Unified Transfer Learning in High-Dimensional Linear Regression

Shuo Shuo Liu
Columbia University

Abstract

Transfer learning plays a key role in modern data analysis when: (1) the target data are scarce but the source data are sufficient; (2) the distributions of the source and target data are heterogeneous. This paper develops an interpretable unified transfer learning model, termed as UTrans, which can detect both transferable variables and source data. More specifically, we establish the estimation error bounds and prove that our bounds are lower than those with target data only. Besides, we propose a source detection algorithm based on hypothesis testing to exclude the nontransferable data. We evaluate and compare UTrans to the existing algorithms in multiple experiments. It is shown that UTrans attains much lower estimation and prediction errors than the existing methods, while preserving interpretability. We finally apply it to the US intergenerational mobility data and compare our proposed algorithms to the classical machine learning algorithms.

1 INTRODUCTION

Predictive models, which employ the training data to make predictions, have been effectively used to guide decision making in various applications. Modern data extraction techniques further improve model performance and statistical inference by utilizing a collection of massive and diverse data (Zhuang et al., 2020; Tripuraneni et al., 2020; Liu and Lin, 2023). With data collected from multiple sources, the superior predictive ability of these models relies on the hypothesis that these multi-source data share a homogeneous or similar distribution. When such hypothesis fails, most predictive models using the training data lose the prediction power and require reconstruction by gathering new data from the same distribution. However, the cost of collecting new data or the privacy limit of integrating multiple data may hinder the reconstruction. To improve the predictive performance, one of the possible solutions is to transfer and integrate the useful source data. In this scenario, transferring data knowledge from one source (namely, *source data*) to another (namely, *target data*) would be required, of which the learning process is called *transfer learning* in the literature (Olivas et al., 2009). The three main themes for researchers in transfer learning are: what to transfer, when to transfer, and how to transfer?

Transfer learning has drawn extensive attention for decades and been applied in many fields including Web-document classification, Wifi data calibration, medical diagnosis, and so on. See more examples in the recent survey paper (Zhuang et al., 2020). Beyond these applications of transfer learning in the machine learning community, some methodological and theoretical works are also developed. Yogatama and Mann (2014) proposes a fast and effective algorithm for automatic hyperparameter tuning that utilizes sequential model-based optimization (SMBO) to construct a common response surface across datasets, enabling generalization. Wei et al. (2018) studies how to automatically determine what and how to transfer by leveraging previous transfer learning experiences. Bellot and van der Schaar (2019) introduces a survival prediction model that enhances predictions in a small data domain, like a local hospital, by leveraging related data from other domains, constructing an ensemble of weak survival predictors that iteratively adapts marginal distributions to improve predictions for target patients of interest. Tripuraneni et al. (2020) studies a two-stage empirical risk minimization procedure to transfer learning and provides generalization bounds with general losses, tasks, and features. However, little attention has been paid to interpretable transfer learning in the statistical framework, which can generate interpretable results and study the corresponding theoretical properties. In this paper, we aim to fill this gap, develop new statistical transfer learn-

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

ing models in the context of high-dimensional data, and improve the predictive performances of the existing transfer learning models.

1.1 High-Dimensional Transfer Learning Models

High-dimensional linear models based on one source data with suitable regularizations have been developed extensively over the past decade (Tibshirani, 1996; Fan and Li, 2001) due to the high-dimensional nature of real-world data. For example, in gene expression data, it is common to encounter a few observations but hundreds of thousands of genes. In financial data, it is widely seen that the number of features is much larger than the number of individual stocks. The high-dimensional linear regression model, with single-source data, takes the form $\mathbf{y}_1 = \mathbf{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\epsilon}_1$, where $\mathbf{y}_1 \in \mathbb{R}^{n_1}$, $\mathbf{X}_1 \in \mathbb{R}^{n_1 \times p}$, $\boldsymbol{\beta}_1 \in \mathbb{R}^p$, and $\boldsymbol{\epsilon}_1 \in \mathbb{R}^{n_1}$. With the high-dimensional data, we allow the dimension $p \gg n_1$ for the unknown coefficient vector $\boldsymbol{\beta}_1$.

Transfer learning has been studied recently in statistical models (Li et al., 2022; Tian and Feng, 2022; Lin and Reimherr, 2022). For example, in the high-dimensional linear regression model (Li et al., 2022), the target model is $y_{0i} = \mathbf{x}_{0i}^\top\boldsymbol{\beta}_0 + \epsilon_{0i}, i = 1, \cdots, n_0$ and the source model from the k-th source data, $k = 1, \cdots, K'$, is $y_{ki} = \mathbf{x}_{ki}^\top\boldsymbol{\beta}_k + \epsilon_{ki}, i = 1, \cdots, n_k$, where $\mathbf{x}_{ki} \in \mathbb{R}^p$ and $\boldsymbol{\beta}_k \in \mathbb{R}^p, k = 0, 1, \cdots, K'$. Useful source data are transferred to the target data only if the transferring set $\mathcal{A}_h$ satisfies $\mathcal{A}_h = \{1 \le k \le K' : \|\boldsymbol{\beta}_0 - \boldsymbol{\beta}_k\|_q \le h\}$ for a relatively small *transferring level* h. This model, named Trans-Lasso, leverages the linear regression model to bridge the source and target data and transfers source data to the target data when $k \in \mathcal{A}_h$. Trans-Lasso solves $\mathbf{w}$ from the source data in the first step and then debiases the estimation from the target data in the second step. Let $n_{\mathcal{A}_h}$ denote the sample size of the source data in $\mathcal{A}_h$. More specifically, the first step solves

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2n_{\mathcal{A}_h}} \sum_{k\in\mathcal{A}_h} \left\| \mathbf{y}^{(k)} - \mathbf{X}^{(k)}\mathbf{w} \right\|_2^2 + \lambda\|\mathbf{w}\|_1 \right\}$$

via integrating the diverse information from multiple sources. Tian and Feng (2022) and Lin and Reimherr (2022) extend the results of Li et al. (2022) to high-dimensional generalized linear models (GLMs) and functional linear models, respectively. Some consistent estimators of $\mathcal{A}_h$ are required, such as the Q-aggregation (Li et al., 2022) and data-splitting estimator under some conditions (Tian and Feng, 2022). Other nonparametric predictive models also exist in the literature, such as the adaptive transfer learning with minimax optimal rates of convergence based on k-nearest neighbour (Cai and Wei, 2021; Reeve et al., 2021). Noteworthy, multi-task learning is a closely related topic to transfer learning, but with different goals and interests. Multi-task learning method integrates multiple learning tasks simultaneously, while exploiting a shared structure across all tasks. For example, see the structure of Data Shared Lasso (Gross and Tibshirani, 2016; Ollier and Viallon, 2017) for high-dimensional multi-task learning. In contrast, the interest of transfer learning is to learn the target data only by transferring some shared knowledge from the source data. Therefore, learning the source data is not the focus of transfer learning.

In this paper, our contributions include

1. We propose a novel unified transfer learning model by redefining the design matrix and the response vector in the context of the high-dimensional linear regression with a flexible penalty function. When the transferring set is known, the theoretical results show that it attains tighter upper bounds of the ℓ_1/ℓ_2 estimation errors than Lasso using the target data only. We also compare our theoretical results to the existing methods.

2. Detecting the transferable data, including transferable source data and transferable variables, is a major task in transfer learning. Our unified model is able to automatically identify the transferable variables after model estimation. To the best of our knowledge, this is the first work for identifying the transferable variables by the model's nature and the first work for detecting transferable source data by hypothesis testing.

2 UNIFIED TRANSFER LEARNING MODELS

Notations: We denote scalars with unbolded letters (e.g., sample size n and dimensionality p), (random) vectors with boldface lowercase letters (e.g., $\mathbf{y}$ and $\boldsymbol{\beta}$), and matrices with boldface capital letters (e.g., $\mathbf{X}$). Let $\{(\mathbf{X}_k, \mathbf{y}_k) : \mathbf{X}_k \in \mathbb{R}^{n_k \times p}, \mathbf{y}_k \in \mathbb{R}^{n_k}\}_{k=1}^{K'}$ denote the multiple source data and let $(\mathbf{X}_0, \mathbf{y}_0)$ be the target data. We use $\top$ to represent the transpose of vectors or matrices, such as $\mathbf{x}^\top$ and $\mathbf{X}^\top$. For a p-dimensional vector $\mathbf{x} = (x_1, \cdots, x_p)$, the ℓ_0 norm is the number of nonzero elements. $\|\mathbf{x}\|_q$ and $\|\mathbf{x}\|_\infty$ are the ℓ_q norm and maximum norm, respectively. $|\mathcal{M}|$ denotes the cardinality of the set $\mathcal{M}$. A set with superscript c denotes its complement. We use letters C and c with different subscriptions to denote the positive and absolute constants. Let $a_n = O(b_n)$ denote $|a_n/b_n| \le c$ for some constant c when n is large enough. Let $a_n = O_P(b_n)$

and $a_n \lesssim b_n$ denote $P(|a_n/b_n| \leq c) \to 1$ for $c < \infty$. Let $a_n = o_P(b_n)$ denote $P(|a_n/b_n| > c) \to 0$ for $c > 0$. Finally, $a_n \asymp b_n$ means that a_n/b_n converges to some positive constant.

Throughout the following sections, we abbreviate $\mathcal{A}_h$ by $\mathcal{A}$ for simplicity and use K to denote the number of transferable source data. The first step (namely, transferring step) of the transfer learning models for high-dimensional linear regression in Li et al. (2022) is essentially equivalent to stacking all source data, assuming $\mathcal{A}$ is known:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2n_{\mathcal{A}}} \left\|\mathbf{y}' - \mathbf{X}'\mathbf{w}\right\|_2^2 + \lambda\|\mathbf{w}\|_1 \right\}, \quad (1)$$

where $\mathbf{y}' = [\mathbf{y}_1^\top, \cdots, \mathbf{y}_K^\top]^\top$, $\mathbf{X}' = [\mathbf{X}_1^\top, \cdots, \mathbf{X}_K^\top]^\top$, and $n_{\mathcal{A}}$ is the total sample size of the source data. Tian and Feng (2022) proposes to stack the source data and the target data in the GLMs in the transferring step. We call these methods as vertical stacking methods. The assumption behind these methods is that the data (the source data in $\mathcal{A}$ or the target and the source data in $\mathcal{A}$) share a similar coefficient $\mathbf{w}$. Stacking the data in the way of Eq. (1) may produce a better estimation when different data are close, but might be insufficient to identify the transferable variables in the source data. For example, we are unable to identify the transferable variables to the target data for the k-th source data. Therefore, we consider a new approach, unified transfer learning models, for transfer learning in the high-dimensional linear regression in this section.

