TransFusion: Covariate-Shift Robust Transfer Learning for High-Dimensional Regression

Zelin He
Dept. of Statistics
Penn State Univ.

Ying Sun
School of EECS,
Penn State Univ.

Jingyuan Liu
Dept. of Statistics
and Data Science,
SOE, Xiamen Univ.

Runze Li
Dept. of Statistics,
Penn State Univ.

Abstract

The main challenge that sets transfer learning apart from traditional supervised learning is the distribution shift, reflected as the shift between the source and target models and that between the marginal covariate distributions. In this work, we tackle model shifts in the presence of covariate shifts in the high-dimensional regression setting. Specifically, we propose a two-step method with a novel fused-regularizer that effectively leverages samples from source tasks to improve the learning performance on a target task with limited samples. Nonasymptotic bound is provided for the estimation error of the target model, showing the robustness of the proposed method to covariate shifts. We further establish conditions under which the estimator is minimax-optimal. Additionally, we extend the method to a distributed setting, allowing for a pretraining-finetuning strategy, requiring just one round of communication while retaining the estimation rate of the centralized version. Numerical tests validate our theory, highlighting the method's robustness to covariate shifts.

Transfer learning is a technique that leverages knowledge from source tasks to improve learning performance in a related but possibly different target task (Torrey and Shavlik, 2010). In this paper, we consider the high-dimensional setting where the target sample size is much smaller than the number of features. In this context, applying transfer learning techniques to extract information from a larger pool of source samples can be particularly beneficial in identifying the model parameters. For example, in genetic studies of rare diseases, transferring information from larger, related set of studies could uncover highly disease-relevant genetic patterns (Taroni et al., 2019).

In contrast to learning from i.i.d. samples, a fundamental challenge in transfer learning is handling the distribution shifts between the source sample $(\mathbf{X}_S, \mathbf{y}_S)$ and the target sample $(\mathbf{X}_T, \mathbf{y}_T)$ (Pan and Yang, 2009). The discrepancy between the distributions typically shows in two ways: 1) *model shift:* $P(\mathbf{y}_S|\mathbf{X}) \neq P(\mathbf{y}_T|\mathbf{X})$, indicating a shift in the learning models, and 2) *covariate shift:* $P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$, indicating a shift in the marginal covariate distributions. In either case, models achieving small training errors on the source tasks may experience high risks on the target task (Lu et al., 2020). Therefore, to improve the learning performance of the target model $P(\mathbf{y}_T|\mathbf{X})$ using knowledge from the source samples, one should not only capture and correct the model shift but also be robust to the covariate shifts. In the high-dimensional setting, handling such differences becomes even more difficult due to accumulated noise and limited samples (Fan et al., 2020). This leads to the following question:

How to tackle *model shifts* in *high-dimensional* transfer learning while being robust to *covariate shifts*?

Apart from the challenge brought by the distribution shift, modern learning problems often involve datasets distributed across multiple computing nodes. In such a scenario, performing centralized training by pooling all the raw data in a single machine can be undesirable due to storage, communication, and privacy issues. This situation prompts another question:

How to transfer knowledge from *distributed* source datasets in a *communication-efficient* manner?

This paper proposes a solution for the above two questions for high-dimensional linear regression with K

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

[1] Correspondence to: J Liu <jingyuan@xmu.edu.cn>.

source tasks, where the target model is p-dimensional with sparsity level s. Our contributions are:

• **Covariate-Shift Robust Regularizer.** We propose a novel fused-regularizer achieving two purposes: it promotes sparse solutions for the high-dimensional model parameter while simultaneously capturing model shifts between source and target datasets. Our theoretical results further show that this regularizer can separate model shifts from shared patterns in a robust manner under covariate shifts.

• **Optimal Estimation Procedure.** Leveraging the proposed regularizer, we introduce a two-step procedure termed TransFusion. When the source tasks are sufficiently diverse, we show applying the first step on the source and target tasks jointly suffices to yield a fast rate of $O(\frac{s \log p}{n_T + K n_S} + \bar{h}\sqrt{\frac{\log p}{n_S}})$, where n_T is the target sample size, n_S is the source sample size, and $\bar{h}$ measures task similarity. The rate significantly improves over the one achieved on target task without transfer learning, i.e., the rate of $O(\frac{s \log p}{n_T})$, when $n_S \gg \bar{h} n_T^2$. For cases that do not meet the diversity criteria, TransFusion incorporates a second step refining the estimate on the target task, ensuring a rate of $O(\frac{s \log p}{n_T + K n_S} + \bar{h}\sqrt{\frac{\log p}{n_T}} \wedge \bar{h}^2)$, which is minimax-optimal when $p \gg s$ and $\bar{h}$ is relatively small.

• **Efficient Distributed Learning.** We develop a distributed variant of our method, termed D-TransFusion, requiring only one-shot communication of the pre-trained local models from source tasks nodes to target task node, significantly reducing communication overhead. More importantly, it offers the flexibility to quickly adapt the models to different downstream tasks while avoiding training from scratch. We further show that when the source sample size n_S is sufficiently large, D-TransFusion achieves the optimal statistical rate, matching its centralized counterpart.

Related Works: This paper develops transfer learning methods for high-dimensional regression problems under both model and covariate shifts. Related works can be broadly divided into the following categories.

Domain Adaptation methods primarily focus on handling covariate shifts, usually assuming the underlying models remain the same (Quinonero-Candela et al. (2008), Redko et al. (2020)). One prevalent approach in this category focuses on aligning the source and target covariate distributions by learning domain-invariant representations (Redko et al. (2020), Mansour et al. (2009), Cortes and Mohri (2011), Cortes and Mohri (2014)). Another line of research involves correcting estimators to address covariate shifts, often using the importance weighting (Quinonero-Candela et al. (2008), Sugiyama and Kawanabe (2012), Chen et al. (2016)). In contrast, we explicitly address model shifts and aim for robustness to covariate shifts.

Multitask learning aims to handle model shifts across multiple tasks and learn shared features to improve the performance of each task (Pan and Yang, 2009). In regression settings, regularization techniques are often employed to promote information transfer. Examples include the Frobenius and spectral norm (Argyriou et al. (2007), Tian et al. (2023)), mixed $\ell_{2,1}$ norm (Lounici et al., 2009), hard-thresholding (Huang et al., 2023), and the total variation norm (Li and Sang (2019), Zhang et al. (2022), Tang and Song (2016)). These works typically require all tasks to have a comparable sample size and emphasize overall task performance. Therefore, they are not directly applicable to transfer learning problems where the target task, often with far fewer samples, is the primary focus.

Transfer Learning has been intensively studied under regression settings (Du et al. (2017), Lei et al. (2021), Lin and Reimherr (2022;2024)). However, most works are restricted to low-dimensional problems. Recently, transfer learning in the high-dimensional regression settings has been studied in Takada and Fujisawa (2020), Bastani (2021), Li et al. (2022) and Tian and Feng (2022). These works, however, deal with scenarios with only a single source or are sensitive to covariate shifts across multiple sources, and their learning accuracy degrades quickly if such shifts are severe. More recent works such as Li et al. (2023) and Liu (2023) attempt to mitigate the impact of covariate shifts. However, these methods either rely on strong assumptions or are computationally demanding. Specifically, Li et al. (2023) established the convergence rate of the proposed estimator assuming the empirical loss function in the high-dimensional setting is smooth, and computing the estimator requires solving a nonsmooth optimization problem with multiple constraints, while Liu (2023) assumes the target sample has a comparable size as the source sample. In contrast, theoretical guarantees of our method are established under weaker, more practical conditions, and numerically it can be computed efficiently using algorithms such as iterative soft thresholding.

Notation: We use bold upper- and lowercase letters for matrices and vectors, respectively. For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we denote its (i, j)-th element by $\mathbf{A}_{ij}$, maximum eigenvalue by $\Lambda_{\max}(\mathbf{A})$, and minimum eigenvalue by $\Lambda_{\min}(\mathbf{A})$. We let $a \vee b$ denote $\max\{a, b\}$ and $a \wedge b$ denote $\min\{a, b\}$. We use $c, c_0, c_1, \ldots$ to denote generic constants independent of n, p and K. Let $a_n = O(b_n)$ and $a_n \lesssim b_n$ denote $|a_n/b_n| \leq c$ for some constant c when n is large enough; $a_n = o(b_n)$ and $b_n \gg a_n$ if $a_n = O(c_n b_n)$ for some $c_n \to 0$; $a \asymp b$ if $a = O(b)$ and $b = O(a)$.

1 Preliminaries

We consider a transfer learning problem involving one target task and K source tasks. For the target task, we observe a sample $(\mathbf{X}^{(0)}, \mathbf{y}^{(0)})$ generated from the target model

$$\mathbf{y}_i^{(0)} = \left(\mathbf{X}_{i\cdot}^{(0)}\right)^\top \boldsymbol{\beta}^{(0)} + \boldsymbol{\epsilon}_i^{(0)}, \quad i = 1, \ldots, n_T,$$

where $\boldsymbol{\beta}^{(0)} \in \mathbb{R}^p$ is the parameter of primary interest and $\boldsymbol{\epsilon}_i^{(0)}$ is the observation noise. We focus on a high-dimensional scenario where the dimension p is much larger than the target sample size n_T, yet the ground truth $\boldsymbol{\beta}^{(0)}$ is a sparse vector with $s := \|\boldsymbol{\beta}^{(0)}\|_0$ nonzero elements, which is much smaller than p, i.e., $p \gg s$.

In addition to the target sample, we also have access to K source samples $\{(\mathbf{X}^{(k)}, \mathbf{y}^{(k)})\}_{k=1}^K$, generated from the source model

$$\mathbf{y}_i^{(k)} = \left(\mathbf{X}_{i\cdot}^{(k)}\right)^\top \boldsymbol{\beta}^{(k)} + \boldsymbol{\epsilon}_i^{(k)}, \; i = 1, \ldots, n_S, \; k = 1, \ldots, K.$$

For the k-th source model, $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^p$ is the unknown source task-specific parameter, and $\boldsymbol{\epsilon}_i^{(k)}$ accounts for the observation noise. For simplicity, we assume the source samples have the same size n_S.

Our goal is to estimate $\boldsymbol{\beta}^{(0)}$ using both the target and source samples under the challenging scenario where distributions of the samples are heterogeneous, as characterized by both model and covariate shift described next.

