \mod K) + 1$ for $j \in [m]$. Sample i.i.d. random vectors $\{\boldsymbol{\delta}_j\}_{j=1}^m$ uniformly from the sphere $\delta\mathbb{S}^{d-1}$ and set $\boldsymbol{\theta}_j^\star = \boldsymbol{\beta}_{z_j^\star}^\star + \boldsymbol{\delta}_j$ for all $j \in [m]$. Replace an ε-fraction of the coefficient vectors by the corresponding procedure in the vanilla case.
- Methods for comparison: clustered ARMUL (4.3), clustered MTL (Example 3.3), independent task learning (Example 3.1) and data pooling (Example 3.2).

3. Low-rank case

- Set $K = 3$ and $\boldsymbol{B}^\star = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3) \in \mathbb{R}^{m\times K}$. Samples $\{\boldsymbol{z}_j^\star\}_{j=1}^m$ independently from $N(\boldsymbol{0}, \boldsymbol{I}_K)$ and another set of i.i.d. vectors $\{\boldsymbol{\delta}_j\}_{j=1}^m$ uniformly from the sphere $\delta\mathbb{S}^{d-1}$. Let $\boldsymbol{\theta}_j^\star = \boldsymbol{B}^\star\boldsymbol{z}_j^\star + \boldsymbol{\delta}_j$ for all $j \in [m]$. Replace an ε-fraction of the coefficient vectors by the corresponding procedure in the vanilla case.
- Methods for comparison: low-rank ARMUL (4.4), low-rank MTL (Example 3.4), independent task learning (Example 3.1) and data pooling (Example 3.2).

Guided by the theories in Section 4, we set the regularization parameter λ in ARMUL algorithms (4.1), (4.3), (4.4) to be $c\sqrt{d/n}$ and select the optimal preconstant c from $\{0.2, 0.4, 0.6, \dots, 2\}$ by 5-fold cross-validation. Below is how we evaluate the quality of each c:

- Step 1: Randomly partition each data set $\mathcal{D}_j$ into 5 (approximately) equally-sized subsets $\{\mathcal{D}_{j\ell}\}_{\ell=1}^5$.
- Step 2: For $\ell = 1, \dots, 5$, define $\widetilde{\mathcal{D}}_j^{(\ell)} = \bigcup_{s\neq\ell}\mathcal{D}_s$, conduct ARMUL on $\{\widetilde{\mathcal{D}}_j^{(\ell)}\}_{j=1}^m$ with $\lambda = c\sqrt{d/n}$, test the obtained models on $\{\widetilde{\mathcal{D}}_{j\ell}\}_{j=1}^m$.
- Step 3: Get the average of mean squared prediction errors over all tasks.

We vary ε in $\{0, 0.2\}$ and δ in $\{0, 0.1, 0.2, \dots, 1\}$ to obtain tasks with different degrees of relatedness. When $\varepsilon = 0$, we measure the maximum estimation error $\max_{j\in[m]} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$. For $\varepsilon = 0$, we measure the maximum estimation error $\max_{j\in[m]} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$ and its restricted version $\max_{j\in S} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$ on the set S of similar tasks. Figures 1 and 2 demonstrate how the estimation errors grow with the heterogeneity parameter δ. The curves and error bands show the means and standard deviations over 100 independent runs, respectively.

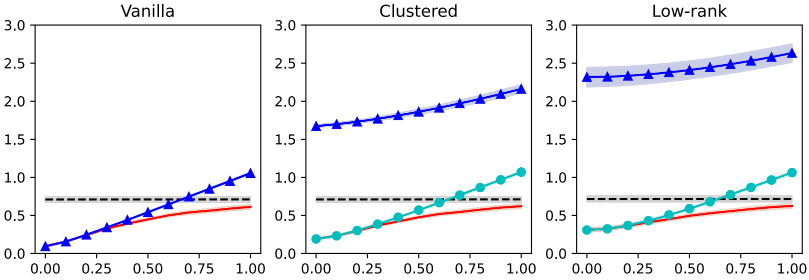

FIG. 1. *Impact of task relatedness when* $\varepsilon = 0$. *From left to right*: *vanilla, clustered and low-rank cases. x-axis*: δ. *y-axis*: $\max_{j\in[m]} \|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$. *Red solid lines*: *ARMUL. Blue triangles*: *data pooling. Black dashed lines*: *independent task learning. Cyan circles*: *clustered MTL* (*middle*) *or low-rank MTL* (*right*).