2.1 $\mathcal{A}$-UTrans: Transfer Learning with Known $\mathcal{A}$

Instead of stacking the target data and the source data in $\mathcal{A}$ vertically, we propose to stack them both vertically and horizontally by

$$\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \\ \mathbf{y}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) + \cdots + \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_K \\ \mathbf{X}_0 \end{bmatrix} \boldsymbol{\beta}_0 + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_1^\top, \cdots, \boldsymbol{\epsilon}_K^\top, \boldsymbol{\epsilon}_0^\top]^\top$. The aforementioned model can be written as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{y} = [\mathbf{y}_1^\top, \cdots, \mathbf{y}_K^\top, \mathbf{y}_0^\top]^\top$, $\boldsymbol{\beta} = [(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0)^\top, (\boldsymbol{\beta}_2 - \boldsymbol{\beta}_0)^\top, \cdots, \boldsymbol{\beta}_0^\top]^\top \in \mathbb{R}^{p^*}$, and

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{X}_1 \\ \mathbf{0} & \mathbf{X}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{X}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{X}_K & \mathbf{X}_K \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{X}_0 \end{bmatrix} \in \mathbb{R}^{(n_{\mathcal{A}}+n_0)\times p^*} \quad (2)$$

where $p^* = Kp + p$ and $\mathcal{A} = \{k : 1 \leq k \leq K\}$. In this paper, we assume that K is fixed.

We consider a more general penalty function that includes the Lasso used in the current literature and other nonconvex regularizers to deal with the high-dimensional data. The loss function of the penalized least square is

$$\mathcal{L}_n(\boldsymbol{\beta}) = \frac{1}{2\,(n_{\mathcal{A}} + n_0)} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + P_\lambda(\boldsymbol{\beta}) \quad (3)$$

and we denote this unified transfer learning model as **$\mathcal{A}$-UTrans**. We solve $\widehat{\boldsymbol{\beta}}_0$ by the coordinate descent algorithm for nonconvex penalized regression (Breheny and Huang, 2011). Note that $\widehat{\boldsymbol{\beta}}_0$ equals the last p elements of $\widehat{\boldsymbol{\beta}}$.

The penalty function $P_\lambda(\boldsymbol{\beta}) = \sum_{j=1}^{p^*} p_\lambda(|\beta_j|)$ satisfies the following conditions

- (i) $P_\lambda(0) = 0$ and $P_\lambda(t)$ is symmetric around 0.
- (ii) $P_\lambda(t)$ is differentiable for $t \neq 0$ and $\lim_{t\to 0^+} P'_\lambda(t) = \lambda L$.
- (iii) $P_\lambda(t)$ is a non-decreasing function for $t \geq 0$.
- (iv) $P_\lambda(t)/t$ is a non-increasing function for $t > 0$.
- (v) There exists $\tau > 0$ such that $P_\lambda(t) + \frac{\tau}{2}t^2$ is convex.

Conditions (i)–(iii) are relatively mild and used in Zhang and Zhang (2012). Condition (iv) makes sure that the bound of error $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$ is vanishingly small. These mild conditions on $P_\lambda(\boldsymbol{\beta})$ are commonly satisfied by many regularizers including Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), and MCP (Zhang, 2010). For more details, refer to Loh and Wainwright (2015).

We argue two benefits of our $\mathcal{A}$-UTrans models. First, the unified transfer learning model explicitly writes the contrasts $\boldsymbol{\beta}_k - \boldsymbol{\beta}_0$. The k-th source data are transferable if $\boldsymbol{\beta}_k - \boldsymbol{\beta}_0 = \mathbf{0}$. This method, therefore, provides an opportunity to detect transferable source by testing $\boldsymbol{\beta}_k - \boldsymbol{\beta}_0 = \mathbf{0}$. In Section 3, we propose to use hypothesis testing to detect transferable source data. Second, other than detecting the transferable source data, our method can also detect the transferable variables in each source data. For example, we obtain the set containing the transferable variables in the k-th source data by $\mathcal{T}_k = \{j : (\widehat{\boldsymbol{\beta}_k - \boldsymbol{\beta}_0})_j = 0, 1 \leq j \leq p\}$.

2.2 Theoretical Properties of $\mathcal{A}$-UTrans

We define the parameter space of $\mathcal{A}$-UTrans by $\Theta(s, h)$, which is

$$\{\boldsymbol{\beta} : \max_{k\in\mathcal{A}} \|\boldsymbol{\beta}_k - \boldsymbol{\beta}_0\|_1 \leq h, \|\boldsymbol{\beta}_0\|_0 \leq s, \|\boldsymbol{\beta}_k - \boldsymbol{\beta}_0\|_0 \leq Cs\}$$

for some constant C and $s = \|\boldsymbol{\beta}_0\|_0$ is the sparsity level. Note that this parameter space specifies the sparsity of $\boldsymbol{\beta}_0$ and constraints the maximum ℓ_1 distance between $\boldsymbol{\beta}_k$ and $\boldsymbol{\beta}_0$ to h. We further impose the following conditions to study the theories of $\mathcal{A}$-UTrans:

- C1. Each row of $\mathbf{X}_k, k \in \mathcal{A} \cup \{0\}$, is independent and identically distributed (i.i.d) normal random vector with mean zero and covariance matrix $\boldsymbol{\Sigma}_k$.
- C2. The random noises ϵ_{ki} in the k-th source data, $i = 1, \cdots, n_k$ and $k \in \mathcal{A} \cup \{0\}$, are i.i.d sub-Gaussian random variable with mean zero and parameter σ_k^2.
- C3. The sample covariance matrix $\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k}\mathbf{X}_k^\top \mathbf{X}_k$, $k \in \mathcal{A} \cup \{0\}$, satisfies the restricted strong convexity (RSC) condition

$$\boldsymbol{\Delta}_k^\top \widehat{\boldsymbol{\Sigma}}_k \boldsymbol{\Delta}_k \geq v_k \|\boldsymbol{\Delta}_k\|_2^2 - \tau_k \sqrt{\frac{\log p}{n_k}} \|\boldsymbol{\Delta}_k\|_1$$

for any $\boldsymbol{\Delta}_k \in \mathbb{R}^p$ and $\|\boldsymbol{\Delta}_k\|_1 \geq 1$, where $v_k > 0$ and $\tau_k \geq 0$.

In C1, the source data and the target data are assumed to have Gaussian designs. The covariance matrix $\boldsymbol{\Sigma}_k$ can be homogeneous or heterogeneous among the source and the target data. Different from Li et al. (2022) whose theories are established separately with the homogeneous and heterogeneous covariance matrices, our theories can incorporate both. This condition is for theoretical convenience and can be relaxed to sub-Gaussian random variable. Condition C2 assumes the sub-Gaussian random noises for the source and the target data, which are used for the convergence rate analysis. Condition C3 assumes the RSC condition for each sample covariance matrix. This condition is widely used to study the non-asymptotic error bounds in high-dimensional statistics. It is shown that the RSC condition is met with high probability under sub-Gaussian assumption (Agarwal et al., 2012; Loh and Wainwright, 2015; Liu et al., 2022). We mention that the RSC condition can be replaced by the restricted eigenvalue (RE) condition (Bickel et al., 2009; Van De Geer and Bühlmann, 2009). For simplicity, denote $n = n_{\mathcal{A}} + n_0$. We have the following RSC condition on the sample covariance matrix of $\mathbf{X}$.

Theorem 1 *Let $\widehat{\boldsymbol{\Sigma}} = \mathbf{X}^\top \mathbf{X}/n$ be the sample covariance matrix of $\mathbf{X}$. With the RSC conditions on each $\widehat{\boldsymbol{\Sigma}}_k$, we have*

$$\widehat{\boldsymbol{\Delta}}^\top \widehat{\boldsymbol{\Sigma}} \widehat{\boldsymbol{\Delta}} \geq v' \|\widehat{\boldsymbol{\Delta}}\|_2^2 - \tau_0 \left(\sqrt{\frac{n_m \log p}{n^2}} + \sqrt{\frac{n_0 \log p}{n^2}} \right) \|\widehat{\boldsymbol{\Delta}}\|_1$$

for $\widehat{\boldsymbol{\Delta}} = \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta} \in \mathbb{R}^{p^*}$, where $v' = \min_k v_k n_k/n > 0$, $\tau_0 = \max_k \tau_k(K+1) \geq 0$, and $n_m = \max_{k \in \mathcal{A}} n_k$, $k \in \mathcal{A}$.

Theorem 1 implies that the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$ in the unified model admits a similar RSC condition as that from a single source data. The term $\sqrt{n_m \log p/n^2} + \sqrt{n_0 \log p/n^2} \lesssim \sqrt{\log p/n}$ in the lower bound is essential for establishing the estimation error bound. Note that this term is upper bounded by $\sqrt{\log p/n_0}$. Thus, a tighter error bound than the model using target data only can be established. From this theorem, we observe

$$\widehat{\boldsymbol{\Delta}}^\top \widehat{\boldsymbol{\Sigma}} \widehat{\boldsymbol{\Delta}} \geq v' \|\widehat{\boldsymbol{\Delta}}\|_2^2 - 2\tau_0 \sqrt{\frac{\log p}{n}} \|\widehat{\boldsymbol{\Delta}}\|_1,$$

which trivially holds for $\frac{\|\widehat{\boldsymbol{\Delta}}\|_1}{\|\widehat{\boldsymbol{\Delta}}\|_2^2} \geq \frac{v'}{2\tau_0}\sqrt{\frac{n}{\log p}}$ since the left-hand side is nonnegative. Thus, we only enforce a type of strong convexity condition over a cone of the form

$$\left\{ \frac{\|\widehat{\boldsymbol{\Delta}}\|_1}{\|\widehat{\boldsymbol{\Delta}}\|_2^2} \leq \frac{v'}{2\tau_0}\sqrt{\frac{n}{\log p}} \right\}.$$

Based on Theorem 1, we have the following ℓ_1/ℓ_2 estimation error bounds.

Theorem 2 (Convergence rates of $\mathcal{A}$-UTrans) *With the conditions on the regularizer $P_\lambda(\boldsymbol{\beta})$ and conditions C1–C3, let $\lambda = c_1\sqrt{\frac{\log p}{n}}$ for a positive constant c_1. Suppose $\mathcal{A}$ is known and $(s \log p/n)^{1/2} + h^{1/2}(\log p/n)^{1/4} = o(1)$, then there exists some positive constant c such that*

$$\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|_2 \lesssim \left(\frac{s \log p}{n}\right)^{1/2} + \left(\frac{\log p}{n}\right)^{1/4} h^{1/2}$$

and

$$\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|_1 \lesssim s\left(\frac{\log p}{n}\right)^{1/2} + \left(\frac{\log p}{n}\right)^{1/4} (sh)^{1/2}$$

hold with probabilities at least $1 - cp^{-1}$, where $h = \max_{k \in \mathcal{A}} \|\boldsymbol{\beta}_k - \boldsymbol{\beta}_0\|_1$.