Model Shift. In our context, the model shift is the situation where each source model differs from the target model, and is measured by $\boldsymbol{\delta}^{(k)} := \boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(0)}$ for $1 \le k \le K$. Throughout the paper, we refer $\boldsymbol{\delta}^{(k)}$ as the "parameter contrast" or "task-specific signal". A source task is considered informative for transfer learning if $\boldsymbol{\delta}^{(k)}$ is relatively small. Formally, let $\boldsymbol{\beta} := ((\boldsymbol{\beta}^{(0)})^\top, (\boldsymbol{\beta}^{(1)})^\top, \ldots, (\boldsymbol{\beta}^{(K)})^\top)^\top \in \mathbb{R}^{(K+1)p}$, we assume $\boldsymbol{\beta}$ belongs to the following parameter space

$$\Theta(s, \mathbf{h}) := \left\{\boldsymbol{\beta} : \|\boldsymbol{\beta}^{(0)}\|_0 \le s, \|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(0)}\|_1 \le h_k\right\}, \tag{1}$$

with $\mathbf{h} := (h_1, \ldots, h_K)^\top$. In (1), the informative level of the k-th source task is quantified by the ℓ_1-sparsity of $\boldsymbol{\delta}^{(k)}$, and is upper bounded by a factor $h_k \ge 0$.

Remark 1. *We choose an ℓ_1-sparse constraint for the high-dimensional contrast $\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(0)}$, as it aligns well with practical applications where model shifts typically spread over multiple dimensions but their overall magnitude does not grow too fast. The results in the paper can be naturally extended to a general ℓ_q-sparse case for $q \in [0, 1]$.*

Covariate Shift. In addition to the shift in the model parameters, we also consider the covariate shift, defined as the difference in the distributions of $\mathbf{X}_{i\cdot}^{(k)}$s across the tasks. In this work, we only impose the following mild tail condition on the distribution of $\mathbf{X}_{i\cdot}^{(k)}$s but allow other distribution characteristics, such as the covariance structures, to vary across different tasks.

Assumption 1 (Sub-Gaussian designs)**.** *For any $0 \le k \le K$, $\mathbf{X}_{i\cdot}^{(k)}$s are independent sub-Gaussian random vectors with mean zero and covariance $\boldsymbol{\Sigma}^{(k)}$. Furthermore, there exists some universal constant c such that $1/c \le \min_{0 \le k \le K} \Lambda_{\min}(\boldsymbol{\Sigma}^{(k)}) \le \max_{0 \le k \le K} \Lambda_{\max}(\boldsymbol{\Sigma}^{(k)}) \le c$.*

Finally, we assume that the random noises follow independent Gaussian distributions, a typical assumption for high-dimensional regression analysis.

Assumption 2 (Gaussian random errors)**.** *For all $0 \le k \le K$, the $\boldsymbol{\epsilon}_i^{(k)}$s are independent Gaussian random variables with zero mean and uniformly upper bounded variance, and are independent of $\mathbf{X}^{(k)}$s.*

2 Covariate-Shift Robust Transfer Learning

We now introduce a method called **TransFusion** (**Trans**fer Learning with a **Fu**sed-Regulariza**tion**), designed to address high-dimensional model shifts in the presence of covariate shifts, thereby transferring knowledge from source tasks to the target task. The method consists of two steps. First, we perform a co-training step using both source and target samples, leveraging the ℓ_0-sparsity of $\boldsymbol{\beta}^{(0)}$ and the ℓ_1-sparsity of the contrast $\boldsymbol{\delta}^{(k)}$. We show that when the source tasks are sufficiently diverse, see Definition 1, performing the first step of the TransFusion method suffices to ensure a fast rate. When such a condition is not met, we further perform a second step by fine-tuning the model on the target dataset. The method is shown to be rate-optimal and robust to covariate shifts.

2.1 Step 1: Co-Training

We start with the first step, a co-training step involving both target and source samples. The challenge is tackling the distribution shifts while extracting the shared pattern between source and target samples to estimate $\boldsymbol{\beta}^{(0)}$. Pooling all data as *i.i.d.* samples leverages larger sample size and reduces noise, but suffers from large bias if the source and target distributions differ significantly. In contrast, training exclusively on the target sample prevents any information transfer from the source samples. It is therefore critical to to strike a balance between these two extremes to improve the estimation of $\boldsymbol{\beta}^{(0)}$. To this end, we propose a co-training step that estimates $\boldsymbol{\beta}^{(k)}$s by solving the following problem:

$$\hat{\boldsymbol{\beta}} \in \underset{\boldsymbol{\beta}\in\mathbb{R}^{(K+1)p}}{\operatorname{argmin}} \left\{ \frac{1}{2N}\sum_{k=0}^{K} \|\mathbf{y}^{(k)} - \mathbf{X}^{(k)}\boldsymbol{\beta}^{(k)}\|_2^2 \right. \\ \left. +\lambda_0\Big(\|\boldsymbol{\beta}^{(0)}\|_1 + \sum_{k=1}^{K} a_k\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(0)}\|_1\Big) \right\}, \quad (2)$$

where $N = Kn_S + n_T$ is the total sample size, λ_0 is the tuning parameter and $\{a_k\}_{k=1}^{K}$ are weights that will be specified later. In (2), the first term measures the average fitness of the models with parameter $\{\boldsymbol{\beta}^{(k)}\}_{k=0}^{K}$, while the fused-regularization term simultaneously promotes the sparsity of $\boldsymbol{\beta}^{(0)}$ and captures the ℓ_1-sparse contrast between $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\beta}^{(k)}$ by penalizing their difference.

We construct the first-step estimator as $\hat{\mathbf{w}} = \frac{n_S}{N}\sum_{k=1}^{K}\hat{\boldsymbol{\beta}}^{(k)} + \frac{n_T}{N}\hat{\boldsymbol{\beta}}^{(0)}$. The motivation for this estimator is twofold: first, averaging across both target and source estimators utilizes the full sample, yielding an estimator with low variance. In addition, when the source datasets are sufficiently diverse, the bias of $\hat{\mathbf{w}}$ is small. When the reduction in variance dominates the increase in bias, the one-step estimator $\hat{\mathbf{w}}$ serves as a promising estimator of $\boldsymbol{\beta}^{(0)}$ than using the target sample alone.

We now formally characterize the source task diversity.

Definition 1 (Source task diversity). *Given $\boldsymbol{\beta} \in \Theta(s, \mathbf{h})$, we quantify the diversity across source tasks with the metric $\|\frac{n_S}{N}\sum_{k=1}^{K}\boldsymbol{\delta}^{(k)}\|_1 \le \varepsilon_D$, where $\boldsymbol{\delta}^{(k)} := \boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(0)}$ is the task-specific signal.*

A small ε_D implies that $\{\boldsymbol{\beta}^{(k)}\}_{k=1}^{K}$ are centered around $\boldsymbol{\beta}^{(0)}$ and cover all the directions, such that the average parameter $\mathbf{w} := \frac{n_S}{N}\sum_{k=1}^{K}\boldsymbol{\beta}^{(k)} + \frac{n_T}{N}\boldsymbol{\beta}^{(0)}$ does not align with any direction significantly more than $\boldsymbol{\beta}^{(0)}$. This kind of assumption is commonly imposed in transfer learning settings (Du et al. (2020),Tripuraneni et al. (2020)).

We proceed to establish the statistical estimation rate for $\hat{\mathbf{w}}$ under the setting of diverse source tasks.

Theorem 1. *Under Assumption 1 and 2, if $n_S \gg s\log p$, then by choosing $\lambda_0 = c_0\sqrt{\log p/N}$ for some universal constant c_0 and $a_k = 8\sqrt{n_S/N}$, we have*

$$\|\hat{\mathbf{w}} - \mathbf{w}\|_2^2 \lesssim \frac{s\log p}{N} + (1+v_n)\bar{h}\sqrt{\frac{\log p}{n_S}}, \qquad (3)$$

and

$$\|\hat{\mathbf{w}} - \boldsymbol{\beta}^{(0)}\|_2^2 \lesssim \frac{s\log p}{N} + (1+v_n)\bar{h}\sqrt{\frac{\log p}{n_S}} + \varepsilon_D^2, \quad (4)$$

with probability at least $1 - c_1\exp(-c_2 n_T) - c_3\exp(-c_4\log p)$, where $v_n := \sqrt{K^2\log p/n_S}\bar{h}$ and $\bar{h} := \frac{n_S}{N}\sum_{k=1}^{K}h_k$.

Let us break down the upper bound provided by equation (4). The first term, $s\log p/N$, represents the rate from estimating an s-sparse coefficient $\boldsymbol{\beta}^{(0)}$ based on $N = Kn_S + n_T$ i.i.d. samples. This term reveals the benefit of using both the source and target datasets for estimating the target parameter $\boldsymbol{\beta}^{(0)}$. The second term, $\bar{h}\sqrt{\log p/n_S}$, accounts for the estimation error of $\boldsymbol{\delta}^{(k)}$ unique to each source task and thus is limited by the source sample size n_S. The factor v_n is sample dependent and is negligible when $n_S \gg \bar{h}^2K^2\log p$. The first two terms together quantify the estimation error $\|\hat{\mathbf{w}} - \mathbf{w}\|_2^2$. The third term, ε_D^2, measures the difference between $\mathbf{w}$ and $\boldsymbol{\beta}^{(0)}$ and contributes to the bias introduced by averaging. Notably, to obtain the bound (4), we do not require a homogeneous distribution of the covariates $\mathbf{X}^{(k)}$s but only impose the mild tail assumption as outlined in Assumption 1, and the bound does not depend on the target sample size n_T.

As a comparison, if we apply the LASSO regression on the target data, the estimation error is of order $O(s\log p/n_T)$. Therefore, if $N \gg n_T$, $n_S \gg \bar{h}^2(n_T^2 \vee K^2\log p)$ and $\varepsilon_D \ll \sqrt{s\log p/n_T}$, that is, the source tasks are sufficiently diverse with adequately large sample size, then one-step TransFusion method achieves a sharper estimation rate. This corroborates our design intuition and quantitatively shows the benefit of transferring information from diverse source tasks even under covariate shifts.

Remark 2 (Adaptive version of TransFusion). *In Theorem 1, we choose a weight a_k that does not depend on h_k, as we treat h_k as an unknown priori. As a compromise, the estimation rate depends on $\bar{h}$, the averaged magnitude of model shifts. In fact, for a general choice of λ_0 and a_k, TransFusion could yield a bound*

$$\|\hat{\mathbf{w}} - \boldsymbol{\beta}^{(0)}\|_2^2 \lesssim s\lambda_0^2 + \sum_{k=1}^{K} a_k\lambda_0 h_k + \varepsilon_D^2$$

under certain conditions (cf. Lemma 5). So if we have some information on h_k, we may adjust a_k accordingly, focusing more on informative datasets with small h_k and less or not at all on those with large h_k. In such cases, this adaptive version of TransFusion could potentially yield a fast estimation rate that is less sensitive to the magnitude of model shifts.