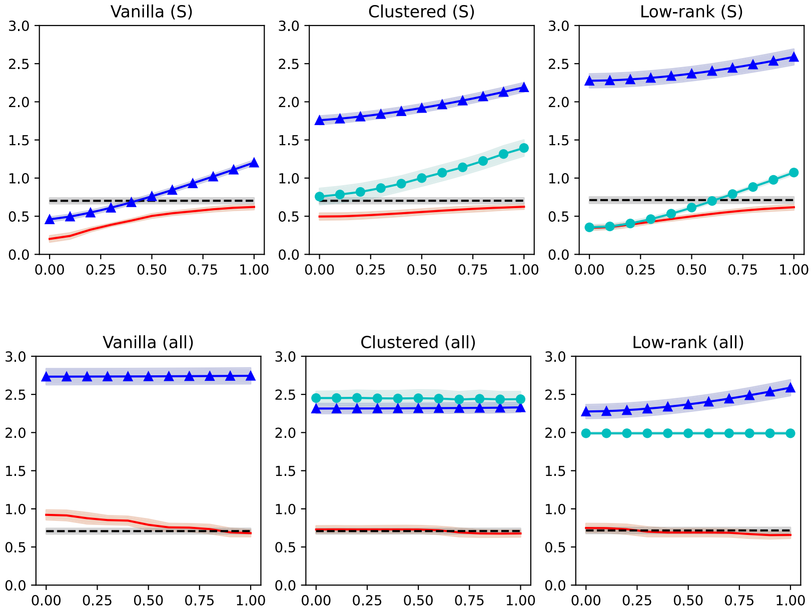

FIG. 2. *Impact of task relatedness when $\varepsilon = 0.2$. From left to right: vanilla, clustered and low-rank cases. x-axis: δ. y-axis: $\max_{j\in S}\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$ (top) or $\max_{j\in[m]}\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$ (bottom). Red solid lines: ARMUL. Blue triangles: data pooling. Black dashed lines: independent task learning. Cyan circles: clustered MTL (middle) or low-rank MTL (right).*

The simulations confirm the adaptivity and robustness of ARMUL methods, as stated in Theorems 4.1, 4.2, 4.4 and 4.5. When $\varepsilon = 0$ and δ is small, the vanilla, clustered and low-rank ARMUL coincide with data pooling, clustered MTL and low-rank MTL, respectively. However, the latter are too rigid and, therefore, deteriorate quickly as δ grows. ARMUL methods, on the other hand, nicely handle model misspecifications and never underperform independent task learning. When ε becomes 0.2, ARMUL methods continue to work well on the set S of similar tasks while data pooling and clustered MTL are badly affected. For the exceptional tasks in S^c, the error curves for $\max_{j\in[m]}\|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^\star\|_2$ in Figure 2 implies that ARMUL methods are still comparable to independent task learning. As we have studied in Theorem 4.1, ARMUL estimators always stay close to the loss minimizers associated to individual tasks. They are generalizations of limited translation estimators [26, 60] to multivariate M-estimation. In contrast, data pooling, clustered MTL and low-rank MTL perform poorly on S^c.

5.2. *Real data.* We evaluate the proposed ARMUL methods on a real-world data set. The Human Activity Recognition (HAR) database is built by [2] from the recordings of 30 volunteers performing activities of daily living while carrying a waist-mounted smartphone with embedded inertial sensors. On average, each volunteer has 343.3 samples (min: 281, max: 409). Each sample corresponds to one of six activities (walking, walking upstairs, walking downstairs, sitting, standing and laying) and has a 561-dimensional feature vector with time and frequency domain variables.