Theorem 2 shows how the estimation errors of $\boldsymbol{\beta}_0$ are affected by $n_{\mathcal{A}}, n_0, s, p$, and h. The ℓ_1 error can be analyzed similarly to the ℓ_2 error, so we only analyze the ℓ_2 error here. In transfer learning, we are more interested in the scenario of a small n_0 but diverging $n_{\mathcal{A}}$ since it is more realistic. First, with a fixed n_0 and $n_{\mathcal{A}} \to \infty$, our result indicates that the estimation error goes to 0. When the size of transferable source data is large enough, the effect of h on estimation is dominated by an extremely large $n_{\mathcal{A}}$. Indeed, as the simulation study shows (see Figure 3 in Section 3 and

also Figure 3 in Tian and Feng (2022)), the estimation error is dominated by a large $n_{\mathcal{A}}$ even with a relatively large h. The scenarios of very large h necessitate the source detection algorithm introduced in Section 3. Besides, $h = 0$ implies that the source data are completely transferable to the target data ($\boldsymbol{\beta}_k = \boldsymbol{\beta}_0$). In this case, the ℓ_2 error becomes $O_P(\sqrt{\frac{s\log p}{n}})$, the convergence rate of stacking all data vertically. Second, without any available source data ($n_{\mathcal{A}} = 0$ and $h = 0$), the ℓ_2 upper bound becomes $\sqrt{\frac{s\log p}{n_0}}$, the same rate as Lasso on target data only. Third, Theorem 2 holds with the condition $s\log p = o(n_{\mathcal{A}})$ when $n_0 \lesssim n_{\mathcal{A}}$, which is weaker than the condition $s\log p = o(n_0)$ for Lasso using the target data only. Fourth, the ℓ_2 error bound of $\mathcal{A}$-Trans-GLM (Theorem 1 of Tian and Feng (2022)) is $(s\log p/n)^{1/2} + [(\log p/n_0)^{1/4}h^{1/2}] \wedge h$. It is not hard to see that ours is the same as $\mathcal{A}$-Trans-GLM when $h \lesssim (\log p/n)^{1/2}$ and tighter than that when $h \gg (\log p/n)^{1/2}$ and $n_0 \ll n_{\mathcal{A}}$.

Theorem 3 (Prediction error bound of $\mathcal{A}$-UTrans) *Let $\mathcal{E}_{n_v} = 1/n_v\|\mathbf{X}_v\left(\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\right)\|_2^2$ be the mean squared prediction error based on testing data $\mathbf{X}_v$. With the same conditions in Theorem 2 and some positive constant c,*

$$\mathcal{E}_{n_v} \lesssim \frac{s\log p}{n_v} + \left(\frac{\log p}{n_v}\right)^{3/4}(sh)^{1/2} + h\left(\frac{\log p}{n_v}\right)^{1/2}$$

holds with probability at least $1-cp^{-1}$, where $\mathbf{X}_v$ is the testing data and n_v is the corresponding testing data size.

3 UTrans: TRANSFER LEARNING WITH SOURCE DETECTION

The $\mathcal{A}$-UTrans algorithm in Section 2 assumes that the source data and the target data are similar to some extent, which might be unrealistic for an arbitrary dataset since h can be small or large. In fact, transferring nontransferable source data to the target data may bring adverse effects and lead to worse performance than the model with target data only (Pan and Yang, 2009; Tian and Feng, 2022). Therefore, a source detection algorithm is necessary in transfer learning.

Recall that our unified model, with $\mathbf{X}_k$ and $\mathbf{X}_0$, explicitly writes out the contrast $\boldsymbol{\beta}_k - \boldsymbol{\beta}_0$ with

$$\boldsymbol{\mu} = \begin{bmatrix}\mathbf{X}_k & \mathbf{X}_k\\ \mathbf{0} & \mathbf{X}_0\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_k - \boldsymbol{\beta}_0\\ \boldsymbol{\beta}_0\end{bmatrix} := \mathbf{W}(\boldsymbol{\beta}_k - \boldsymbol{\beta}_0) + \mathbf{Z}\boldsymbol{\beta}_0$$

where $\mu = E(Y|\mathbf{Z},\mathbf{W})$, $\mathbf{W} = (\mathbf{X}_k^\top, \mathbf{0})^\top$, and $\mathbf{Z} = (\mathbf{X}_k^\top, \mathbf{X}_0^\top)^\top$. Let $\boldsymbol{\beta} = [(\boldsymbol{\beta}_k - \boldsymbol{\beta}_0)^\top, \boldsymbol{\beta}_0^\top]^\top$ (note that $\boldsymbol{\beta}$ is defined differently from that in Section 2). By testing $H_0 : \boldsymbol{\beta}_k - \boldsymbol{\beta}_0 = \mathbf{0}$ vs $H_1 : \boldsymbol{\beta}_k - \boldsymbol{\beta}_0 \neq \mathbf{0}$, we detect if the source data $\mathbf{X}_k$ are transferable to $\mathbf{X}_0$.

Both the parameter of interest $\boldsymbol{\beta}_k - \boldsymbol{\beta}_0$ and the nuisance parameter $\boldsymbol{\beta}_0$ are p-dimensional. Methods on testing the high-dimensional vector with high-dimensional nuisance parameter is very limited in the literature. Recently, Chen et al. (2022) proposes a U test statistic for the high-dimensional regression models, which extends the results of testing the low-dimensional parameter of interest in Goeman et al. (2011) and Guo and Chen (2016). We propose an asymptotic α-level test that rejects H_0 if

$$|\hat{U}_{n_k}|/\sqrt{2\hat{R}_{n_k}} > z_{1-\alpha/2}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-th quantile of a standard normal distribution and

$$\hat{U}_{n_k} = \frac{1}{n_k}\sum_{i\neq i'}^{n_k}\left\{(y_i - \hat{\mu}_{\varnothing i})(y_{i'} - \hat{\mu}_{\varnothing i'})\,\mathbf{x}_{ki}^\top\mathbf{x}_{ki'}\right\}$$

$$\hat{R}_{n_k} = \tfrac{1}{n_k^2 - n_k}\textstyle\sum_{i\neq i'}^{n_k}\left\{(y_i - \hat{\mu}_{\varnothing i})^2(y_{i'} - \hat{\mu}_{\varnothing i'})^2\left(\mathbf{x}_{ki}^\top\mathbf{x}_{ki'}\right)^2\right\},$$

$\hat{\mu}_{\varnothing i} = \mathbf{x}_{ki}^\top\widehat{\boldsymbol{\beta}}_0$ where $\widehat{\boldsymbol{\beta}}_0$ is obtained by fitting $\mu = \mathbf{z}^\top\boldsymbol{\beta}_0$ under the null hypothesis. Note that $\mathbf{z}$ is high-dimensional, so we obtain $\widehat{\boldsymbol{\beta}}_0$ with the Lasso regression. Denote $\Lambda_W^\epsilon = \mathrm{tr}[E\{\mathrm{var}(\epsilon)\mathbf{x}_k\mathbf{x}_k^\top\}^2]$ where $\epsilon = \mathbf{y}_k - \mathbf{x}_k^\top\boldsymbol{\beta}_k$. Assume

- C4. Under H_0, there exist finite positive constants c_1 and C_1 such that

  $$c_1 \leq \lambda_{\min}\left\{E(\mathbf{X}_k\mathbf{X}_k^\top)\right\} \leq \lambda_{\max}\left\{E(\mathbf{X}_k\mathbf{X}_k^\top)\right\} \leq C_1,$$

  where $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of $E(\mathbf{X}_k\mathbf{X}_k^\top)$, respectively.

Theorem 4 *Assume the conditions C1–C2 and C4 and $s\log p/n = o(1)$. Under H_0, if $n_k s\log p/(n\sqrt{2\Lambda_W^\epsilon}) = o(1)$, then*

$$\lim_{n_k\to\infty}\sup_{\|\boldsymbol{\beta}_0\|_2 = O(1)} P\left(\frac{|\hat{U}_{n_k}|}{\sqrt{2\hat{R}_{n_k}}} > z_{1-\alpha/2}\right) = \alpha.$$

Theorem 4 shows the probability of making the type I error (incorrectly excluding $\mathbf{X}_k$ when it is transferable). Under some conditions, we find that the probability of making such error becomes small as $n_k \to \infty$.

Algorithm UTrans utilizes the tool of hypothesis testing to detect transferable source data. To the best of our knowledge, this is the first work of using statistical inference for source detection in transfer learning.

Algorithm 1: UTrans

Input: $\{(\mathbf{X}_k, \mathbf{y}_k), 0 \le k \le K'\}$.

for $k \leftarrow 1$ **to** K' **do**

(1) write $\mathbf{W} = (\mathbf{X}_k^\top, \mathbf{0})^\top$ and $\mathbf{Z} = (\mathbf{X}_k^\top, \mathbf{X}_0^\top)^\top$.

(2) estimate $\widehat{\boldsymbol{\beta}}_0$ by fitting the model $\mu = \mathbf{z}^\top \boldsymbol{\beta}_0$.

(3) compute $\hat{\mu}_{\varnothing i} = \mathbf{x}_{ki}^\top \widehat{\boldsymbol{\beta}}_0$ and calculate the test statistic $t_k = |\hat{U}_{n_k}| / \sqrt{2\hat{R}_{n_k}}$.

end

$\widehat{\mathcal{A}} = \{k : t_k \le z_{1-\alpha/2/K'}\}$.

$\mathcal{A}$**-UTrans:** obtain $\widehat{\boldsymbol{\beta}}_0$ by minimizing (3); obtain $\mathcal{T}_k = \{j : (\widehat{\boldsymbol{\beta}_k - \boldsymbol{\beta}_0})_j = 0\}$.

Output: $\widehat{\mathcal{A}}$, $\widehat{\boldsymbol{\beta}}_0$, and $\mathcal{T}_k$.

We point out the benefit of our source detection algorithm. Compared to Trans-GLM (Tian and Feng, 2022) which depends on the unknown constant C_0, our algorithm has no extra unknown parameters. In fact, C_0 determines the threshold to select the transferable source data. Without knowing the true value of C_0, a large value overestimates $\mathcal{A}$ and a small value underestimates $\mathcal{A}$. Another round of cross validation can be run to find C_0, but increases computational cost. Nevertheless, our algorithm estimates $\widehat{\mathcal{A}}$ by directly testing $\boldsymbol{\beta}_k - \boldsymbol{\beta}_0 = \mathbf{0}$, which is more computationally efficient.