Remark 3 (Scalability with task number K). *TransFusion incorporates a novel fused regularizer capturing the task-specific signals in the joint learning step. This technique robustifies the method against covariate-shift* *and introduces a dependency of the convergence rate on K as a tradeoff. Specifically, the convergence rate of the first-step estimator is given by (3) with $v_n := \sqrt{K^2 \log p / n_S \bar{h}}$ due to the non-strong convexity of the local empirical loss (cf. Lemma 4). If we increase K while fixing n_S, for large K, the sum will be dominated by the second term, which grows with K. Otherwise, if we increase n_S with K, TransFusion would have a consistent error improvement. This is supported by the simulation results and discussions in Appendix E.*

2.2 Step 2: Local Debias

Despite its merits, the one-step TransFusion method may experience large bias when ε_D is large. This is especially the case when the source tasks exhibit a skewed model shift towards one specific direction than $\boldsymbol{\beta}^{(0)}$. In such cases, we employ an additional debias step that refines the initial estimator $\hat{\mathbf{w}}$ and mitigates the impact of ε_D. Specifically, we correct $\hat{\mathbf{w}}$ using the target sample as:

$$\hat{\boldsymbol{\delta}} \in \operatorname*{argmin}_{\boldsymbol{\delta}\in\mathbb{R}^p}\left\{\frac{1}{2n_T}\left\|\mathbf{y}^{(0)}-\mathbf{X}^{(0)}\hat{\mathbf{w}}-\mathbf{X}^{(0)}\boldsymbol{\delta}\right\|_2^2+\tilde{\lambda}\|\boldsymbol{\delta}\|_1\right\},$$

$$\hat{\boldsymbol{\beta}}^{(0)}_{\text{TransFusion}}=\hat{\mathbf{w}}+\hat{\boldsymbol{\delta}}. \tag{5}$$

Next, we demonstrate that with an appropriate choice of estimation strategy and tuning parameters, we can attain an optimal estimation rate of $\boldsymbol{\beta}^{(0)}$ without requiring a small ε_D. Define the event

$$A=\{s\log p/n_S \geq \bar{h}\sqrt{\log p/n_T}\}, \tag{6}$$

and A^c as its complement. The following theorem establishes an upper bound on the estimation error for the two-step TransFusion algorithm.

Theorem 2. *Under the assumptions of Theorem 1, if $n_T \gtrsim s\log p$, $n_S \gtrsim K^2 s\log p$ and $\bar{h}\sqrt{\log p/n_T} + Ks\log p/n_S = o(1)$, then by choosing the parameters*

$$\lambda_0 = c_0\left(\sqrt{\frac{\log p}{N}}\mathbb{1}_A+\sqrt{\frac{\log p}{n_S}}\mathbb{1}_{A^c}\right),$$

$$a_k = 8\left(\sqrt{\frac{n_S}{N}}\mathbb{1}_A+\frac{n_S}{N}\mathbb{1}_{A^c}\right),$$

and $\tilde{\lambda} = c_1\sqrt{\log p/n_T}$ for some universal constants c_0 and c_1, the solution of the two-step TransFusion method satisfies

$$\|\hat{\boldsymbol{\beta}}^{(0)}_{\text{TransFusion}}-\boldsymbol{\beta}^{(0)}\|_2^2 \lesssim \frac{s\log p}{N}+\bar{h}\sqrt{\frac{\log p}{n_T}}, \tag{7}$$

with probability at least $1-c_2\exp(-c_3\log p)$.

By comparing the results of Theorem 1 and 2 [cf. (4) and (7)], we see when $\sqrt{\log p/n_T} \lesssim \varepsilon_D^2/\bar{h}$, performing the second step improves the estimation precision. The ratio $\varepsilon_D/\bar{h}$ quantifies the normalized (by the magnitude of $\boldsymbol{\delta}^{(k)}$s) source task diversity, and thus our result shows applying the second step is beneficial for non-diverse source tasks. Note that the condition on the sample size n_T and n_S in Theorem 2 is stronger than Theorem 1. Such a condition is required to ensure the target-specific signals $\boldsymbol{\delta}^{(k)}$ being accurately captured to perform the correction in (5).

On the other hand, if $\sqrt{\log p/n_T} \gtrsim \varepsilon_D^2/\bar{h}$ applying the second step may even harm the model performance. Therefore, choosing between the one-step and two-step TransFusion methods carefully is key to getting the optimal estimation results. The following corollary provides guidelines for making this choice.

Corollary 1. *Under the assumptions of Theorem 2, if we apply the one-step TransFusion method when $n_T \lesssim \log p/\bar{h}^2$ and apply the two-step TransFusion method otherwise, then obtained estimator $\hat{\boldsymbol{\beta}}^{(0)}_{\text{TransFusion}-2}$ satisfies*

$$\|\hat{\boldsymbol{\beta}}^{(0)}_{\text{TransFusion}-2}-\boldsymbol{\beta}^{(0)}\|_2^2 \lesssim \frac{s\log p}{N}+\bar{h}\sqrt{\frac{\log p}{n_T}}\wedge\bar{h}^2, \tag{8}$$

with probability at least $1-c_2\exp(-c_3\log p)$.

Next, we establish the minimax optimality of the above strategy under certain conditions. The following result follows from minor modifications of Theorem 2 in Li et al. (2022).

Proposition 1. *Under Assumption 1 and Assumption 2, if $N \gg s\log p$, $h_k \asymp \bar{h}$ and $\bar{h}\sqrt{\log p/n_T} = o(1)$, then any estimator $\hat{\boldsymbol{\beta}}'$ that is a measurable function of the sample $\{(\mathbf{X}^{(k)},\mathbf{y}^{(k)})\}_{0\leq k\leq K}$ satisfies*

$$\inf_{\hat{\boldsymbol{\beta}}'}\sup_{\boldsymbol{\beta}\in\Theta(s,\mathbf{h})}\left\|\hat{\boldsymbol{\beta}}'-\boldsymbol{\beta}^{(0)}\right\|_2^2 \gtrsim \frac{s\log p}{N}+\frac{s\log p}{n_T}\wedge\bar{h}\sqrt{\frac{\log p}{n_T}}\wedge\bar{h}^2, \tag{9}$$

with probability at least $1/2$.

Comparing with the upper bound (8), we can conclude that, given the conditions outlined in Theorem 2, if source datasets are sufficient informative such that $\bar{h} \lesssim s\sqrt{\log p/n_T}$, then the proposed procedure is minimax optimal, even under covariate shifts.

Remark 4 (Implementation of TransFusion)**.** *Notice that although the two-step TransFusion method only involves one tuning parameter in each step, as discussed in Theorem 2 and Corollary 1, it relies on a dichotomous strategy that depends on the value of $\bar{h}$. However, it can still be practically applied without knowing $\bar{h}$ in advance by implementing both choices and selecting the one with smaller validation error. On the computation front, the global minimizer of each*

TransFusion step can be efficiently found by numerical algorithms such as iterative soft-thresholding. The implementation details are provided in Appendix D.

2.3 Understanding the Robustness of the TransFusion Method to Covariate Shift

In this section, we discuss the underlying mechanisms that make the TransFusion method robust to covariate shifts via comparing with the two-step method proposed in Li et al. (2022) and Tian and Feng (2022). While both methods aim to address the high-dimensional transfer learning problem, we take a significantly different approach in the first co-training step.

In their approach, the first step pools samples from both target and source tasks, and performs a sparse regression to obtain the initial estimator. In the linear regression setting, this estimator comes with an asymptotic bias expressed as

$$\boldsymbol{\delta}_{\text{Pooling}} := \left(\sum_{k=1}^{K} \boldsymbol{\Sigma}^{(k)}\right)^{-1} \sum_{k=1}^{K} \boldsymbol{\Sigma}^{(k)} \boldsymbol{\delta}^{(k)}. \tag{10}$$

Due to the weights introduced by the covariance matrices, the contrast $\boldsymbol{\delta}^{(k)}$s can be amplified by a factor of C_Σ, defined as

$$C_\Sigma := 1 + \max_{j \le p} \max_{k} \left\| e_j^\top \left(\boldsymbol{\Sigma}^{(k)} - \boldsymbol{\Sigma}^{(0)}\right) \left(\sum_{1 \le k \le K} \frac{1}{K} \boldsymbol{\Sigma}^{(k)}\right)^{-1} \right\|_1 .$$

Consequently, in the linear regression setting, their estimator yields the following estimation rate (Li et al., 2022, Theorem 4):

$$\frac{s \log p}{N} + \left(C_\Sigma \sqrt{\frac{\log p}{n_T}} \bar{h}\right) \wedge \left(C_\Sigma^2 \bar{h}^2\right).$$

When the $\boldsymbol{\Sigma}^{(k)}$s are dissimilar, the factor C_Σ can diverge with dimension p even if Assumption 1 holds, considerably deteriorating the estimation accuracy. See Appendix C for a detailed discussion.

In contrast, as shown in (8), our method is robust to such covariance heterogeneity and thus doesn't involve the C_Σ factor. This is achieved by incorporating a fused-regularizer, allowing us to accurately capture task-specific signals under covariate shifts. Solving the objective leads to an initial estimator with asymptotic bias

$$\boldsymbol{\delta}_{\text{TransFusion}} := \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\delta}^{(k)},$$

which is free from the impact of the covariance matrices. This bias is much smaller than $\boldsymbol{\delta}_{\text{Pooling}}$ under covariate shift settings with a large C_Σ.

3 D-TranFusion: Distributed Transfer Learning in One-Shot

In this section, we consider the distributed transfer learning problem where the target and K source samples are stored by different computing nodes. Such a setting is of primary interest in learning problems involving a massive amount of training data, where brute-forcely pooling the raw data is not admissible due to practical constraints such as storage limitation, communication cost, and privacy concerns.

This motivates us to consider developing a communication-efficient distributed TransFusion method, termed D-TransFusion. Our method is based on TransFusion and leverages the idea of divide-and-conquer to facilitate communication efficiency, aiming to achieve a comparable estimation error as TransFusion but using only one-shot communication. Specifically, D-TransFusion consists of the following two steps.

Step 1. Each node k computes an estimator $\tilde{\boldsymbol{\beta}}^{(k)}$ (to be specified later) locally based on source sample $(\mathbf{X}^{(k)}, \mathbf{y}^{(k)})$ and transmits it to the target node. The target node then aggregates them with its own sample $(\mathbf{X}^{(0)}, \mathbf{y}^{(0)})$ via solving the following problem:

$$\hat{\boldsymbol{\beta}}_C \in \underset{\boldsymbol{\beta} \in \mathbb{R}^{(K+1)p}}{\operatorname{argmin}} \left\{ \frac{1}{2N} \sum_{k=1}^{K} \|\sqrt{n_S}(\tilde{\boldsymbol{\beta}}^{(k)} - \boldsymbol{\beta}^{(k)})\|_2^2 + \frac{1}{2N} \|\mathbf{y}^{(0)} - \mathbf{X}^{(0)} \boldsymbol{\beta}^{(0)}\|_2^2 + \lambda_0 \mathcal{R}(\boldsymbol{\beta}) \right\}, \tag{11}$$

where $\mathcal{R}(\boldsymbol{\beta}) := \|\boldsymbol{\beta}^{(0)}\|_1 + \sum_{k=1}^{K} a_k \|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(0)}\|_1$. With the solution $\hat{\boldsymbol{\beta}}_C$, the target node computes $\hat{\mathbf{w}}_C = \frac{n_S}{N} \sum_{k=1}^{K} \hat{\boldsymbol{\beta}}_C^{(k)} + \frac{n_T}{N} \hat{\boldsymbol{\beta}}_C^{(0)}$.