We model each volunteer as a task and aim to distinguish between sitting and the other activities. The problem is therefore formulated as multitask logistic regression with $m = 30$ tasks. We conduct principal component analysis to reduce the dimension to 100. Together with the intercept term, the preprocessed data have $d = 101$ variables in total. We randomly select 20% of the data from each task for testing, and train logistic models on the rest of the data. The sample sizes $\{n_j\}_{j=1}^m$ for training range from 225 to 328. We apply three ARMUL

TABLE 1
Test error rates (in percentage) on the HAR data set

ARMUL			Benchmarks			
Vanilla	Clustered	Low-rank	ITL	Data pooling	Clustered	Low-rank
1.12 (0.25)	**0.84 (0.22)**	**0.80 (0.19)**	1.95 (0.32)	3.48 (0.39)	2.15 (0.33)	1.30 (0.23)

methods (vanilla, clustered and low-rank) and four benchmark approaches (independent task learning, data pooling, clustered MTL and low-rank MTL) to standardized data. For each ARMUL method, we set $w_j = n_j$ and $\lambda_j = c\sqrt{d/n_j}$ in (3.4), as is suggested by our results for general sample sizes (Theorems D.1 and D.2). The constant factor c is chosen from $\{0.05, 0.1, 0.15, \ldots, 0.5\}$ using 5-fold cross-validation. We use the same procedure to select the number of clusters K in clustered methods from $\{2, 3, 4, 5\}$ and the rank K in low-rank methods from $\{1, 2, 3, 4, 5\}$. Finally, we compute the misclassification error on testing data for each method.

Table 1 summarizes the means and standard deviations (in parentheses) of test error rates (in percentage) over 100 independent runs, where ITL stands for independent task learning. The randomness comes from train/test splits and cross-validation. We see that ARMUL methods significantly outperform benchmarks. In addition, we observe several interesting phenomena.

- The tasks are rather heterogeneous, since data pooling and clustered MTL are even worse than independent task learning. As the method becomes more flexible (from data pooling to clustered MTL and then low-rank MTL), the performance gets better. The same trend appears in ARMUL methods as well.
- An ARMUL method augments a basic multitask learning method with models for individual tasks. Such augmentation brings great benefits: even the augmented version of data pooling (i.e., vanilla ARMUL) works better than the raw version of low-rank MTL.

6. Discussions. We introduced a framework for multitask learning named ARMUL that can be used as a wrapper around any multitask learning algorithm of the form (3.1). We analyzed its adaptivity to unknown task relatedness, where the unsquared ℓ_2 penalty function plays a crucial role. We also verified the theories by extensive numerical experiments. We hope that our framework can spur further research in related fields. It would be interesting to develop methods for high-dimensional problems with sparsity or other structures, and build inferential tools for uncertainty quantification. Since heterogeneous data sets are often collected and stored at multiple sites, communication-efficient procedures for distributed statistical inference are desirable. Another direction is to extend our methods to meta-learning, also known as learning to learn [62]. The goal is to extract from existing tasks useful knowledge (e.g., common representation) that facilitates learning future tasks of similar type. Our framework could provide a principled way of dealing with misspecified similarity structure.

Acknowledgment. We are grateful to two anonymous referees and the Associate Editor for their helpful comments. We thank Chen Dan, Dongming Huang, Yuhang Wu and Yichen Zhang for discussions.

Funding. Kaizheng Wang's research is supported by a NSF Grant DMS-2210907 and a startup grant at Columbia University. We acknowledge computing resources from Columbia University's Shared Research Computing Facility project, which is supported by NIH Research Facility Improvement Grant 1G20RR030893-01, and associated funds from the New

York State Empire State Development, Division of Science Technology and Innovation (NYSTAR) Contract C090171, both awarded April 15, 2010. Part of the research was conducted when Yaqi Duan was affiliated with the Laboratory for Information and Decision Systems at Massachusetts Institute of Technology and the Department of Operations Research and Financial Engineering at Princeton University.

SUPPLEMENTARY MATERIAL

Supplement to "Adaptive and robust multitask learning." (DOI: 10.1214/23-AOS2319SUPP; .pdf). Proofs of the results in the paper can be found in the Supplementary Material.

REFERENCES

[1] Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.* **6** 1817–1853. MR2249873

[2] Anguita, D., Ghio, A., Oneto, L., Parra Perez, X. and Reyes Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* 437–442.

[3] Antoniadis, A. (2007). Wavelet methods in statistics: Some recent developments and their applications. *Stat. Surv.* **1** 16–55. MR2520413 https://doi.org/10.1214/07-SS014

[4] Argyriou, A., Evgeniou, T. and Pontil, M. (2008). Convex multi-task feature learning. *Mach. Learn.* **73** 243–272.