4 EXPERIMENTS

We illustrate the performances of our $\mathcal{A}$-UTrans and UTrans in various settings in terms of the averaged ℓ_2-estimation error and the mean squared prediction error. More specifically, we compare the following models: (1) $\mathcal{A}$-Trans-GLM and Trans-GLM: a two-step transferring model for linear regression without and with source detection, respectively, proposed by Tian and Feng (2022); (2) Trans-Lasso: a two-step transfer learning model for linear regression with source detection, proposed by Li et al. (2022); (3) naive-Lasso: a model that fits the target data only using Lasso regression (Tibshirani, 1996); (4) $\mathcal{A}$-UTrans-Lasso and $\mathcal{A}$-UTrans-SCAD: the proposed unified transfer learning models with Lasso and SCAD penalties. R packages *glmtrans* and *glmnet* are used to implement Trans-GLM and Lasso on the target data only, respectively (R Core Team, 2024). Our UTrans is implemented by the R package *ncvreg*.

We consider 10 different settings for the number of source data, i.e., K (subsection 4.1) and K' (subsection 4.2) range from 1 to 10. With each K and K', experiments are replicated 200 times. Note that methods in Li et al. (2022) and Tian and Feng (2022) mainly differ in the source detection algorithms, so we only compare Trans-Lasso in subsection 4.2.

4.1 Simulation with Known $\mathcal{A}$

This subsection is to show the theoretical properties in Theorem 2 and the advantages of our $\mathcal{A}$-UTrans algorithms in high-dimensional transfer learning with different dimensionalities p, target sizes n_0, and transferring levels h. We consider simulations with $n_0 \in \{50, 75, 100\}$, $p \in \{300, 500, 600, 900\}$, and $h \in \{5, 10, 20, 40\}$. We let the sample size of the source data $n_k = 100$ for all $k = 1, \cdots, K$ and fix the sparsity level $s = 5$ in the target data.

For the target data, let $\boldsymbol{\beta}_0 = (\mathbf{0.5}_s, \mathbf{0}_{p-s})$, where $\mathbf{0.5}_s$ means s repetitions of 0.5 and $\mathbf{0}_{p-s}$ means $p-s$ repetitions of 0. Each target sample $\mathbf{x}_{0i} \overset{iid}{\sim} \mathcal{N}(\mathbf{0}_p, \boldsymbol{\Sigma})$ with element $\Sigma_{jj'} = 0.5^{|j-j'|}$ for $i = 1, \cdots, n_0$ and $1 \le j, j' \le p$. For the k-th source data, we let $\boldsymbol{\beta}_k = (\mathbf{0.5}_s + (h/p)\mathcal{R}_s, \mathbf{0}_{p-s})$, where $\mathcal{R}_s$ is a s-dimensional independent Rademacher variable. Each sample is generated from a p-dimensional $\mathcal{N}(\mathbf{0}_p, \boldsymbol{\Sigma} + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top)$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_p, 0.3^2\mathbf{I}_p)$.

Figure 1 depicts the mean squared prediction errors of all models with different simulation settings. More specifically, row A shows the results under different dimensionalities p. We fix $n_0 = 100$ and $h = 5$. First, our proposed $\mathcal{A}$-UTrans-Lasso and $\mathcal{A}$-UTrans-SCAD outperform all the others. Second, the naive-Lasso model fluctuates with the highest error no matter how K increases, since K controls the number of source data and naive-Lasso fits the target data only. Row B shows the MSPEs of all models with different target sizes n_0. We fix $p = 500$ and $h = 5$. First, our proposed $\mathcal{A}$-UTrans algorithms have the best performances even with small sample sizes. This evidence shows the benefit of transfer learning when the size of target data is small. Second, the MSPEs of all models decrease as n_0 increases while $\mathcal{A}$-UTrans-SCAD attains the lowest error. Row C illustrates the MSPEs of all models with various h. We fix $n_0 = 100$ and $p = 500$. As the level h increases, prediction errors of all transfer learning models with small K increase but they fluctuate as K increases.

Figure 2 shows the averaged ℓ_2 estimation errors of the four methods. More specifically, our $\mathcal{A}$-UTrans algorithms obtain much lower errors than the others. As K increases, the errors of $\mathcal{A}$-UTrans-Lasso and $\mathcal{A}$-UTrans-SCAD drop dramatically. This further shows that our algorithms have lower errors than the two-step $\mathcal{A}$-Trans-GLM. The condition for improving the target model in Li et al. (2022) and Tian and Feng (2022) allows h as large as $\sqrt{n_0/\log p}$. In other words, they make a better upper bound better than the naive

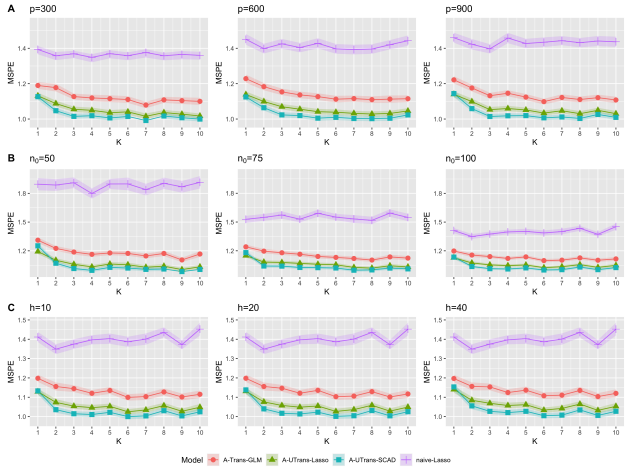

Figure 1: Mean squared prediction errors (MSPEs) of the proposed unified model and the existing transfer learning models with different settings of p (row A), n_0 (row B), and h (row C) for each $k = 1, \cdots, K$. Shade areas are calculated by MSPE $\pm 0.1 \times$ standard deviation (SD).

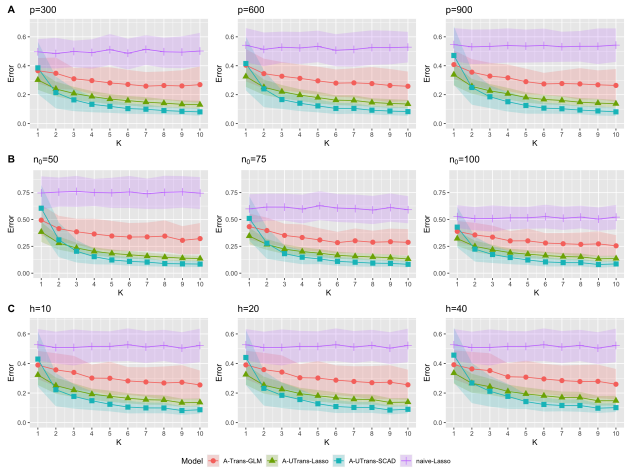

Figure 2: The averaged ℓ_2 estimation errors of naive-Lasso, $\mathcal{A}$-Trans-GLM, $\mathcal{A}$-UTrans-Lasso, and $\mathcal{A}$-UTrans-SCAD with different settings. Shade areas are calculated by estimate $\pm$ SD.

Lasso under this condition. With these three simulation settings, this condition is satisfied in most settings and therefore improvement or theoretical property is granted. While, our $\mathcal{A}$-UTrans still outperforms $\mathcal{A}$-Trans-GLM. Overall, this simulation study presents that our proposed $\mathcal{A}$-UTrans maintains relatively low prediction errors in all settings. Particularly, $\mathcal{A}$-UTrans-SCAD outperforms all the others with relatively lower errors.

4.2 Simulation with Source Detection

In subsection 4.1, we consider the cases when the values of h are relatively small. Here, we consider the cases with relatively large h and examine the effectiveness of the source detection algorithms. We fix $p = 500$, $K' = 10$, and the source data sizes $n_k = 200$ for $k \in \mathcal{A}$. The target data are simulated in the same way as subsection 4.1. For the k-th source data, each sample is generated from a t-distribution with degrees of freedom 4 and the covariance $\Sigma_{jj'} = 0.5^{|j-j'|}$ for $i = 1, \cdots, n_k$ and $1 \leq j \neq j' \leq p$. Note that we violate the assumptions C1 and C2 to show the robustness of UTrans with different data distributions. We let $\boldsymbol{\beta}_0 = (-\mathbf{0.4}_3, -\mathbf{0.5}_3, \mathbf{0.6}_4, \mathbf{0}_{490})$, $\boldsymbol{\beta}_k = \boldsymbol{\beta}_0$ if the k-th source data are transferable, and $\boldsymbol{\beta}_k = \boldsymbol{\beta}_0 + h\mathcal{R}_p$ otherwise.

Figure 3 and Figure 4 show the estimation and prediction errors from naive-Lasso, Trans-GLM, Trans-GLM*, Trans-Lasso, UTrans, and UTrans*, respectively. An algorithm with * denotes its pooled version, i.e., combining all the source data and target data. The x-axis ka represents the number of transferable source data. The first row of Figure 3 shows the estimation results with different target sizes and we fix $h = 0.25$. The second row demonstrates the results with different values of h and we fix $n_0 = 100$. When $n_0 = 75$, our algorithm UTrans obtains much lower estimation errors than Trans-GLM, which demonstrates the benefit of using transfer learning in the target data with relatively small size. As h increases, our UTrans keeps the lowest estimation errors among all algorithms in all settings, which shows the effectiveness of excluding nontransferable source data. Overall, this study reveals that our proposed UTrans works better than existing algorithms in small target data and noisy source data. Similar patterns for the prediction errors can be observed in Figure 4.

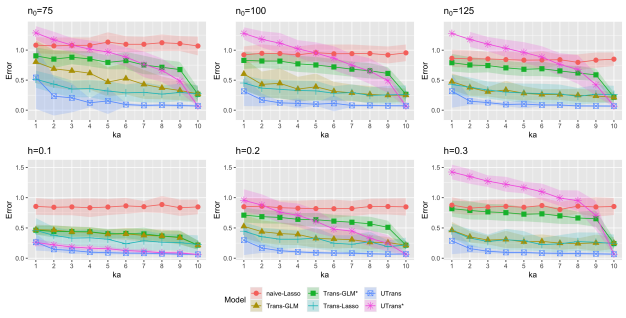

Figure 3: The averaged ℓ_2 estimation errors of naive-Lasso, Trans-Lasso, Trans-GLM, and UTrans with different settings. Shade areas are calculated by estimate $\pm$ SD.

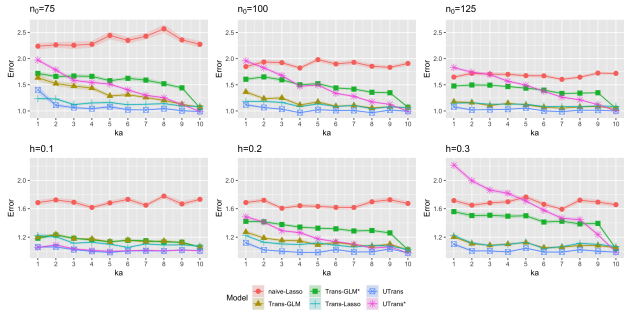

Figure 4: Mean squared prediction errors of the proposed unified model and the existing transfer learning models with different settings of n_0 and h. Shade areas are calculated by MSPE $\pm$ 0.1 $\times$ SD.