Step 2. The target node corrects $\hat{\mathbf{w}}_C$ on its local sample $(\mathbf{X}^{(0)}, \mathbf{y}^{(0)})$ by solving

$$\hat{\boldsymbol{\delta}}_C \in \underset{\boldsymbol{\delta} \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2n_T} \left\| \mathbf{y}^{(0)} - \mathbf{X}^{(0)} \hat{\mathbf{w}}_C - \mathbf{X}^{(0)} \boldsymbol{\delta} \right\|_2^2 + \tilde{\lambda} \|\boldsymbol{\delta}\|_1 \right\},$$

and outputs the estimator $\hat{\boldsymbol{\beta}}^{(0)}_{\text{D-TransFusion}} = \hat{\mathbf{w}}_C + \hat{\boldsymbol{\delta}}_C$.

Comparing with (2) and (5), we can see that D-TransFusion differs from the centralized TransFusion method only in the first step. To avoid the involvement of source samples, D-TransFusion replaces the least square loss $\|\mathbf{y}^{(k)} - \mathbf{X}^{(k)} \boldsymbol{\beta}^{(k)}\|_2^2$ by the squared loss $\|\sqrt{n_S}(\tilde{\boldsymbol{\beta}}^{(k)} - \boldsymbol{\beta}^{(k)})\|_2^2$, wherein $\tilde{\boldsymbol{\beta}}^{(k)}$ serves as a "pseudo sample" summarizing the information of $\boldsymbol{\beta}^{(k)}$ that the k-th source sample contains. By doing so, only one-shot communication is required to transmit the summary statistics $\tilde{\boldsymbol{\beta}}^{(k)}$ from the source to the target node, significantly reducing the communication overhead. Here, $\tilde{\boldsymbol{\beta}}^{(k)}$ is carefully selected as a de-biased

LASSO estimator (Javanmard and Montanari, 2014) that minimizes estimator variance while also controlling the bias under a given threshold. See Appendix G for a detailed discussion. This choice ensures that D-TransFusion can significantly reduce communication overhead while achieving the minimum loss of sample efficiency compared to centralized TransFusion.

More importantly, D-TransFusion allows for pre-training on each source data nodes before transfer learning. The decoupling of training on the source and target samples eliminates the need for training from scratch when the target samples change, and thus enhances the model's adaptability to downstream tasks.

We now establish the statistical precision of the one-step D-TransFusion method. Define $\delta_k = \frac{s\log p}{N} + \frac{n_S}{N}\sqrt{\frac{\log p}{n_S}} h_k$ and $\delta_0 = \frac{Ks\log p}{N} + \sqrt{\frac{\log p}{n_S}}\bar{h}$. The following theorem provides an upper bound for the estimation error of the one-step D-TransFusion estimator $\hat{\mathbf{w}}_C$.

Theorem 3. *Under Assumption 1 and 2 and the assumptions $n_S \gg Ks^2\log p$, $n_S \gtrsim (\bar{h}^2 \vee K^2)s\log p$, $h_k \asymp \bar{h}$, if we construct $\{\tilde{\boldsymbol{\beta}}^{(k)}\}_{k=1}^K$ through (57) and (58) with parameters $\tilde{\lambda}_k = \mu_k = c_1\sqrt{\log p/n_S}$, and solve problem (11) with parameters λ_0 and $\{a_k\}_{k=1,\dots,K}$ chosen such that*

$$\lambda_0 = c_0\left(\sqrt{\frac{\log p}{N}} + \delta_0\right),$$

$$a_k\lambda_0 = c_0\Big(8 \vee \frac{\bar{h}}{h_k}\Big)\left(\sqrt{\frac{n_S}{N}\frac{\log p}{N}} + \delta_k\right), \tag{12}$$

for some universal constant c_0 and c_1, then with probability at least $1 - c_2\exp(-c_3 n_T) - c_4\exp(-c_5\log p)$,

$$\|\hat{\mathbf{w}}_C - \boldsymbol{\beta}^{(0)}\|_2^2 \lesssim s\frac{\log p}{N} + \sqrt{\frac{\log p}{n_S}}\bar{h} + \varepsilon_D^2 + s\delta_0^2 + \sum_{k=1}^K \delta_k h_k, \tag{13}$$

if we further assume $\bar{h} = O(1)$, then we have

$$\|\hat{\mathbf{w}}_C - \boldsymbol{\beta}^{(0)}\|_2^2 \lesssim s\frac{\log p}{N} + \sqrt{\frac{\log p}{n_S}}\bar{h} + \varepsilon_D^2. \tag{14}$$

Let us now compare the result to Theorem 1. Under the additional assumption $n_S \gg Ks^2\log p$, $n_S \gtrsim (\bar{h}^2 \vee K^2)s\log p$, $h_k \asymp \bar{h}$, which requires the source tasks roughly equally informative with sufficiently large size, the estimation error of D-TransFusion is larger than TransFusion by $s\delta_0^2 + \sum_{k=1}^K \delta_k h_k$. This reflects the cost of sample efficiency for achieving one-shot communication. When $\bar{h} = O(1)$, such difference is negligible compared to the estimation error of TransFusion, the statistical accuracy of $\hat{\mathbf{w}}_C$ matches that of the centralized counterpart $\hat{\mathbf{w}}$ in the asymptotic sense.

The second step of D-TransFusion can be designed in analogue to that of TransFusion. Recall the event $A = \{s\log p/n_S \geq \bar{h}/\sqrt{\log p/n_T}\}$ defined in (6). The following theorem establishes the statistical estimation rate of the final estimator obtained from the two-step D-TransFusion method.

Theorem 4. *Under the assumptions of Theorem 3, if further assume $n_T \gtrsim s\log p$, $\bar{h}\sqrt{\log p/n_T} = o(1)$, then by choosing λ_0 and $\{a_k\}_{k=1,\dots,K}$ such that*

$$\lambda_0 = c_0\left(\sqrt{\frac{\log p}{N}}\mathbb{1}_A + \sqrt{\frac{\log p}{n_S}}\mathbb{1}_{A^c} + \delta_0\right),$$

$$a_k\lambda_0 = c_0\Big(8 \vee \frac{\bar{h}}{h_k}\Big)\left(\sqrt{\frac{n_S}{N}\frac{\log p}{N}} + \delta_k\right),$$

and $\tilde{\lambda} = c_1\sqrt{\log p/n_T}$ for some universal constants c_0 and c_1, we have

$$\|\hat{\boldsymbol{\beta}}^{(0)}_{\text{D-TransFusion}} - \boldsymbol{\beta}^{(0)}\|_2^2 \lesssim \frac{s\log p}{N} + \sqrt{\frac{\log p}{n_T}}\bar{h}, \tag{15}$$

with probability at least $1 - c_2\exp(-c_3\log p)$.

Theorem 4 ensures that the two-step D-TransFusion method achieves a statistical rate of the same order as the centralized two-step TransFusion method under the previously discussed additional conditions. By employing similar reasoning as in Corollary 1 and Proposition 1, we can further establish conditions under which D-TransFusion is minimax optimal. These results demonstrate that D-TransFusion is an efficient and robust solution when dealing with large-scale distributed datasets with covariate shifts.

Remark 5 (The efficacy of D-TransFusion)**.** *D-TransFusion aims to address the scenario where the source datasets are not co-located and cannot to be merged. This differs from the traditional distributed computing paradigm, where one splits the whole data into parts and parallelizes the cost due to the large data size. As for the implementation cost, since K debiased lasso estimators are computed in step 1 and transmitted in step 2, it requires per source node storing and transmitting a p-dimensional vector. The computation cost readily follows that of the debiased and standard lasso, provided in Lee et al. (2017). Although concerns may arise about the computation cost of the debiased Lasso, it is noteworthy that under mild conditions, the de-biased lasso estimator $\tilde{\boldsymbol{\beta}}^{(k)}$ can be replaced by other asymptotically unbiased estimator such as the SCAD estimator (Fan and Li (2001)), which enables the D-TransFusion to enjoy both comparable computational complexity, but distributed to K parallel processors, and statistical precision to its non-distributed counterpart given a moderate task number K.*

4 Simulation

We evaluate the empirical performance of our proposed methods, *TransFusion* and *D-TransFusion*, and compare with existing methods including *Trans-Lasso* (Li et al., 2022) and *TransHDGLM* (Li et al., 2023). For two-step approaches, we report and compare the performance of both steps to better understand scenarios where the second step is necessary. As a baseline, we include the estimation error obtained by the LASSO regression on the target task, which we call *Lasso (baseline)*. Each simulation setting is replicated with 100 independent trials, and we report the average performance. All methods are implemented based on R package *glmnet* with standard configuration, and parameters are chosen via 10-fold cross-validation.

We follow a similar experimental setup as in Li et al. (2022) and Li et al. (2023) by considering a high-dimensional linear regression problem with $p = 500$ and sparsity level $s = 10$. The target model is set as $\boldsymbol{\beta}_j^{(0)} = 0.3$ for $1 \leq j \leq s$ and $\boldsymbol{\beta}_j^{(0)} = 0$ otherwise. We generate $n_T = 150$ independent target samples $(\mathbf{X}^{(0)}, \mathbf{y}^{(0)})$ by $\mathbf{y}^{(0)} = \mathbf{X}^{(0)}\boldsymbol{\beta}^{(0)} + \boldsymbol{\epsilon}^{(0)}$ with $\mathbf{X}_{i\cdot}^{(0)} \sim N(0, \boldsymbol{I})$ and $\boldsymbol{\epsilon}_i^{(0)} \sim N(0, 1)$.

The source sample size is set to be $n_S = 200$, and the source task number K varies in the range $\{1, 3, 5, 7, 9\}$. We set $h_k = 12$ for $1 \leq k \leq K$. To simulate model and covariate shift we consider the following parameter configurations for the source tasks.

Model Shift. To investigate the impact of task diversity, we simulate two types of model shifts.
(i) *Diverse source tasks.* For $k = 1, \ldots, K-1$ we set $\boldsymbol{\beta}^{(k)} = \boldsymbol{\beta}^{(0)} + \boldsymbol{\delta}^{(k)}$ with $\boldsymbol{\delta}_j^{(k)} \sim N(0, (h_k/50)^2)$ for $1 \leq j \leq 50$ and $\boldsymbol{\delta}_j^{(k)} = 0$ otherwise. The last source model is generated with $\boldsymbol{\delta}^{(K)} = -\sum_{k=1}^{K-1} \boldsymbol{\delta}^{(k)}$ so that the task diversity measure $\varepsilon_D = 0$.

(ii) *Non-diverse source tasks.* Each k-th task-specific signal is generated as $\boldsymbol{\delta}_j^{(k)} \sim N(0.1, (h_k/50)^2)$ for $1 \leq j \leq 50$ and $\boldsymbol{\delta}_j^{(k)} = 0$ otherwise.