[5] Asiaee, A., Oymak, S., Coombes, K. R. and Banerjee, A. (2019). Data enrichment: Multi-task learning in high dimension with theoretical guarantees. In *Adaptive and Multitask Learning Workshop at the ICML*. IMLS, Long Beach, CA.

[6] Balcan, M.-F., Khodak, M. and Talwalkar, A. (2019). Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning* 424–433. PMLR.

[7] Baxter, J. (2000). A model of inductive bias learning. *J. Artificial Intelligence Res.* **12** 149–198. MR1752410 https://doi.org/10.1613/jair.731

[8] Ben-David, S. and Schuller, R. (2003). Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24–27, 2003. Proceedings* 567–580. Springer, Berlin.

[9] Bickel, P. J. (1983). Minimax estimation of the mean of a normal distribution subject to doing well at a point. In *Recent Advances in Statistics* 511–528. Academic Press, New York. MR0736544

[10] Bickel, P. J. (1984). Parametric robustness: Small biases can be worthwhile. *Ann. Statist.* **12** 864–879. MR0751278 https://doi.org/10.1214/aos/1176346707

[11] Breiman, L. and Friedman, J. H. (1997). Predicting multivariate responses in multiple linear regression. *J. Roy. Statist. Soc. Ser. B* **59** 3–54. MR1436554 https://doi.org/10.1111/1467-9868.00054

[12] Cai, T., Liu, M. and Xia, Y. (2022). Individual data protected integrative regression analysis of high-dimensional heterogeneous data. *J. Amer. Statist. Assoc.* **117** 2105–2119. MR4528492 https://doi.org/10.1080/01621459.2021.1904958

[13] Caruana, R. (1997). Multitask learning. *Mach. Learn.* **28** 41–75.

[14] Chen, A., Owen, A. B. and Shi, M. (2015). Data enriched linear regression. *Electron. J. Stat.* **9** 1078–1112. MR3352068 https://doi.org/10.1214/15-EJS1027

[15] Chen, J., Zhou, J. and Ye, J. (2011). Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* 42–50.

[16] Chen, S., Zhang, B. and Ye, T. (2021). Minimax rates and adaptivity in combining experimental and observational data. arXiv preprint. Available at arXiv:2109.10522.

[17] Chen, S., Zheng, Q., Long, Q. and Su, W. J. (2021). A theorem of the alternative for personalized federated learning. arXiv preprint. Available at arXiv:2103.01901.

[18] Collier, O. and Dalalyan, A. S. (2019). Multidimensional linear functional estimation in sparse Gaussian models and robust estimation of the mean. *Electron. J. Stat.* **13** 2830–2864. MR3998929 https://doi.org/10.1214/19-EJS1590

[19] DENEVI, G., CILIBERTO, C., GRAZZI, R. and PONTIL, M. (2019). Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning* 1566–1575. PMLR.

[20] DONOHO, D. and MONTANARI, A. (2016). High dimensional robust M-estimation: Asymptotic variance via approximate message passing. *Probab. Theory Related Fields* **166** 935–969. MR3568043 https://doi.org/10.1007/s00440-015-0675-z

[21] DONOHO, D. L. and JOHNSTONE, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81** 425–455. MR1311089 https://doi.org/10.1093/biomet/81.3.425

[22] DONOHO, D. L., JOHNSTONE, I. M., HOCH, J. C. and STERN, A. S. (1992). Maximum entropy and the nearly black object. *J. Roy. Statist. Soc. Ser. B* **54** 41–81. MR1157714

[23] DU, S. S., HU, W., KAKADE, S. M., LEE, J. D. and LEI, Q. (2020). Few-Shot Learning via Learning the Representation, Provably. In *International Conference on Learning Representations*.

[24] DUAN, Y. and WANG, K. (2023). Supplement to "Adaptive and robust multi-task learning." https://doi.org/10.1214/23-AOS2319SUPP

[25] EFRON, B. and HASTIE, T. (2016). *Computer Age Statistical Inference*: *Algorithms*, *evidence*, *and data science*. *Institute of Mathematical Statistics* (*IMS*) *Monographs* **5**. Cambridge Univ. Press, New York. MR3523956 https://doi.org/10.1017/CBO9781316576533

[26] EFRON, B. and MORRIS, C. (1972). Limiting the risk of Bayes and empirical Bayes estimators. II. The empirical Bayes case. *J. Amer. Statist. Assoc.* **67** 130–139. MR0323015

[27] EFRON, B. and MORRIS, C. (1973). Stein's estimation rule and its competitors—an empirical Bayes approach. *J. Amer. Statist. Assoc.* **68** 117–130. MR0388597

[28] EVGENIOU, T., MICCHELLI, C. A. and PONTIL, M. (2005). Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.* **6** 615–637. MR2249833

[29] EVGENIOU, T. and PONTIL, M. (2004). Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 109–117.