5 INTERGENERATIONAL MOBILITY DATA

5.1 Data Description

We use the county-level data collected from the national census data, the Opportunity Atlas, and Data Commons to illustrate our UTrans. Intergenerational mobility is measured as the change in income percentile for the children of all parents at the 75th national income percentile when they are aged 26. Furthermore, we subset states with the numbers of counties larger than 50 for analysis. Of which, states with the numbers of counties between 50 and 75 are treated as the target states while others larger than 75 are treated as the source states. Besides, we add two-way interactions of the county-level characteristics. Overall, the processed data contain 1803 counties and 7875 predictors. The states of interest (target states) include Alabama (AL-66), Arkansas (AR-64), California (CA-52), Florida (FL-65), Louisiana (LA-58), Minnesota (MN-69), New York (NY-61), Oklahoma (OK-60), Pennsylvania (PA-64), and Wisconsin (WI-68). The source states include Georgia (GA-127), Illinois (IL-88), Indiana (IN-87), Iowa (IA-81), Kentucky (KY-98), Michigan (MI-76), Missouri (MO-87), North Carolina (NC-95), Ohio (OH-88), Tennessee (TN-86), Texas (TX-150), and Virginia (VA-113). Number in the brackets denotes sample size, i.e., the number of counties.

5.2 Predictive Analysis

We compare our UTrans to the following algorithms: Trans-GLM, Trans-GLM*, random forest (RF), RF*, XGBoost, XGBoost*, support vector machine (SVM), SVM*, UTrans, and UTrans*, where * denotes the pooled version, i.e., stacking both the source data and the target data. We repeat our experiment 200 times and evaluate these algorithms by the mean squared prediction error. When applying these algorithms, we treat one state as the target data. To make predictions, we randomly split 80% of the target data as training data and the remaining 20% as testing data.

Table 1 shows the mean squared prediction errors for each target state. For each target state, the algorithm with the best performance is highlighted in bold. Notably, UTrans performs the best in three states (CA, MN, and WI). Compared to XGBoost and SVM and their pooled versions, UTrans still maintains relatively low prediction errors. Compared to RF and RF*, which have the lowest errors in three states, our UTrans is also more interpretable in terms of variable importance. The coefficients in linear regression represent the strength and direction of the relationship. RF is generally more difficult to interpret than linear regression since the individual trees can interact in complex ways and the importance of each feature may not be easily discernible from the output. Compared to the transfer learning models Trans-Lasso and Trans-GLM, UTrans performs better than them in all the target states.

6 BROADER IMPACT

In this paper, we propose a novel, unified, and interpretable transfer learning model with high-dimensional data. To the best of our knowledge, this unified model is the first work on transfer learning that identifies both transferable variables and transferable source data; It is also the first work that incorporates the statistical inference tool into transfer learning for source detection. Multiple researches can be directed based on our framework. First, our unified model may be extended to the nonlinear models with extra conditions, such as logistic regression, survival models, etc. Second, our model will shed a light on the statistical learning community since it explicitly writes the contrasts. Developing more powerful tools for source detection is very critical in transfer learning.

Table 1: The mean squared prediction errors for each target state. Model names with stars are run on the pooled data. The bold numbers indicate the lowest prediction errors.

Model	AL	AR	CA	FL	LA	MN	NY	OK	PA	WI
RF	**4.6647**	8.6112	0.4349	**2.2448**	1.1496	1.4110	0.4450	6.9399	**0.5996**	1.4255
RF*	6.1231	7.7380	0.9830	2.3068	5.3322	1.7444	0.5937	7.2940	1.1103	1.7436
XGBoost	6.2520	12.6309	0.6331	2.8952	1.5041	2.6424	0.6036	8.8445	0.8695	1.7809
XGBoost*	5.9989	21.6605	0.6137	3.3179	4.7088	1.5407	0.7315	7.7890	27.9124	24.7909
SVM	5.1180	7.5670	0.3473	2.4102	0.9202	1.1043	0.3923	6.3199	0.7585	1.2256
SVM*	5.0375	**7.5015**	0.3339	2.3980	**0.8894**	1.2443	**0.3791**	**6.2356**	0.7465	1.2017
Trans-Lasso	5.9565	9.6555	0.5729	2.4748	3.3234	1.4261	0.4930	7.0710	0.9106	1.4848
Trans-GLM	5.4706	8.2521	0.4202	2.6550	0.9783	1.0620	0.3943	6.4283	0.9253	1.2637
Trans-GLM*	5.5371	7.9981	0.4185	2.6622	0.9902	1.0621	1.0901	6.3438	0.8828	1.2772
UTrans	5.0616	7.5426	**0.3308**	2.4154	0.8924	**1.0566**	0.3810	6.2586	0.7548	**1.2011**
UTrans*	5.0572	7.5406	0.3328	2.4151	0.8924	**1.0566**	0.3811	6.2581	0.7549	1.2066

References

Agarwal, A., Negahban, S., and Wainwright, M. J. (2012). Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452 – 2482.

Bellot, A. and van der Schaar, M. (2019). Boosting transfer learning with survival data from heterogeneous domains. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 57–65. PMLR.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732.

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232.

Cai, T. T. and Wei, H. (2021). Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. *The Annals of Statistics*, 49(1):100–128.

Chen, J., Li, Q., and Chen, H. Y. (2022). Testing generalized linear models with high-dimensional nuisance parameters. *Biometrika*.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Goeman, J. J., Van Houwelingen, H. C., and Finos, L. (2011). Testing against a high-dimensional alternative in the generalized linear model: asymptotic type i error control. *Biometrika*, pages 381–390.

Gross, S. M. and Tibshirani, R. (2016). Data shared lasso: A novel tool to discover uplift. *Computational Statistics & Data Analysis*, 101:226–235.

Guo, B. and Chen, S. X. (2016). Tests for high dimensional generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1079–1102.

Li, S., Cai, T. T., and Li, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 84(1):149—173.

Lin, H. and Reimherr, M. (2022). On transfer learning in functional linear regression. *arXiv preprint arXiv:2206.04277*.

Liu, S. S. and Lin, L. (2023). Adaptive weighted multiview clustering. In *Conference on Health, Inference, and Learning*, pages 19–36. PMLR.

Liu, W., Yu, X., and Li, R. (2022). Multiple-splitting projection test for high-dimensional mean vectors. *Journal of Machine Learning Research*, 23(71):1–27.

Loh, P.-L. and Wainwright, M. J. (2015). Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16(19):559–616.

Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R., Serrano, L., et al. (2009). *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI global.

Ollier, E. and Viallon, V. (2017). Regression modelling on stratified data with the lasso. *Biometrika*, 104(1):83–96.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Reeve, H. W., Cannings, T. I., and Samworth, R. J. (2021). Adaptive transfer learning. *The Annals of Statistics*, 49(6):3618–3649.

Tian, Y. and Feng, Y. (2022). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, pages 1–14.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tripuraneni, N., Jordan, M., and Jin, C. (2020). On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33:7852–7862.

Van De Geer, S. A. and Bühlmann, P. (2009). On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Wei, Y., Zhang, Y., Huang, J., and Yang, Q. (2018). Transfer learning via learning to transfer. In *35th International Conference on Machine Learning, ICML 2018*, volume 11, page 8059.

Yogatama, D. and Mann, G. (2014). Efficient transfer learning method for automatic hyperparameter tuning. In *Artificial Intelligence and Statistics*, pages 1077–1085. PMLR.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942.

Zhang, C.-H. and Zhang, T. (2012). A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

Technical Proofs

Lemma 1 (Proposition 5.16 (Vershynin, 2010)) *Let $x_1, \cdots, x_n$ be independent centered sub-exponential random variables, and let $M = \max_i \|x_i\|_{\psi_1}$. Then, for every $\mathbf{a} = (a_1, \cdots, a_n)^\top \in \mathbb{R}^n$ and every $t \geq 0$, we have*

$$P\left(\left\|\sum_{i=1}^n a_i x_i\right\| \geq t\right) \leq 2\exp\left[-c\min\left(\frac{t^2}{M^2\|\mathbf{a}\|_2^2}, \frac{t}{M\|\mathbf{a}\|_\infty}\right)\right],$$

where $c > 0$ is an absolute constant.

Lemma 2 (Lemmas 4(b) and 5 of Loh and Wainwright (2015)) *With the regularization function P_λ satisfying the conditions (i)–(v),*

1. *For any $\mathbf{w}$, we have $\lambda L\|\mathbf{w}\|_1 \leq P_\lambda(\mathbf{w}) + \tau/2\|\mathbf{w}\|_2^2$*
2. *Let $\mathcal{I}$ be the index set of the s^* largest elements of $\mathbf{v}$ in magnitude. Suppose $\xi > 0$ is such that $\xi P_\lambda(\mathbf{v}_\mathcal{I}) - P_\lambda(\mathbf{v}_{\mathcal{I}^c}) \geq 0$, then*

$$\xi P_\lambda(\mathbf{v}_\mathcal{I}) - P_\lambda(\mathbf{v}_{\mathcal{I}^c}) \leq \lambda L\left(\xi\|\mathbf{v}_\mathcal{I}\|_1 - \|\mathbf{v}_{\mathcal{I}^c}\|_1\right).$$

Moreover, if $\boldsymbol{\beta}^*$ is s^*-sparse, then for an vector $\boldsymbol{\beta}$ such that $\xi P_\lambda(\boldsymbol{\beta}^*) - P_\lambda(\boldsymbol{\beta}) > 0$ and $\xi \geq 1$, we have

$$\xi P_\lambda(\boldsymbol{\beta}^*) - P_\lambda(\boldsymbol{\beta}) \leq \lambda L(\xi\|\mathbf{v}_\mathcal{I}\|_1 - \|\mathbf{v}_{\mathcal{I}^c}\|_1)$$

where $\mathbf{v} = \boldsymbol{\beta} - \boldsymbol{\beta}^*$.