Covariate Shift. To demonstrate the robustness of TransFusion to covariate shifts, we consider two settings with different covariate distributions. In each setting, we generate n_S independent samples for each source task.
(a) *Homogeneous design.* Each $\mathbf{X}_{i\cdot}^{(k)} \sim N(0, \boldsymbol{I})$.

(b) *Heterogeneous design.* Each $\mathbf{X}_{i\cdot}^{(k)} \sim N(0, \boldsymbol{\Sigma}^{(k)})$ with $\boldsymbol{\Sigma}^{(k)} = (\boldsymbol{A}^{(k)})^\top(\boldsymbol{A}^{(k)}) + \boldsymbol{I}$. Here $\boldsymbol{A}^{(k)}$ is a random matrix with each entry equals 0.3 with probability 0.3 and equals 0 with probability 0.7.

We consider four experimental settings based on combinations of model design (i) and (ii) with covariate design (a) and (b) to generate the source samples. Fig. 1 reports the ℓ_2 estimation error of $\boldsymbol{\beta}^{(0)}$ versus the source task number K in these four settings. Fig. 2 reports a focused comparison between the performance of D-TransFusion and TransFusion in heterogeneous design (b). More simulation results are reported in the Appendix E. The following comments are in order.

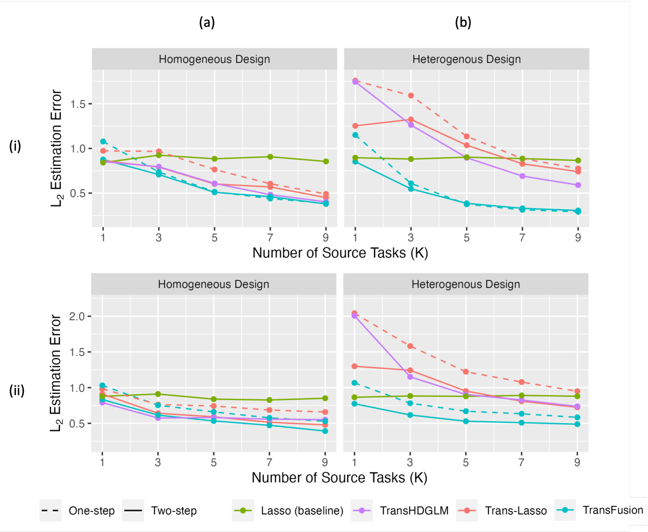

Figure 1: Comparison of estimation errors under (i) diverse and (ii) non-diverse source task settings with (a) homogeneous design and (b) heterogeneous design.

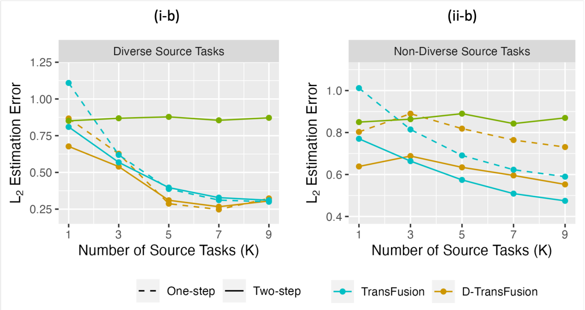

Figure 2: Comparison of the estimation errors of D-TransFusion and TransFusion methods under (i) diverse and (ii) non-diverse source task settings with (b) heterogeneous design.

- Robustness to covariate shift. Fig.1 (i-a) shows in the diverse source tasks setting (i) when there is no covariate shift (a setting in favor of *Trans-Lasso*), TransFusion achieves comparable estimation error with the state-of-the-art methods. However, when covariate shift exists, a comparison of Fig.1 (i-b) with (i-a) shows the estimator errors obtained by both *Trans-Lasso* and *TransHDGLM* increase significantly, and are even larger than *Lasso (baseline)* for small values of K. On the contrary, the performance of TransFusion remains the same, showing its robustness to covariate shift. When the source tasks are non-diverse, Fig.1 (ii-a) and (ii-b) reveal similar advantages achieved by two-step TransFusion. Consequently, TransFusion offers better reliability when the degree of covariate shift and source task diversity are unknown a priori.

• Impact of source task diversity. In Theorem 1 we have proved that when the source tasks are sufficiently diverse, applying one-step TransFusion suffices to obtain a small estimation error. Fig.1 (i-a) and (i-b) corroborate this statement: under both homogeneous and heterogeneous covariate designs, one- and two-step TransFusion yields comparable estimation errors. In the more challenging setting with non-diverse source tasks, Fig.1 (ii-a) and (ii-b) indicate applying the second de-bias step reduces the estimation error, which is also consistent with the implications of Theorem 2.

• D-TransFusion matches the performance of TransFusion. Fig. 2 demonstrates for small K, D-TransFusion attains comparable or even smaller estimation error than TransFusion, but uses only one-shot communication and has the ability to quickly adapt to downstream tasks. As K increases, the gap reduces, and TransFusion gradually outperforms D-TransFusion. This is because D-TransFusion has a more restrictive growth condition on the source sample size n_S with K, a requirement common to divide-and-conquer type methods. It also reveals the tradeoff between sample efficiency and communication cost.

In addition to the simulation study using synthetic data, the covariate-shift robustness of *TransFusion* method is further validated on real-world application through the MNIST handwritten digit classification task. The results are provided in Appendix F.

5 Conclusion

In this paper, we introduce a novel solution to tackle model shifts in high-dimensional transfer learning problems, ensuring robustness to covariate shifts and efficiency in knowledge transfer across distributed datasets. We provide a theoretical guarantee, showing its capacity to fully utilize the source samples to achieve an optimal estimation rate with one-shot communication and under covariate shifts. Simulation results validate our theory and showcase the state-of-the-art performance of the proposed method under various settings.

References

Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. *Advances in Neural Information Processing Systems*, 23, 2010.

Andreas Argyriou, Massimiliano Pontil, Yiming Ying, and Charles Micchelli. A spectral regularization framework for multi-task structure learning. *Advances in Neural Information Processing Systems*, 20, 2007.

Hamsa Bastani. Predicting with proxies: Transfer learning in high dimension. *Management Science*, 67(5):2964–2984, 2021.

Xiangli Chen, Mathew Monfort, Anqi Liu, and Brian D Ziebart. Robust covariate shift regression. In *Artificial Intelligence and Statistics*, pages 1270–1279. PMLR, 2016.

Corinna Cortes and Mehryar Mohri. Domain adaptation in regression. In *International Conference on Algorithmic Learning Theory*, pages 308–323. Springer, 2011.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

Simon S Du, Jayanth Koushik, Aarti Singh, and Barnabás Póczos. Hypothesis transfer learning via transformation functions. *Advances in Neural Information Processing Systems*, 30, 2017.

Simon Shaolei Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. In *International Conference on Learning Representations*, 2020.

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

Jianqing Fan, Runze Li, Cun-Hui Zhang, and Hui Zou. *Statistical foundations of data science*. CRC press, 2020.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Xinmeng Huang, Kan Xu, Donghwan Lee, Hamed Hassani, Hamsa Bastani, and Edgar Dobriban. Optimal heterogeneous collaborative linear regression and contextual bandits. *arXiv preprint arXiv:2306.06291*, 2023.

Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

Jason D Lee, Qiang Liu, Yuekai Sun, and Jonathan E Taylor. Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(1): 115–144, 2017.

Qi Lei, Wei Hu, and Jason Lee. Near-optimal linear regression under distribution shift. In *International Conference on Machine Learning*, pages 6164–6174. PMLR, 2021.

Furong Li and Huiyan Sang. Spatial homogeneity pursuit of regression coefficients for large datasets. *Journal of the American Statistical Association*, 2019.

Sai Li, T Tony Cai, and Hongzhe Li. 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, 2022.

Sai Li, Linjun Zhang, T Tony Cai, and Hongzhe Li. Estimation and inference for high-dimensional generalized linear models with knowledge transfer. *Journal of the American Statistical Association*, pages 1–12, 2023.

Xudong Li, Defeng Sun, and Kim-Chuan Toh. A highly efficient semismooth Newton augmented lagrangian method for solving lasso problems. *SIAM Journal on Optimization*, 28(1):433–458, 2018.

Haotian Lin and Matthew Reimherr. On transfer learning in functional linear regression. *arXiv preprint arXiv:2206.04277*, 2022.

Haotian Lin and Matthew Reimherr. Smoothness adaptive hypothesis transfer learning. *arXiv preprint arXiv:2402.14966*, 2024.

Shuo Shuo Liu. Unified transfer learning models for high-dimensional linear regression. *arXiv preprint arXiv:2307.00238*, 2023.

Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Advances in Neural Information Processing Systems*, 24, 2011.

Karim Lounici, Massimiliano Pontil, Alexandre B Tsybakov, and Sara Van De Geer. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468*, 2009.

Shangyun Lu, Bradley Nott, Aaron Olson, Alberto Todeschini, Hossein Vahabi, Yair Carmon, and Ludwig Schmidt. Harder or different? a closer look at distribution shift in dataset reproduction. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, volume 5, page 15, 2020.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Ryan Mcdonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon Mann. Efficient large-scale distributed training of conditional maximum entropy models. *Advances in Neural Information Processing Systems*, 22, 2009.

Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4): 538–557, 2012.

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

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. MIT Press, 2008.

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020.

Davod Khojasteh Salkuyeh and Fatemeh Panjeh Ali Beik. An explicit formula for the inverse of arrowhead and doubly arrow matrices. *International Journal of Applied and Computational Mathematics*, 4:1–8, 2018.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Masaaki Takada and Hironori Fujisawa. Transfer learning via l1 regularization. *Advances in Neural Information Processing Systems*, 33:14266–14277, 2020.

Lu Tang and Peter XK Song. Fused lasso approach in regression coefficients clustering: learning parameter heterogeneity in data integration. *The Journal of Machine Learning Research*, 17(1):3915–3937, 2016.

Jaclyn N Taroni, Peter C Grayson, Qiwen Hu, Sean Eddy, Matthias Kretzler, Peter A Merkel, and Casey S Greene. Multiplier: a transfer learning framework for transcriptomics reveals systemic features of rare disease. *Cell systems*, 8(5):380–394, 2019.

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

Ye Tian, Gu Yuqi, and Feng Yang. Learning from similar linear representations: Adaptivity, minimaxity, and robustness. *arXiv preprint arXiv:2303.17765*, 2023.

Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33:7852–7862, 2020.

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

Xin Zhang, Jia Liu, and Zhengyuan Zhu. Learning coefficient heterogeneity over networks: A distributed spanning-tree-based fused-lasso regression. *Journal of the American Statistical Association*, pages 1–13, 2022.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101 (476):1418–1429, 2006.