[30] GANNAZ, I. (2007). Robust estimation and wavelet thresholding in partially linear models. *Stat. Comput.* **17** 293–310. MR2409795 https://doi.org/10.1007/s11222-007-9019-x

[31] HANNEKE, S. and KPOTUFE, S. (2022). A no-free-lunch theorem for multitask learning. *Ann. Statist.* **50** 3119–3143. MR4524491 https://doi.org/10.1214/22-aos2189

[32] HIRIART-URRUTY, J.-B. and LEMARÉCHAL, C. (1993). *Convex Analysis and Minimization Algorithms*. *I*: *Fundamentals*. *Grundlehren der Mathematischen Wissenschaften* [*Fundamental Principles of Mathematical Sciences*] **305**. Springer, Berlin. MR1261420

[33] HODGES, J. L. JR. and LEHMANN, E. L. (1952). The use of previous experience in reaching statistical decisions. *Ann. Math. Stat.* **23** 396–407. MR0050240 https://doi.org/10.1214/aoms/1177729384

[34] HSU, D., KAKADE, S. M. and ZHANG, T. (2012). A tail inequality for quadratic forms of subgaussian random vectors. *Electron. Commun. Probab.* **17** no. 52, 6. MR2994877 https://doi.org/10.1214/ECP.v17-2079

[35] HUBER, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Stat.* **35** 73–101. MR0161415 https://doi.org/10.1214/aoms/1177703732

[36] HUBER, P. J. (1981). *Robust Statistics*. *Wiley Series in Probability and Mathematical Statistics*. Wiley, New York. MR0606374

[37] JACOB, L., BACH, F. and VERT, J.-P. (2008). Clustered multi-task learning: A convex formulation. In *Proceedings of the* 21*st International Conference on Neural Information Processing Systems* 745–752.

[38] JALALI, A., RAVIKUMAR, P. and SANGHAVI, S. (2013). A dirty model for multiple sparse regression. *IEEE Trans. Inf. Theory* **59** 7947–7968. MR3142275 https://doi.org/10.1109/TIT.2013.2280272

[39] JAMES, W. and STEIN, C. (1960). Estimation with quadratic loss. In *Proc.* 4*th Berkeley Sympos. Math. Statist. and Prob.*, *Vol. I* 361–379. Univ. California Press, Berkeley-Los Angeles, CA. MR0133191

[40] KE, Z. T., FAN, J. and WU, Y. (2015). Homogeneity pursuit. *J. Amer. Statist. Assoc.* **110** 175–194. MR3338495 https://doi.org/10.1080/01621459.2014.892882

[41] KONSTANTINOV, N., FRANTAR, E., ALISTARH, D. and LAMPERT, C. (2020). On the sample complexity of adversarial multi-source PAC learning. In *International Conference on Machine Learning* 5416–5425. PMLR.

[42] KUMAR, A. and HAL, D. III (2012). Learning task grouping and overlap in multi-task learning. In *Proceedings of the* 29*th International Coference on International Conference on Machine Learning* 1723–1730.

[43] LENZERINI, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* 233–246.

[44] LEPSKIĬ, O. V. (1991). On a problem of adaptive estimation in Gaussian white noise. *Theory Probab. Appl.* **35** 454–466.

[45] LIU, G., LIN, Z. and YU, Y. (2010). Robust subspace segmentation by low-rank representation. In *Proceedings of the* 27*th International Conference on International Conference on Machine Learning* 663–670.

[46] LOUNICI, K., PONTIL, M., VAN DE GEER, S. and TSYBAKOV, A. B. (2011). Oracle inequalities and optimal inference under group sparsity. *Ann. Statist.* **39** 2164–2204. MR2893865 https://doi.org/10.1214/11-AOS896

[47] LOW, M. G. (1997). On nonparametric confidence intervals. *Ann. Statist.* **25** 2547–2554. MR1604412 https://doi.org/10.1214/aos/1030741084

[48] MAITY, S., SUN, Y. and BANERJEE, M. (2019). Meta-analysis of heterogeneous data: Integrative sparse regression in high-dimensions. arXiv preprint. Available at arXiv:1912.11928.