Proof of Theorem 1

Denote $n = n_\mathcal{A} + n_0$. First, it is not hard to derive

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\mathbf{X}^\top\mathbf{X} = \frac{1}{n}\begin{bmatrix} \mathbf{X}_1^\top & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2^\top & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{X}_K^\top & \mathbf{0} \\ \mathbf{X}_1^\top & \mathbf{X}_2^\top & \cdots & \cdots & \mathbf{X}_K^\top & \mathbf{X}_0^\top \end{bmatrix}\begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{X}_1 \\ \mathbf{0} & \mathbf{X}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{X}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{X}_K & \mathbf{X}_K \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{X}_0 \end{bmatrix}$$

$$= \frac{1}{n}\begin{bmatrix} n_1\widehat{\boldsymbol{\Sigma}}_1 & \mathbf{0} & \cdots & \cdots & \mathbf{0} & n_1\widehat{\boldsymbol{\Sigma}}_1 \\ \mathbf{0} & n_2\widehat{\boldsymbol{\Sigma}}_2 & \mathbf{0} & \cdots & \mathbf{0} & n_2\widehat{\boldsymbol{\Sigma}}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & n_K\widehat{\boldsymbol{\Sigma}}_K & n_K\widehat{\boldsymbol{\Sigma}}_K \\ n_1\widehat{\boldsymbol{\Sigma}}_1 & \cdots & \cdots & \cdots & n_K\widehat{\boldsymbol{\Sigma}}_K & \sum_{k\in\mathcal{A}\cup\{0\}} n_k\widehat{\boldsymbol{\Sigma}}_k \end{bmatrix}.$$

For any $\boldsymbol{\Delta} = (\boldsymbol{\Delta}_1^\top, \cdots, \boldsymbol{\Delta}_K^\top, \boldsymbol{\Delta}_0^\top)^\top$, we have

$$\boldsymbol{\Delta}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Delta} = \begin{bmatrix} n_1/n\boldsymbol{\Delta}_1^\top\widehat{\boldsymbol{\Sigma}}_1 + n_1/n\boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_1 \\ \vdots \\ n_K/n\boldsymbol{\Delta}_K^\top\widehat{\boldsymbol{\Sigma}}_K + n_K/n\boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_K \\ \sum_{k\in\mathcal{A}} n_k/n\boldsymbol{\Delta}_k^\top\widehat{\boldsymbol{\Sigma}}_k + \sum_{k\in\mathcal{A}\cup\{0\}} n_k/n\boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_k \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\Delta}_1 \\ \vdots \\ \boldsymbol{\Delta}_K \\ \boldsymbol{\Delta}_0 \end{bmatrix}$$

$$= \sum_{k\in\mathcal{A}} \frac{n_k}{n}\left\{\boldsymbol{\Delta}_k^\top\widehat{\boldsymbol{\Sigma}}_k\boldsymbol{\Delta}_k + 2\boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_k\boldsymbol{\Delta}_k + \boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_k\boldsymbol{\Delta}_0\right\} + \frac{n_0}{n}\boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_0\boldsymbol{\Delta}_0$$

$$= \sum_{k\in\mathcal{A}} \frac{1}{n}\|\mathbf{X}_k\boldsymbol{\Delta}_k + \mathbf{X}_k\boldsymbol{\Delta}_0\|_2^2 + \frac{n_0}{n}\boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_0\boldsymbol{\Delta}_0$$

$$= \sum_{k\in\mathcal{A}} \frac{n_k}{n}\left(\boldsymbol{\Delta}_0 + \boldsymbol{\Delta}_k\right)^\top\widehat{\boldsymbol{\Sigma}}_k\left(\boldsymbol{\Delta}_0 + \boldsymbol{\Delta}_k\right) + \frac{n_0}{n}\boldsymbol{\Delta}_0^\top\widehat{\boldsymbol{\Sigma}}_0\boldsymbol{\Delta}_0$$

$$\geq \sum_{k\in\mathcal{A}}\left(v''\|\boldsymbol{\Delta}_k + \boldsymbol{\Delta}_0\|_2^2 - \tau_k\sqrt{\frac{n_k\log p}{n^2}}\|\boldsymbol{\Delta}_k + \boldsymbol{\Delta}_0\|_1\right) + v_0'\|\boldsymbol{\Delta}_0\|_2^2 - \tau_0\sqrt{\frac{n_0\log p}{n^2}}\|\boldsymbol{\Delta}_0\|_1,$$

where $v'' = \min_k v_k n_k/n$, $v' = v''/2$, $v_0' = (2K+1)v'$, and the last inequality follows the RSC conditions on $\widehat{\mathbf{\Sigma}}_k$. In the context of our model, we replace $\mathbf{\Delta}$ by $\widehat{\mathbf{\Delta}} = \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}$. We observe

$$\begin{aligned} v'\|\widehat{\mathbf{\Delta}}\|_2^2 &= v' \sum_{k\in\mathcal{A}} \|\widehat{\mathbf{\Delta}}_k\|^2 + v'\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|^2 \\ &= v' \sum_{k\in\mathcal{A}} \|\widehat{\mathbf{\Delta}}_k + \widehat{\mathbf{\Delta}}_0 - \widehat{\mathbf{\Delta}}_0\|^2 + v'\|\widehat{\mathbf{\Delta}}_0\|^2 \\ &\le 2v' \sum_{k\in\mathcal{A}} \left(\|\widehat{\mathbf{\Delta}}_k + \widehat{\mathbf{\Delta}}_0\|^2 + \|\widehat{\mathbf{\Delta}}_0\|^2\right) + v'\|\widehat{\mathbf{\Delta}}_0\|^2 \\ &= v'' \sum_{k\in\mathcal{A}} \|\widehat{\mathbf{\Delta}}_k + \widehat{\mathbf{\Delta}}_0\|_2^2 + v_0'\|\widehat{\mathbf{\Delta}}_0\|_2^2. \end{aligned} \tag{4}$$

Let $\tau_k = \tau$ for $k \in \mathcal{A}$ and $\tau_0 = \tau(K+1)$. Then, we can also derive

$$\begin{aligned} &\tau \sum_{k\in\mathcal{A}} \sqrt{\frac{n_k \log p}{n^2}} \|\widehat{\mathbf{\Delta}}_k + \widehat{\mathbf{\Delta}}_0\|_1 \le \tau \sum_{k\in\mathcal{A}} \sqrt{\frac{n_m \log p}{n^2}} \left(\|\widehat{\mathbf{\Delta}}_k\|_1 + \|\widehat{\mathbf{\Delta}}_0\|_1\right) \\ =&\tau\sqrt{\frac{n_m \log p}{n^2}}\|\widehat{\mathbf{\Delta}}\|_1 + \tau\sqrt{\frac{n_m \log p}{n^2}}(K-1)\|\widehat{\mathbf{\Delta}}_0\|_1 \\ \le&\tau\sqrt{\frac{n_m \log p}{n^2}}\|\widehat{\mathbf{\Delta}}\|_1 + \tau K\sqrt{\frac{n_m \log p}{n^2}}\|\widehat{\mathbf{\Delta}}\|_1 = \tau_0\sqrt{\frac{n_m \log p}{n^2}}\|\widehat{\mathbf{\Delta}}\|_1 \end{aligned} \tag{5}$$

$$\tau\sqrt{\frac{n_0 \log p}{n^2}}\|\widehat{\mathbf{\Delta}}_0\|_1 \le \tau_0\sqrt{\frac{n_0 \log p}{n^2}}\|\widehat{\mathbf{\Delta}}\|_1. \tag{6}$$

Finally, combining inequalities (4), (5), and (6), we have

$$\widehat{\mathbf{\Delta}}^\top \widehat{\mathbf{\Sigma}}\widehat{\mathbf{\Delta}} \ge v'\|\widehat{\mathbf{\Delta}}\|_2^2 - \tau_0\left(\sqrt{\frac{n_m \log p}{n^2}} + \sqrt{\frac{n_0 \log p}{n^2}}\right)\|\widehat{\mathbf{\Delta}}\|_1 \text{ for } \widehat{\mathbf{\Delta}} \in \mathbb{R}^{p^*} \text{ and } \|\widehat{\mathbf{\Delta}}\|_1 \ge 1.$$

According to Lemma 10 of Liu et al. (2022), the aforementioned inequality with $\|\mathbf{\Delta}\|_1 \ge 1$ actually implies

$$\widehat{\mathbf{\Delta}}^\top \widehat{\mathbf{\Sigma}}\widehat{\mathbf{\Delta}} \ge v'\|\widehat{\mathbf{\Delta}}\|_2^2 - \tau_0\left(\sqrt{\frac{n_m \log p}{n^2}} + \sqrt{\frac{n_0 \log p}{n^2}}\right)\|\widehat{\mathbf{\Delta}}\|_1 \text{ for } \widehat{\mathbf{\Delta}} \in \mathbb{R}^{p^*}$$

for a constant $\tau_0 \ge 0$ and $v' > 0$.

Proof of Theorem 2

First, minimizing the regularized loss function is equivalent to minimizing

$$\frac{1}{2}\boldsymbol{\beta}^\top \widehat{\mathbf{\Sigma}}\boldsymbol{\beta} - \frac{1}{n}\mathbf{y}^\top \mathbf{X}\boldsymbol{\beta} + P_\lambda(\boldsymbol{\beta}).$$

Let $\widehat{\mathbf{\Delta}} = \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}$. The first-order condition implies that for any solution $\widehat{\boldsymbol{\beta}}$ in the interior of the constraint set, $\widehat{\mathbf{\Sigma}}\widehat{\boldsymbol{\beta}} - \frac{1}{n}\mathbf{X}^\top\mathbf{y} + \nabla P_\lambda(\widehat{\boldsymbol{\beta}}) = \mathbf{0}$ and therefore

$$\widehat{\mathbf{\Delta}}^\top \widehat{\mathbf{\Sigma}}\widehat{\boldsymbol{\beta}} + \langle \nabla P_\lambda(\widehat{\boldsymbol{\beta}}) - \frac{1}{n}\mathbf{X}^\top\mathbf{y}, \widehat{\mathbf{\Delta}}\rangle = 0. \tag{7}$$

For simplicity, we use τ for τ_0. The RSC condition on each $\widehat{\mathbf{\Sigma}}_k$ from Theorem 1 implies

$$\widehat{\mathbf{\Delta}}^\top \widehat{\mathbf{\Sigma}}\widehat{\mathbf{\Delta}} \ge v'\|\widehat{\mathbf{\Delta}}\|_2^2 - \tau\left(\sqrt{\frac{n_m \log p}{n^2}} + \sqrt{\frac{n_0 \log p}{n^2}}\right)\|\widehat{\mathbf{\Delta}}\|_1. \tag{8}$$

Subtracting (7) from (8), we have

$$-\widehat{\boldsymbol{\Delta}}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}-\langle\nabla P_\lambda(\widehat{\boldsymbol{\beta}})-\frac{1}{n}\mathbf{X}^\top\mathbf{y},\widehat{\boldsymbol{\Delta}}\rangle\geq v'\|\widehat{\boldsymbol{\Delta}}\|_2^2-\tau\left(\sqrt{\frac{n_m\log p}{n^2}}+\sqrt{\frac{n_0\log p}{n^2}}\right)\|\widehat{\boldsymbol{\Delta}}\|_1. \tag{9}$$