A Proof of Theorems

In this section, we provide proof of all the theorems. Throughout the section, we adopt the following notations to analyze the solution of the problem (2):

$$\mathbf{y} := \begin{pmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \\ \vdots \\ \mathbf{y}^{(K)} \\ \mathbf{y}^{(0)} \end{pmatrix} \quad \mathbf{X} := \begin{pmatrix} \mathbf{X}^{(1)} & 0 & \cdots & 0 & \mathbf{X}^{(1)} \\ 0 & \mathbf{X}^{(2)} & \cdots & 0 & \mathbf{X}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathbf{X}^{(K)} & \mathbf{X}^{(K)} \\ 0 & 0 & \cdots & 0 & \mathbf{X}^{(0)} \end{pmatrix} \quad \boldsymbol{\theta}^* = \begin{pmatrix} (\boldsymbol{\theta}^*)^{(1)} \\ (\boldsymbol{\theta}^*)^{(2)} \\ \vdots \\ (\boldsymbol{\theta}^*)^{(K)} \\ (\boldsymbol{\theta}^*)^{(0)} \end{pmatrix} := \begin{pmatrix} \boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(0)} \\ \boldsymbol{\beta}^{(2)} - \boldsymbol{\beta}^{(0)} \\ \vdots \\ \boldsymbol{\beta}^{(K)} - \boldsymbol{\beta}^{(0)} \\ \boldsymbol{\beta}^{(0)} \end{pmatrix}. \tag{16}$$

where recall that $\boldsymbol{\beta}^{(0)}$ is the target model parameter and $\boldsymbol{\beta}^{(k)}$s are source model parameters.

Under this transformation, solving problem (2) is equivalent as solving

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\{\mathcal{L}(\boldsymbol{\theta}) + \lambda_0 \mathcal{R}(\boldsymbol{\theta})\}, \tag{17}$$

where we define $\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2N}\left\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\right\|_2^2$ and $\lambda_0\mathcal{R}(\boldsymbol{\theta}) := \lambda_0\left\|\boldsymbol{\theta}^{(0)}\right\|_1 + \lambda_0\sum_{k=1}^K a_k \left\|\boldsymbol{\theta}^{(k)}\right\|_1 = \sum_{k=0}^K \lambda_0 a_k \left\|\boldsymbol{\theta}^{(k)}\right\|_1$ for any $\boldsymbol{\theta} \in \mathbb{R}^{(K+1)p}$. Since there exists a one-to-one transformation between $\boldsymbol{\theta}^*$ and $\boldsymbol{\beta}$, we can quantify the estimation error $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$ by analyzing $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$.

We first establish an essential property of sub-Gaussian design matrices, which serves as fundamental building blocks for our subsequent analysis. The following lemma 1 shows that the least square objective function has a *restricted* strongly convex (RSC) and *restricted* smooth (RSM) property. For a detailed discussion and proof of this lemma, interested readers can refer to Lemma 13 of Loh and Wainwright (2011) and Lemma 6 of Agarwal et al. (2010).

Lemma 1 (RSC and RSM property)**.** *Under Assumption 1, for any $\mathbf{\Delta} \in \mathbb{R}^p$, with probability at least $1 - c_1\exp(-c_2 n_k)$,*

$$\frac{1}{n_k}\left\|\mathbf{X}^{(k)}\mathbf{\Delta}\right\|_2^2 = \mathbf{\Delta}^\top\hat{\mathbf{\Sigma}}^{(k)}\mathbf{\Delta} \geq \alpha_k\|\mathbf{\Delta}\|_2^2 - \beta_k\frac{\log p}{n_k}\|\mathbf{\Delta}\|_1^2,$$
$$\frac{1}{n_k}\left\|\mathbf{X}^{(k)}\mathbf{\Delta}\right\|_2^2 = \mathbf{\Delta}^\top\hat{\mathbf{\Sigma}}^{(k)}\mathbf{\Delta} \leq \gamma_k\|\mathbf{\Delta}\|_2^2 + \tau_k\frac{\log p}{n_k}\|\mathbf{\Delta}\|_1^2,$$

where $\alpha_k = \frac{1}{2}\Lambda_{\min}(\mathbf{\Sigma}^{(k)}) \geq 1/c$,$\gamma_k = 2\Lambda_{\max}(\mathbf{\Sigma}^{(k)}) \leq c$ and $\beta_k, \tau_k \leq c$, $n_S = n_S$ for $k = 1, \ldots, K$ and n_T for $k = 0$.

A.1 Proof of Theorem 1

Define $\hat{\mathbf{\Delta}} := \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$ as the estimation error of $\hat{\boldsymbol{\theta}}$ and the corresponding k-th block $\hat{\mathbf{\Delta}}^{(k)} := \hat{\boldsymbol{\theta}}^{(k)} - (\boldsymbol{\theta}^*)^{(k)}$. For brevity, we will omit the superscript and write $(\boldsymbol{\theta}^*)^{(k)}$ as $\boldsymbol{\theta}^{(k)}$ for $0 \leq k \leq K$ when there is no ambiguity.

Further define $\hat{\mathbf{\Delta}}^{\boldsymbol{w}} := \sum_{k=1}^K \frac{n_S}{N}\hat{\mathbf{\Delta}}^{(k)} + \hat{\mathbf{\Delta}}^{(0)} = \hat{\mathbf{w}} - \mathbf{w}$ as the estimation error of the parameter average $\mathbf{w}$. Our goal is to establish an upper bound for $\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\|_2^2$.

The proof of Theorem 1 relies on three key technical lemmas. The proof of these lemmas is in Appendix B. The first lemma establishes an upper bound for the first-order term of the Taylor series expansion of $\mathcal{L}(\boldsymbol{\theta})$.

Lemma 2. *Under Assumption 1 and 2, if $n_S \gtrsim \log p$, then by choosing $\lambda_0 = c_0\sqrt{\frac{\log p}{N}}$ and $\lambda_k = a_k\lambda_0 = c_0\sqrt{\frac{n_S}{N}\frac{\log p}{N}}$ for some appropriate constant c_0, we have for any $\mathbf{\Delta} = \left(\left(\mathbf{\Delta}^{(1)}\right)^\top, \ldots, \left(\mathbf{\Delta}^{(K)}\right)^\top, \left(\mathbf{\Delta}^{(0)}\right)^\top\right)^\top \in \mathbb{R}^{(K+1)p}$,*

$$\left|\left\langle\nabla\mathcal{L}\left(\boldsymbol{\theta}^*\right), \mathbf{\Delta}\right\rangle\right| \leq \sum_{k=1}^K \frac{\lambda_k}{2}\left\|\mathbf{\Delta}^{(k)}\right\|_1 + \frac{\lambda_0}{2}\left\|\mathbf{\Delta}^{(0)}\right\|_1.$$

with probability larger than $1 - c_1 \exp\left(-c_2 \log p\right)$.

Recall that we define $\lambda_k = a_k \lambda_0$. The next lemma establishes a restricted set of directions in which $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} = \hat{\mathbf{w}} - \mathbf{w}$ lies.

Lemma 3. *Under Assumption 1 and 2, and the conditions of Lemma 2, if further assume* $\lambda_k \geq 8\lambda_0 \frac{n_S}{N}$ *and* $n_S > n_T$*, then the estimation error* $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}$ *satisfies the inequality*

$$\sum_{k=1}^{K} \lambda_k \left\|\hat{\boldsymbol{\Delta}}^{(k)}\right\|_1 + 2\lambda_0 \left\|\hat{\boldsymbol{\Delta}}^{(0)}\right\|_1 \leq 8\lambda_0 \left\|\hat{\boldsymbol{\Delta}}_S^{\boldsymbol{w}}\right\|_1 + 8\sum_{k=1}^{K} \lambda_k h_k,$$

with probability larger than $1 - c_1 \exp(-c_2 \log p)$*, where* S *is the support set of* $\boldsymbol{\beta}^{(0)}$.

The following lemma ensures a property analogous to restricted strong convexity for $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}$.

Lemma 4. *Under Assumption 1 and 2 and the conditions of Lemma 3, the estimation error* $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}$ *satisfies*

$$\hat{\boldsymbol{\Delta}}^{\top} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Delta}} = \mathcal{L}\left(\boldsymbol{\theta}^* + \hat{\boldsymbol{\Delta}}\right) - \mathcal{L}\left(\boldsymbol{\theta}^*\right) - \left\langle \nabla \mathcal{L}\left(\boldsymbol{\theta}^*\right), \hat{\boldsymbol{\Delta}} \right\rangle \geq (1 - u_n)\alpha_{\min} \left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_2^2 - v_n \sum_{k=1}^{K} \lambda_k h_k \tag{18}$$

with probability larger than $1 - c_1 \exp(c_2 n_T) - c_3 \exp(c_4 \log p)$*, where* $\hat{\boldsymbol{\Sigma}} := \frac{1}{N}\mathbf{X}^{\top}\mathbf{X}$*,*

$$u_n := \frac{256 \beta_{\max} \lambda_0^2}{\alpha_{\min} \lambda_k^2 \wedge (\lambda_0^2/(K+1))} \frac{s \log p}{N}, \quad v_n := \frac{256 \beta_{\max}}{\lambda_k^2 \wedge (\lambda_0^2/(K+1))} \frac{\log p}{N} \left(\sum_{k=1}^{K} \lambda_k h_k\right),$$
$$\alpha_{\min} := \min_{0 \leq k \leq K} \alpha_k, \quad \beta_{\max} := \max_{0 \leq k \leq K} \beta_k,$$

with RSC constants (α_k, β_k) *defined in Lemma 1.*

We now turn to the proof of the theorem. In the following proof, we make use of the function $F : \mathbb{R}^{(K+1)p} \to \mathbb{R}$, given by

$$F(\boldsymbol{\Delta}) = \mathcal{L}\left(\boldsymbol{\theta}^* + \boldsymbol{\Delta}\right) - \mathcal{L}\left(\boldsymbol{\theta}^*\right) + \lambda_0 \mathcal{R}\left(\boldsymbol{\theta}^* + \boldsymbol{\Delta}\right) - \lambda_0 \mathcal{R}\left(\boldsymbol{\theta}^*\right),$$

where $\boldsymbol{\theta}^*$ is the transformed model parameter defined in (16), and $\boldsymbol{\Delta} = \left(\left(\boldsymbol{\Delta}^{(1)}\right)^{\top}, \ldots, \left(\boldsymbol{\Delta}^{(K)}\right)^{\top}, \left(\boldsymbol{\Delta}^{(0)}\right)^{\top}\right)^{\top} \in \mathbb{R}^{(K+1)p}$.