[49] MAURER, A., PONTIL, M. and ROMERA-PAREDES, B. (2016). The benefit of multitask representation learning. *J. Mach. Learn. Res.* **17** Paper No. 81, 32. MR3517104

[50] MCCOY, M. and TROPP, J. A. (2011). Two proposals for robust PCA using semidefinite programming. *Electron. J. Stat.* **5** 1123–1160. MR2836771 https://doi.org/10.1214/11-EJS636

[51] MCDONALD, A. M., PONTIL, M. and STAMOS, D. (2016). New perspectives on k-support and cluster norms. *J. Mach. Learn. Res.* **17** Paper No. 155, 38. MR3555046

[52] MEI, S., BAI, Y. and MONTANARI, A. (2018). The landscape of empirical risk for nonconvex losses. *Ann. Statist.* **46** 2747–2774. MR3851754 https://doi.org/10.1214/17-AOS1637

[53] MOUSAVI KALAN, M., FABIAN, Z., AVESTIMEHR, S. and SOLTANOLKOTABI, M. (2020). Minimax lower bounds for transfer learning with linear and one-hidden layer neural networks. *Adv. Neural Inf. Process. Syst.* **33** 1959–1969.

[54] NEGAHBAN, S. and WAINWRIGHT, M. J. (2008). Joint support recovery under high-dimensional scaling: Benefits and perils of $\ell_{1,\infty}$-regularization. *Adv. Neural Inf. Process. Syst.* **21** 1161–1168.

[55] OBOZINSKI, G., WAINWRIGHT, M. J. and JORDAN, M. I. (2011). Support union recovery in high-dimensional multivariate regression. *Ann. Statist.* **39** 1–47. MR2797839 https://doi.org/10.1214/09-AOS776

[56] PARIKH, N. and BOYD, S. (2014). Proximal algorithms. *Found. Trends Optim.* **1** 127–239.

[57] PONG, T. K., TSENG, P., JI, S. and YE, J. (2010). Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM J. Optim.* **20** 3465–3489. MR2763512 https://doi.org/10.1137/090763184

[58] SHE, Y. and OWEN, A. B. (2011). Outlier detection using nonconvex penalized regression. *J. Amer. Statist. Assoc.* **106** 626–639. MR2847975 https://doi.org/10.1198/jasa.2011.tm10390

[59] SHEN, X. and HUANG, H.-C. (2010). Grouping pursuit through a regularization solution surface. *J. Amer. Statist. Assoc.* **105** 727–739. MR2724856 https://doi.org/10.1198/jasa.2010.tm09380

[60] STEIN, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *Ann. Statist.* **9** 1135–1151. MR0630098

[61] TANG, L. and SONG, P. X. K. (2016). Fused lasso approach in regression coefficients clustering—learning parameter heterogeneity in data integration. *J. Mach. Learn. Res.* **17** Paper No. 113, 23. MR3543519

[62] THRUN, S. and PRATT, L. (2012). *Learning to Learn*. Springer, Berlin.

[63] TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58** 267–288. MR1379242

[64] TIBSHIRANI, R., SAUNDERS, M., ROSSET, S., ZHU, J. and KNIGHT, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **67** 91–108. MR2136641 https://doi.org/10.1111/j.1467-9868.2005.00490.x

[65] TRIPURANENI, N., JORDAN, M. and JIN, C. (2020). On the Theory of Transfer Learning: The Importance of Task Diversity. *Adv. Neural Inf. Process. Syst.* **33**.

[66] WU, S., ZHANG, H. R. and RÉ, C. (2020). Understanding and improving information transfer in multi-task learning. arXiv preprint. Available at arXiv:2005.00944.

[67] XU, H., CARAMANIS, C. and SANGHAVI, S. (2012). Robust PCA via outlier pursuit. *IEEE Trans. Inf. Theory* **58** 3047–3064. MR2952532 https://doi.org/10.1109/TIT.2011.2173156

[68] XU, K. and BASTANI, H. (2021). Learning across bandits in high dimension via robust statistics. arXiv preprint. Available at arXiv:2112.14233.