Since the function $P_{\tau,\lambda}(\mathbf{w})=P_\lambda(\mathbf{w})+\frac{\tau}{2}\|\mathbf{w}\|_2^2$ is convex (Loh and Wainwright, 2015; Liu et al., 2022),

$$-\langle\nabla P_\lambda(\widehat{\boldsymbol{\beta}}),\widehat{\boldsymbol{\Delta}}\rangle\leq P_\lambda(\boldsymbol{\beta})-P_\lambda(\widehat{\boldsymbol{\beta}})+\frac{\tau}{2}\|\widehat{\boldsymbol{\Delta}}\|_2^2. \tag{10}$$

Combining (9) and (10), we have

$$\begin{aligned}
&v'\|\widehat{\boldsymbol{\Delta}}\|_2^2-\tau\left(\sqrt{\frac{n_m\log p}{n^2}}+\sqrt{\frac{n_0\log p}{n^2}}\right)\|\widehat{\boldsymbol{\Delta}}\|_1\\
\leq&-\widehat{\boldsymbol{\Delta}}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}+\frac{1}{n}\mathbf{X}^\top\mathbf{y}\widehat{\boldsymbol{\Delta}}+P_\lambda(\boldsymbol{\beta})-P_\lambda(\widehat{\boldsymbol{\beta}})+\tau/2\|\widehat{\boldsymbol{\Delta}}\|_2^2
\end{aligned}$$

$$\begin{aligned}
v'\|\widehat{\boldsymbol{\Delta}}\|_2^2-\tau/2\|\widehat{\boldsymbol{\Delta}}\|_2^2\leq&P_\lambda(\boldsymbol{\beta})-P_\lambda(\widehat{\boldsymbol{\beta}})+\left(\left\|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}-\frac{1}{n}\mathbf{X}^\top\mathbf{y}\right\|_\infty\right)\|\widehat{\boldsymbol{\Delta}}\|_1\\
&+\tau\left(\sqrt{\frac{n_m\log p}{n^2}}+\sqrt{\frac{n_0\log p}{n^2}}\right)\|\widehat{\boldsymbol{\Delta}}\|_1
\end{aligned}$$

$$\begin{aligned}
&v'\|\widehat{\boldsymbol{\Delta}}\|_2^2-\tau/2\|\widehat{\boldsymbol{\Delta}}\|_2^2\\
\leq&P_\lambda(\boldsymbol{\beta})-P_\lambda(\widehat{\boldsymbol{\beta}})+\left\{\left\|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}-\frac{1}{n}\mathbf{X}^\top\mathbf{y}\right\|_\infty+\tau\left(\sqrt{\frac{n_m\log p}{n^2}}+\sqrt{\frac{n_0\log p}{n^2}}\right)\right\}\|\widehat{\boldsymbol{\Delta}}\|_1
\end{aligned}$$

Next, we only need to bound $\|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}-\frac{1}{n}\mathbf{X}^\top\mathbf{y}\|_\infty$. Note that

$$\begin{aligned}
&\left\|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}-\frac{1}{n}\mathbf{X}^\top\mathbf{y}\right\|_\infty=\left\|\frac{1}{n}\mathbf{X}^\top\boldsymbol{\epsilon}\right\|_\infty\\
\leq&\left\|\frac{2}{n}\sum_{k\in\mathcal{A}}\mathbf{X}_k^\top\boldsymbol{\epsilon}_k\right\|_\infty+\left\|\frac{1}{n}\mathbf{X}_0^\top\boldsymbol{\epsilon}_0\right\|_\infty\\
\leq&c_1\sqrt{\frac{n_{\mathcal{A}}\log p}{n^2}}+c_2\sqrt{\frac{n_0\log p}{n^2}}
\end{aligned}$$

for some constants c_1 and c_2 with probability at least $1-4p^{-1}$. The last inequality follows the fact that the product of sub-Gaussian random variables is a sub-exponential random variable. Therefore, $x_{ij}\epsilon_i$ is sub-exponential according to condition C2. Using Lemma 1 with $\mathbf{a}=[1,\cdots,1]^\top$, we have

$$P\left(\frac{2}{n}\left\|\sum_{k\in\mathcal{A}}\mathbf{X}_k^\top\boldsymbol{\epsilon}_k\right\|_\infty>t\right)\leq 2p\max_{j\leq p,k\in\mathcal{A}}\exp\left\{-c\min\left(\frac{n^2t^2}{4M_k^2n_{\mathcal{A}}},\frac{nt}{2M_k}\right)\right\},$$

where $M_k=\max_{1\leq i\leq n_k}\|x_{ki}\|_{\psi_1}$. With $\log p=o(n_{\mathcal{A}})$ and $t=c_1\sqrt{n_{\mathcal{A}}\log p/n^2}$, we have

$$P\left(\frac{2}{n}\left\|\sum_{k\in\mathcal{A}}\mathbf{X}_k^\top\boldsymbol{\epsilon}_k\right\|_\infty\leq c_1\sqrt{\frac{n_{\mathcal{A}}\log p}{n^2}}\right)\geq 1-2p^{-1}$$

for some constant c_1. Similarly, we have

$$P\left(\frac{1}{n}\left\|\mathbf{X}_0^\top\boldsymbol{\epsilon}_0\right\|_\infty\leq c_2\sqrt{\frac{n_0\log p}{n^2}}\right)\geq 1-2p^{-1}$$

for some constant c_2. The last inequality follows by combining the aforementioned two inequalities such that

$$\left\|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}-\frac{1}{n}\mathbf{X}^\top\mathbf{y}\right\|_\infty \leq c_1\sqrt{\frac{n_{\mathcal{A}}\log p}{n^2}}+c_2\sqrt{\frac{n_0\log p}{n^2}} \asymp \sqrt{\frac{\log p}{n}}$$

with probability at least $1-4p^{-1}$. Then

$$\left\|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}-\frac{1}{n}\mathbf{X}^\top\mathbf{y}\right\|_\infty + \tau\left(\sqrt{\frac{n_m\log p}{n^2}}+\sqrt{\frac{n_0\log p}{n^2}}\right) \leq c_1\sqrt{\frac{\log p}{n}},$$

for large enough c_1.

Let $\lambda = 2c_1\sqrt{\frac{\log p}{n}}$, we have

$$\begin{aligned} v'\|\widehat{\boldsymbol{\Delta}}\|_2^2 - \tau/2\|\widehat{\boldsymbol{\Delta}}\|_2^2 &\leq P_\lambda(\boldsymbol{\beta}) - P_\lambda(\widehat{\boldsymbol{\beta}}) + \lambda/2\|\widehat{\boldsymbol{\Delta}}\|_1 \\ &\leq P_\lambda(\boldsymbol{\beta}) - P_\lambda(\widehat{\boldsymbol{\beta}}) + 1/2P_\lambda(\widehat{\boldsymbol{\Delta}}) + \tau/4\|\widehat{\boldsymbol{\Delta}}\|_2^2 \\ &\leq P_\lambda(\boldsymbol{\beta}) - P_\lambda(\widehat{\boldsymbol{\beta}}) + 1/2P_\lambda(\boldsymbol{\beta}) + 1/2P_\lambda(\widehat{\boldsymbol{\beta}}) + \tau/4\|\widehat{\boldsymbol{\Delta}}\|_2^2, \end{aligned}$$

where the second inequality follows Lemma 2. With the second inequality in Lemma 2, we finally have

$$2v'\|\widehat{\boldsymbol{\Delta}}\|_2^2 - 3\tau/2\|\widehat{\boldsymbol{\Delta}}\|_2^2 \leq 3\lambda L\|\widehat{\boldsymbol{\Delta}}_{\mathcal{I}}\|_1 - \lambda L\|\widehat{\boldsymbol{\Delta}}_{\mathcal{I}^c}\|_1.$$

Besides,

$$\begin{aligned} \|\widehat{\boldsymbol{\Delta}}_{\mathcal{I}^c}\|_1 &= \sum_{k\in\mathcal{A}}\left\|\left[\widehat{\boldsymbol{\beta}_k-\boldsymbol{\beta}}_0-(\boldsymbol{\beta}_k-\boldsymbol{\beta}_0)\right]_{\mathcal{I}^c}\right\|_1 + \left\|(\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_0)_{\mathcal{I}^c}\right\|_1 \\ &\geq \sum_{k\in\mathcal{A}}\left\|(\widehat{\boldsymbol{\beta}_k-\boldsymbol{\beta}}_0)_{\mathcal{I}^c}\right\|_1 - \sum_{k\in\mathcal{A}}\|(\boldsymbol{\beta}_k-\boldsymbol{\beta}_0)_{\mathcal{I}^c}\|_1 + \left\|(\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_0)_{\mathcal{I}^c}\right\|_1 \\ &\geq \sum_{k\in\mathcal{A}}\left\|(\widehat{\boldsymbol{\beta}_k-\boldsymbol{\beta}}_0)_{\mathcal{I}^c}\right\|_1 - Kh + \left\|(\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_0)_{\mathcal{I}^c}\right\|_1, \end{aligned} \tag{11}$$

which implies

$$-\lambda L\|\widehat{\boldsymbol{\Delta}}_{\mathcal{I}^c}\|_1 \leq -\lambda L\sum_{k\in\mathcal{A}}\left\|(\widehat{\boldsymbol{\beta}_k-\boldsymbol{\beta}}_0)_{\mathcal{I}^c}\right\|_1 + \lambda LKh - \lambda L\left\|(\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_0)_{\mathcal{I}^c}\right\|_1. \tag{12}$$

With Theorem 1, Eq. (11), and Eq. (12), we obtain

$$\begin{aligned} &2v'\|\widehat{\boldsymbol{\Delta}}\|_2^2 - 3\tau/2\|\widehat{\boldsymbol{\Delta}}\|_2^2 \leq 3\lambda L\|\widehat{\boldsymbol{\Delta}}_{\mathcal{I}}\|_1 - \lambda L\|\widehat{\boldsymbol{\Delta}}_{\mathcal{I}^c}\|_1 \\ \leq& 3\lambda L\|\widehat{\boldsymbol{\Delta}}_{\mathcal{I}}\|_1 + \lambda LKh \\ \lesssim& 3\lambda\sqrt{s}\|\widehat{\boldsymbol{\Delta}}\|_2 + \lambda h. \end{aligned}$$

Let $a = 2v' - 3\tau/2$ for simplicity. We have

$$a\|\widehat{\boldsymbol{\Delta}}\|_2^2 \lesssim 3\lambda\sqrt{s}\|\widehat{\boldsymbol{\Delta}}\|_2 + \lambda h.$$