By Lemma 2, triangle inequality, and the fact that $\left\|\boldsymbol{\theta}_{S^c}^{(0)}\right\|_1 = 0$, $\left\|\boldsymbol{\theta}^{(k)}\right\|_1 \leq h_k$ for $1 \leq k \leq K$, we have

$$\begin{aligned}
F(\hat{\boldsymbol{\Delta}}) =& \mathcal{L}\left(\boldsymbol{\theta}^* + \hat{\boldsymbol{\Delta}}\right) - \mathcal{L}\left(\boldsymbol{\theta}^*\right) + \lambda_0 \mathcal{R}\left(\boldsymbol{\theta}^* + \hat{\boldsymbol{\Delta}}\right) - \lambda_0 \mathcal{R}\left(\boldsymbol{\theta}^*\right) \\
\geq & - \left\|\mathcal{L}\left(\boldsymbol{\theta}^*\right)\right\|_{\infty} \|\hat{\boldsymbol{\Delta}}\|_1 + \hat{\boldsymbol{\Delta}}^{\top} \nabla^2 \mathcal{L}\left(\boldsymbol{\theta}^* + \gamma \hat{\boldsymbol{\Delta}}\right) \hat{\boldsymbol{\Delta}} \quad (\gamma \in (0,1)) \\
& + \sum_{k=1}^{K} \lambda_k \left(\left\|\boldsymbol{\theta}^{(k)} + \hat{\boldsymbol{\Delta}}^{(k)}\right\|_1 - \left\|\boldsymbol{\theta}^{(k)}\right\|_1\right) + \lambda_0 \left\|\boldsymbol{\theta}^{(0)} + \hat{\boldsymbol{\Delta}}^{(0)}\right\|_1 - \lambda_0 \left\|\boldsymbol{\theta}^{(0)}\right\|_1 \\
\geq & - \sum_{k=1}^{K} \frac{\lambda_k}{2} \left\|\hat{\boldsymbol{\Delta}}^{(k)}\right\|_1 - \frac{\lambda_0}{2} \left\|\hat{\boldsymbol{\Delta}}^{(0)}\right\|_1 + \hat{\boldsymbol{\Delta}}^{\top} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Delta}} \\
& + \sum_{k=1}^{K} \lambda_k \left(\left\|\hat{\boldsymbol{\Delta}}^{(k)}\right\|_1 - 2\left\|\boldsymbol{\theta}^{(k)}\right\|_1\right) + \lambda_0 \left(\left\|\boldsymbol{\theta}_S^{(0)}\right\|_1 - \left\|\hat{\boldsymbol{\Delta}}_S^{(0)}\right\|_2 + \left\|\hat{\boldsymbol{\Delta}}_{S^c}^{(0)}\right\|_1 - \left\|\boldsymbol{\theta}_{S^c}^{(0)}\right\|_1 - \left\|\boldsymbol{\theta}_S^{(0)}\right\|_1 - \left\|\boldsymbol{\theta}_{S^c}^{(0)}\right\|_1\right) \\
\geq & \hat{\boldsymbol{\Delta}}^{\top} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Delta}} + \frac{\lambda_0}{2} \left(\left\|\hat{\boldsymbol{\Delta}}_{S^c}^{(0)}\right\|_1 - 3\left\|\hat{\boldsymbol{\Delta}}_S^{(0)}\right\|_1\right) + \sum_{k=1}^{K} \frac{\lambda_k}{2} \left\|\hat{\boldsymbol{\Delta}}^{(k)}\right\|_1 - 2\sum_{k=1}^{K} \lambda_k h_k.
\end{aligned}$$

with probability larger than $1 - c_1 \exp\left(-c_2 \log p\right)$.

By the definition of $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}$, we have $\hat{\boldsymbol{\Delta}}^{(0)} = \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} - \sum_{k=1}^{K} \frac{n_S}{N} \hat{\boldsymbol{\Delta}}^{(k)}$. Therefore, applying triangle inequality yields

$$
\begin{aligned}
F(\hat{\boldsymbol{\Delta}}) \geq \hat{\boldsymbol{\Delta}}^{\top} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Delta}} + \frac{1}{2}\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}_{S^c} \right\|_1 - \frac{1}{2}\lambda_0 \sum_{k=1}^{K} \frac{n_S}{N} \left\| \hat{\boldsymbol{\Delta}}^{(k)}_{S^c} \right\|_1 - \frac{3}{2}\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}_{S} \right\|_1 - \frac{3}{2}\lambda_0 \sum_{k=1}^{K} \frac{n_S}{N} \left\| \hat{\boldsymbol{\Delta}}^{(k)}_{S} \right\|_1 \\
+ \sum_{k=1}^{K} \frac{\lambda_k}{2} \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 - 2 \sum_{k=1}^{K} \lambda_k h_k. \qquad (19)
\end{aligned}
$$

Recall that we select $\lambda_0, \ldots, \lambda_k$ such that $\frac{\lambda_k}{2} \geq \frac{3}{2} \frac{n_S}{N} \lambda_0$, so we have

$$
\sum_{k=1}^{K} \frac{\lambda_k}{2} \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 - \frac{3}{2}\lambda_0 \sum_{k=1}^{K} \frac{n_S}{N} \left\| \hat{\boldsymbol{\Delta}}^{(k)}_{S} \right\|_1 - \frac{1}{2}\lambda_0 \sum_{k=1}^{K} \frac{n_S}{N} \left\| \hat{\boldsymbol{\Delta}}^{(k)}_{S^c} \right\|_1 \geq 0. \qquad (20)
$$

Notice that $\hat{\boldsymbol{\theta}}$ is the solution to the problem (17). We then have $\hat{\boldsymbol{\Delta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* = \underset{\boldsymbol{\Delta}}{\mathrm{argmin}} F(\boldsymbol{\Delta})$. Since $F(\mathbf{0}) = 0$, it follows that $F(\hat{\boldsymbol{\Delta}}) \leq 0$. This summing with (19) and (20) leads to

$$
\begin{aligned}
0 &\geq \hat{\boldsymbol{\Delta}}^{\top} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Delta}} - \frac{3}{2}\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}_{S} \right\|_1 + \frac{1}{2}\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}_{S^c} \right\|_1 - 2 \sum_{k=1}^{K} \lambda_k h_k \qquad (21) \\
&= \hat{\boldsymbol{\Delta}}^{\top} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Delta}} - \frac{3}{2}\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \right\|_1 + 2\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}_{S^c} \right\|_1 - 2 \sum_{k=1}^{K} \lambda_k h_k. \qquad (22)
\end{aligned}
$$

We now establish the upper bound for error measured in ℓ_2 norm, $\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_2$. An application of Lemma 4 on (21) yields that with probability larger than $1 - c_1 \exp(c_2 n_T) - c_3 \exp(c_4 \log p)$,

$$
0 \geq (1 - u_n)\alpha_{\min} \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \right\|_2^2 - \frac{3}{2}\lambda_0 \sqrt{s} \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \right\|_2 - (2 + v_n) \sum_{k=1}^{K} \lambda_k h_k
$$

where we use the fact that $\left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}_{S^c} \right\|_1 \geq 0$. If $u_n = o(1)$, we can show that for a sufficiently large n_S,

$$
\begin{aligned}
\left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \right\|_2 &\leq \frac{\frac{3}{2}\lambda_0\sqrt{s} + \sqrt{\frac{9}{4}\lambda_0^2 s + 4\left(1 - u_n\right)\left(2 + v_n\right)\alpha_{\min} \sum_{k=1}^{K} \lambda_k h_k}}{2\left(1 - u_n\right)\alpha_{\min}} \\
&\lesssim \sqrt{\frac{s \log p}{N}} + \sqrt{(1 + v_n) \sum_{k=1}^{K} \frac{n_S}{N} \sqrt{\frac{\log p}{n_S}} h_k} \qquad (23)
\end{aligned}
$$

by plugging in the choice of λ_0 and λ_ks, which is the desired result.

It remains to show the order of v_n and prove that $u_n = o(1)$ under the conditions of Theorem 1. By the assumptions in Theorem 1 and the choice of $\lambda_0, \ldots, \lambda_K$, we have

$$
u_n = \frac{256\beta_{\max}\lambda_0^2}{\alpha_{\min}\lambda_k^2 \wedge (\lambda_0^2/(K+1))} \frac{s \log p}{N} \lesssim \frac{s \log p}{n_S} = o(1),
$$

and

$$
v_n = \frac{256\beta_{\max}}{\lambda_k^2 \wedge (\lambda_0^2/(K+1))} \frac{\log p}{N} \left(\sum_{k=1}^{K} \lambda_k h_k \right) \lesssim \sqrt{\frac{K^2 \log p}{n_S}} \bar{h}.
$$

The proof is then finished.

A.2 Proof of Theorem 2

Similar to the arguments in the proof of Theorem 1, we first define $\tilde{\mathcal{L}}(\boldsymbol{\delta}) = \frac{1}{2n_T}\left\|\mathbf{y}^{(0)} - \mathbf{X}^{(0)}\hat{\mathbf{w}} - \mathbf{X}^{(0)}\boldsymbol{\delta}\right\|_2^2$ and $\tilde{F}(\boldsymbol{\Delta}) = \tilde{\mathcal{L}}\left(\boldsymbol{\delta}^* + \boldsymbol{\Delta}\right) - \tilde{\mathcal{L}}\left(\boldsymbol{\delta}^*\right) + \tilde{\lambda}\left\|\boldsymbol{\delta}^* + \boldsymbol{\Delta}\right\|_1 - \tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1$, where $\boldsymbol{\delta}^* = \boldsymbol{\beta}^{(0)} - \mathbf{w}$ is the contrast between the target parameter and the averaged parameter. Denoting $\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} = \hat{\boldsymbol{\delta}} - \boldsymbol{\delta}^*$. Recall that $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} = \hat{\mathbf{w}} - \mathbf{w}$, by Hölder inequality and triangle inequality,

$$\begin{aligned}
\left\langle \nabla\tilde{\mathcal{L}}\left(\boldsymbol{\delta}^*\right), \hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right\rangle &= \frac{1}{n_T}\left\langle \left(\mathbf{X}^{(0)}\right)^\top\left[\mathbf{y}^{(0)} - \mathbf{X}^{(0)}\boldsymbol{\beta}^{(0)} - \mathbf{X}^{(0)}\left(\hat{\mathbf{w}} + \boldsymbol{\delta}^* - \boldsymbol{\beta}^{(0)}\right)\right], \hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right\rangle \\
&= \frac{1}{n_T}\left\langle \left(\mathbf{X}^{(0)}\right)^\top\left[\boldsymbol{\epsilon}^{(0)} - \mathbf{X}^{(0)}(\hat{\mathbf{w}} - \mathbf{w})\right], \hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right\rangle \\
&\le \frac{1}{n_T}\left\|\left(\mathbf{X}^{(0)}\right)^\top\boldsymbol{\epsilon}^{(0)}\right\|_\infty \|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 + \frac{1}{2}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right)^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} + \frac{1}{2}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}.
\end{aligned}$$