Let $x = \|\widehat{\boldsymbol{\Delta}}\|_2$, then we solve the quadratic inequality $ax^2 - 3\lambda\sqrt{s}x - \lambda h \lesssim 0$ and we have

$$\|\widehat{\boldsymbol{\Delta}}\|_2 \lesssim \lambda\sqrt{s} + \sqrt{\lambda h}.$$

Plugging in the choice of λ, we have

$$\|\widehat{\boldsymbol{\Delta}}\|_2 \lesssim \sqrt{\frac{s\log p}{n}} + \left(\frac{\log p}{n}\right)^{1/4}\sqrt{h}.$$

Since $\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0$ is a subset of $\widehat{\boldsymbol{\Delta}}$, this result also holds for $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|_2$, i.e.,

$$\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|_2 \lesssim \sqrt{\frac{s\log p}{n}} + \left(\frac{\log p}{n}\right)^{1/4}\sqrt{h}.$$

Immediately from the ℓ_2 error of $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|_2$, we have

$$\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|_1 \lesssim s\sqrt{\frac{\log p}{n}} + \left(\frac{\log p}{n}\right)^{1/4}\sqrt{sh}.$$

Proof of Theorem 3

For simplicity, we drop the subscript v in the testing data $(\mathbf{X}_v, \mathbf{y}_v)$. Let $\mathcal{L}_n(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2$ and $\widehat{\boldsymbol{\Delta}}_0 = \widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0$, then the prediction error is

$$\langle \nabla\mathcal{L}_n(\widehat{\boldsymbol{\beta}}_0) - \nabla\mathcal{L}_n(\boldsymbol{\beta}_0), \widehat{\boldsymbol{\Delta}}_0\rangle = \frac{1}{n}\left\|\mathbf{X}(\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0)\right\|_2^2 = (\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0)^\top\widehat{\boldsymbol{\Sigma}}(\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0) = \widehat{\boldsymbol{\Delta}}_0^\top\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Delta}}_0.$$

Assume the RSC condition on the test data such that $\boldsymbol{\Delta}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Delta} \geq v\|\boldsymbol{\Delta}\|_2^2 - \tau\sqrt{\log p/n}\|\boldsymbol{\Delta}\|_1$ for any $\boldsymbol{\Delta} \in \mathbb{R}^p$. Similar to the proof of Theorem 2, we have

$$-\langle \nabla P_\lambda(\widehat{\boldsymbol{\beta}}_0), \widehat{\boldsymbol{\Delta}}_0\rangle \leq P_\lambda(\boldsymbol{\beta}_0) - P_\lambda(\widehat{\boldsymbol{\beta}}_0) + \tau/2\|\widehat{\boldsymbol{\Delta}}_0\|_2^2.$$

The first-order condition implies

$$\langle \nabla\mathcal{L}_n(\widehat{\boldsymbol{\beta}}_0) + \nabla P_\lambda(\widehat{\boldsymbol{\beta}}_0), -\widehat{\boldsymbol{\Delta}}_0\rangle \geq 0.$$

Therefore, the prediction error

$$\begin{aligned}&\langle \nabla\mathcal{L}_n(\widehat{\boldsymbol{\beta}}_0) - \nabla\mathcal{L}_n(\boldsymbol{\beta}_0), \widehat{\boldsymbol{\Delta}}_0\rangle \leq \langle -\nabla\mathcal{L}_n(\boldsymbol{\beta}_0) - \nabla P_\lambda(\widehat{\boldsymbol{\beta}}_0), \widehat{\boldsymbol{\Delta}}_0\rangle\\ \leq& P_\lambda(\boldsymbol{\beta}_0) - P_\lambda(\widehat{\boldsymbol{\beta}}_0) + \tau/2\|\widehat{\boldsymbol{\Delta}}_0\|_2^2 + \|\nabla\mathcal{L}_n(\boldsymbol{\beta}_0)\|_\infty\|\widehat{\boldsymbol{\Delta}}_0\|_1.\end{aligned}$$

Let $\mathcal{M}$ be the support set of $\boldsymbol{\beta}$, i.e., $\mathcal{M} = \{j : \beta_j \neq 0\}$. Next, we bound $P_\lambda(\boldsymbol{\beta}_0) - P_\lambda(\widehat{\boldsymbol{\beta}}_0)$ by

$$\begin{aligned}&P_\lambda(\boldsymbol{\beta}_0) - P_\lambda(\widehat{\boldsymbol{\beta}}_0) = P_\lambda(\boldsymbol{\beta}_0) - P_\lambda(\widehat{\boldsymbol{\beta}}_{0\mathcal{M}}) - P_\lambda(\widehat{\boldsymbol{\beta}}_{0\mathcal{M}^c})\\ \leq& P_\lambda(\widehat{\boldsymbol{\Delta}}_{0\mathcal{M}}) - P_\lambda(\widehat{\boldsymbol{\beta}}_{0\mathcal{M}^c})\\ =& P_\lambda(\widehat{\boldsymbol{\Delta}}_{0\mathcal{M}}) - P_\lambda(\widehat{\boldsymbol{\Delta}}_{0\mathcal{M}^c})\\ \leq& \lambda L(\|\widehat{\boldsymbol{\Delta}}_{0\mathcal{M}}\|_1 - \|\widehat{\boldsymbol{\Delta}}_{0\mathcal{M}^c}\|_1)\\ \leq& \lambda L\|\widehat{\boldsymbol{\Delta}}_0\|_1.\end{aligned}$$

Together with the result $\|\nabla\mathcal{L}_n(\boldsymbol{\beta}_0)\|_\infty \lesssim \lambda$ (from the proof of Theorem 1 or Loh and Wainwright (2015)), we have

$$\begin{aligned}&\langle \nabla\mathcal{L}_n(\widehat{\boldsymbol{\beta}}_0) - \nabla\mathcal{L}_n(\boldsymbol{\beta}_0), \widehat{\boldsymbol{\Delta}}_0\rangle\\ \lesssim& \lambda L\|\widehat{\boldsymbol{\Delta}}_0\|_1 + \frac{\tau}{2}\|\widehat{\boldsymbol{\Delta}}_0\|_2^2 + \lambda\|\widehat{\boldsymbol{\Delta}}_0\|_1\\ \lesssim& \lambda\sqrt{s}\|\widehat{\boldsymbol{\Delta}}_0\|_2 + \|\widehat{\boldsymbol{\Delta}}_0\|_2^2.\end{aligned}$$

The result follows by plugging in the ℓ_2 error bound in Theorem 2 such that

$$\frac{1}{n}\left\|\mathbf{X}(\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0)\right\|_2^2 \lesssim \frac{s\log p}{n} + \left(\frac{\log p}{n}\right)^{3/4}\sqrt{sh} + h\sqrt{\frac{\log p}{n}}.$$

6.1 Proof of Theorem 4

We decompose $\hat{U}_{n_k}$ by

$$\begin{aligned}\hat{U}_{n_k} =& \underbrace{\frac{1}{n_k}\sum_{i\neq i'}^{n_k}\left\{(y_i - \mu_i)(y_{i'} - \mu_{i'})\mathbf{x}_{ki}^\top\mathbf{x}_{ki'}\right\}}_{I_{\hat{U}_{n_k}}} + \underbrace{\frac{1}{n_k}\sum_{i\neq i'}^{n_k}\left\{(\mu_i - \hat{\mu}_{\varnothing i})(\mu_{i'} - \hat{\mu}_{\varnothing i'})\mathbf{x}_{ki}^\top\mathbf{x}_{ki'}\right\}}_{II_{\hat{U}_{n_k}}}\\ &+ \underbrace{\frac{2}{n_k}\sum_{i\neq i'}^{n_k}\left\{(y_i - \mu_i)(y_{i'} - \hat{\mu}_{\varnothing i'})\mathbf{x}_{ki}^\top\mathbf{x}_{ki'}\right\}}_{III_{\hat{U}_{n_k}}}.\end{aligned}$$

Note that the size is proved under H_0, We first exam $II_{\hat{U}_{n_k}}$: note that

$$\frac{II_{\hat{U}_{n_k}}}{n_k} = \underbrace{(\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0)^\top \left[\frac{1}{n_k}\sum_{i=1}^{n_k} \mathbf{z}_i \mathbf{w}_i^\top\right] \left[\frac{1}{n_k}\sum_{i=1}^{n_k} \mathbf{z}_i \mathbf{w}_i^\top\right] (\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0)}_{\text{II1}} - \underbrace{\frac{1}{n_k^2}\sum_{i=1}^{n_k} (\mu_i - \hat{\mu}_{\varnothing i})^2 \mathbf{w}_i^\top \mathbf{w}_i}_{\text{II2}}.$$

For II1, let $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n_k}\sum_{i=1}^{n_k} \mathbf{z}_i \mathbf{w}_i^\top = \frac{1}{n_k}\sum_{i=1}^{n_k} \mathbf{x}_{ki}\mathbf{x}_{ki}^\top$ and $\boldsymbol{\Sigma} = E(\mathbf{x}_{ki}\mathbf{x}_{ki}^\top)$. Then, it can be shown that

$$\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\|_\infty = \tau = O_p\left(\sqrt{\frac{\log p}{n_k}}\right).$$

Similar to A2 in Chen et al. (2022), we see that $|II1| = O_p(\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_0\|_2^2) = O_p\left(\frac{s\log p}{n}\right)$, where $n = n_0 + n_k$.

For II2, $n_k II2 \le \|\mu - \hat{\mu}_{\varnothing}\|_\infty^2 \frac{1}{n_k}\sum_{i=1}^{n_k} \mathbf{x}_{ki}^\top \mathbf{x}_{ki} = o_p(\sqrt{2\Lambda_W^\epsilon})$.

Finally,

$$II_{\hat{U}_{n_k}} = n_k II1 + n_k II2 = O_p(n_k s\log p/n) + o_p(\sqrt{2\Lambda_W^\epsilon}) = o_p(\sqrt{2\Lambda_W^\epsilon})$$

when $n_k s\log p/n/\sqrt{2\Lambda_W^\epsilon} = o(1)$.

We next exam $III_{\hat{U}_{n_k}}$: Similar to Chen et al. (2022), we obtain $|III1| = O_p[\frac{1}{\sqrt{nn_k}}\sqrt{s\log p}(2\Lambda_W^\epsilon)^{1/4}]$ and $n_k III2 = o_p(\sqrt{2\Lambda_W^\epsilon})$. Finally,

$$III_{\hat{U}_{n_k}} = n_k III1 + n_k III2 = O_p\left[\sqrt{\frac{n_k s\log p}{n}}(2\Lambda_W^\epsilon)^{1/4}\right] + o_p\left(\sqrt{2\Lambda_W^\epsilon}\right) = o_p\left(\sqrt{2\Lambda_W^\epsilon}\right)$$

when $n_k s\log p/n/\sqrt{2\Lambda_W^\epsilon} = o(1)$. Remaining steps are the same as Chen et al. (2022).