By Lemma 2, if $n_T \gtrsim \log p$, we can choose $\tilde{\lambda} = c\sqrt{\frac{\log p}{n_T}}$ for some constant c so that $\frac{1}{n_T}\left\|\left(\mathbf{X}^{(0)}\right)^\top\boldsymbol{\epsilon}^{(0)}\right\|_\infty \le \frac{\tilde{\lambda}}{2}$ with probability larger than $1 - c_1\exp(c_2\log p)$. Therefore, it holds that

$$\begin{aligned}
\tilde{\mathcal{L}}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} + \boldsymbol{\delta}^*\right) - \tilde{\mathcal{L}}(\widehat{\boldsymbol{\Delta}}^{\delta}) &= \left\langle \nabla\tilde{\mathcal{L}}\left(\boldsymbol{\delta}^*\right), \hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right\rangle + \left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right)^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} \\
&\ge -\frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 + \frac{1}{2}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right)^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} - \frac{1}{2}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}
\end{aligned}$$

So by the optimality condition,

$$\begin{aligned}
0 &\ge \tilde{F}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}) \\
&\ge \tilde{\mathcal{L}}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} + \boldsymbol{\delta}^*\right) - \tilde{\mathcal{L}}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}) + \tilde{\lambda}\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} + \boldsymbol{\delta}^*\right\|_1 - \tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1 \\
&\ge \tilde{\mathcal{L}}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} + \boldsymbol{\delta}^*\right) - \tilde{\mathcal{L}}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}) + \tilde{\lambda}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 - 2\tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1 \\
&\ge \frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 + \frac{1}{2}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right)^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} - \frac{1}{2}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} - 2\tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1 \qquad (24)
\end{aligned}$$

Since the result involves $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}$, we first establish the following auxiliary lemma:

Lemma 5. *Under Assumption 1 and 2, $n_S \gtrsim \log p$, $n_S > n_T$, if we choose $\lambda_0 \gtrsim \sqrt{\frac{\log p}{N}}$, $\lambda_k = a_k\lambda_0 \gtrsim \sqrt{\frac{n_S}{N}}\sqrt{\frac{\log p}{N}}$ such that*

$$\lambda_k \ge 8\lambda_0\frac{n_S}{N},\ u_n = \frac{256\beta_{\max}\lambda_0^2}{\alpha_{\min}\lambda_k^2 \wedge (\lambda_0^2/(K+1))}\frac{s\log p}{N} = o(1) \text{ and } v_n = \frac{256\beta_{\max}}{\lambda_k^2 \wedge (\lambda_0^2/(K+1))}\frac{\log p}{N}\left(\sum_{k=1}^K \lambda_k h_k\right) = O(1),$$

then we have

$$\begin{aligned}
\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_2 &\lesssim \sqrt{s}\lambda_0 + \sqrt{\sum_{k=1}^K \lambda_k h_k} \\
\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_1 &\lesssim s\lambda_0 + \sqrt{s}\sqrt{\sum_{k=1}^K \lambda_k h_k} + \frac{\sum_{k=1}^K \lambda_k h_k}{\lambda_0}
\end{aligned}$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$.

Notice that according to the theorem statement, the choice of λ_0 and λ_ks depends on the event A. It can be verified that either selection fulfills the conditions outlined in Lemma 5. We now discuss by cases and apply Lemma 5 to prove the result.

Case 1: We start with the case when the event A holds. That is, we have

$$s\log p/n_S \geq \bar{h}\sqrt{\log p/n_T} \tag{25}$$

Under this condition, we choose

$$\lambda_0 = c_1\sqrt{\frac{\log p}{N}}, \text{ and } a_k = 8\sqrt{\frac{n_S}{N}}.$$

Applying Lemma 5 yields

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_2 \lesssim \sqrt{\frac{s\log p}{N}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} \tag{26}$$

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_1 \lesssim s\sqrt{\frac{\log p}{N}} + \sqrt{\sqrt{\frac{\log p}{n_S}}s\bar{h}} + \sqrt{\frac{N}{n_S}}\bar{h} \tag{27}$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$.

(i). If $\frac{1}{2}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \geq 2\tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1$, according to (24), we have

$$0 \geq \frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 + \frac{1}{2}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right)^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} - (\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}. \tag{28}$$

By Lemma 1 and the condition $n_T \gtrsim \log p$, we have $(\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \leq \gamma_0\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_2^2 + \tau_0\frac{\log p}{n_T}\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_1^2$ for some constants γ_0 and τ_0, this together with (28) indicates

$$\begin{aligned}\frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 &\leq (\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \leq \gamma_0\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_2^2 + \tau_0\frac{\log p}{n_T}\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_1^2 \\ &\lesssim \frac{s\log p}{N} + (1 + \frac{s\log p}{n_T})\sqrt{\frac{\log p}{n_S}}\bar{h} + \frac{(s\log p)^2}{n_T N} + \frac{N}{n_S}\bar{h}^2\frac{\log p}{n_T}.\end{aligned} \tag{29}$$

where the last inequality is based on the results in (26) and (27). In Theorem 2 we assume $\frac{s\log p}{n_T} = O(1)$ and $\bar{h}\sqrt{\log p/n_T} = o(1)$. According to (25), we have $(N/n_S)\bar{h}\sqrt{\log p/n_T} \leq (K+1)s\log p/n_S = O(1)$. Applying these results to (29) yields $\frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 = o_p(1)$.

On the other hand, if we apply Lemma 1 to (28), we then have

$$0 \geq \left(1 - \frac{\beta_0}{\tilde{\lambda}}\frac{\log p}{n_T}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1\right)\frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 + \frac{1}{2}\alpha_0\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2^2 - \gamma_0\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_2^2 - \tau_0\frac{\log p}{n_T}\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_1^2.$$

Notice that we choose $\tilde{\lambda} = c\sqrt{\frac{\log \mathrm{p}}{n_T}}$ for some universal constant c. Therefore, we may choose $c > \sqrt{2\beta_0}$ so we have $\beta_0\log p/n_T < \tilde{\lambda}^2/2$. This together with (29) and the fact that $\frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 = o_p(1)$ leads to

$$\frac{1}{2}\alpha_0\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2^2 \leq \gamma_0\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_2^2 + \tau_0\frac{\log p}{n_T}\left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_1^2.$$

Based on similar arguments to those in (29), we have

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2 \lesssim \sqrt{\frac{s\log p}{N}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} + \sqrt{\frac{N}{n_S}}\sqrt{\frac{\log p}{n_T}}\bar{h}.$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$.

(ii). If $\frac{1}{2}(\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}})^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \leq 2\tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1$, we have

$$0 \geq \frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 + \frac{1}{2}\left(\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right)^\top\hat{\boldsymbol{\Sigma}}^{(0)}\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}} - 4\tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1$$

which implies that $\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 \le 8\|\boldsymbol{\delta}^*\|_1 \le 8\bar{h}$.

By applying Lemma 1 again, we have

$$\begin{aligned} 0 &\ge \frac{\tilde{\lambda}}{2}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 + \frac{1}{2}\alpha_0\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2^2 - \frac{1}{2}\beta_0\frac{\log p}{n_T}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1^2 - 4\tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1 \\ &\ge \frac{1}{2}\alpha_0\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2^2 - 32\beta_0\frac{\log p}{n_T}\left\|\boldsymbol{\delta}^*\right\|_1^2 - 4\tilde{\lambda}\left\|\boldsymbol{\delta}^*\right\|_1 . \end{aligned}$$

with probability larger than $1 - c_1\exp(c_2 n_T)$.

So in this case, we have

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2 \le \sqrt{\frac{64\beta_0}{\alpha_0}\frac{\log p}{n_T}\bar{h}^2 + 8\frac{\tilde{\lambda}}{\alpha_0}\bar{h}} \lesssim \sqrt{\frac{\log p}{n_T}}\bar{h} + \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}}$$

and

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2 \le \|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_1 \le 8\bar{h}$$

Under the assumption that $\bar{h}\sqrt{\frac{\log p}{n_T}} = o(1)$, we have $\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\|_2 \lesssim \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}} \wedge \bar{h}$.

Therefore, by combining the results from the two cases discussed above, we have

$$\left\|\hat{\mathbf{w}} + \hat{\boldsymbol{\delta}} - \boldsymbol{\beta}^{(0)}\right\|_2 \le \left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right\|_2 + \left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_2 \lesssim \sqrt{\frac{s\log p}{N}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} + \sqrt{K+1}\sqrt{\frac{\log p}{n_T}}\bar{h} + \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}} \wedge \bar{h}$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$.

Since A holds, together with the condition $K^2 s\log p/n_S = O(1)$, we have

$$\sqrt{K+1}\sqrt{\frac{\log p}{n_T}}\bar{h} \le \sqrt{K+1}\frac{s\log p}{n_S} \le \sqrt{\frac{(K+1)^2 s\log p}{n_S}}\sqrt{\frac{s\log p}{N}} \lesssim \sqrt{\frac{s\log p}{N}}$$

which implies

$$\left\|\hat{\mathbf{w}} + \hat{\boldsymbol{\delta}} - \boldsymbol{\beta}^{(0)}\right\|_2 \le \left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{\delta}}\right\|_2 + \left\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\right\|_2 \lesssim \sqrt{\frac{s\log p}{N}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} + \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}} \wedge \bar{h}.$$

Case 2: Next we discuss the case when the event A^c holds, i.e.,

$$s\log p/n_S \le \bar{h}\sqrt{\log p/n_T}$$

In this case, we choose

$$\lambda_0 = c_0\sqrt{\frac{\log p}{n_S}}, \text{ and } a_k = \frac{8n_S}{N}$$

Applying Lemma 5 again, we have

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_2 \lesssim \sqrt{\frac{s\log p}{n_S}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} \tag{30}$$

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_1 \lesssim s\sqrt{\frac{\log p}{n_S}} + \sqrt{\sqrt{\frac{\log p}{n_S}}s\bar{h}} + \bar{h} \tag{31}$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$.

Plugging the new bound (31) and (30) into the arguments in Case 1 leads to

$$\left\|\hat{\mathbf{w}} + \hat{\boldsymbol{\delta}} - \boldsymbol{\beta}^{(0)}\right\|_2 \lesssim \sqrt{\frac{s\log p}{n_S}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} + \sqrt{\frac{\log p}{n_T}}\bar{h} + \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}} \wedge \bar{h}.$$

Recall that in Theorem 2 we assume $\sqrt{\log p/n_T}\bar{h} = o(1)$ and $\log p/n_T = O(1)$. Therefore, in the above bound, the third term has a smaller order comparing to the fourth term. Hence, we have

$$\left\|\hat{\mathbf{w}} + \hat{\boldsymbol{\delta}} - \boldsymbol{\beta}^{(0)}\right\|_2 \lesssim \sqrt{\frac{s\log p}{n_S}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} + \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}} \wedge \bar{h}.$$

As we assume A^c holds in this case, we have $s\log p/n_S \leq \bar{h}\sqrt{\log p/n_T}$, which further implies

$$\left\|\hat{\mathbf{w}} + \hat{\boldsymbol{\delta}} - \boldsymbol{\beta}^{(0)}\right\|_2 \lesssim \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}}.$$

Combining the results from the two cases discussed above, we have

$$\left\|\hat{\mathbf{w}} + \hat{\boldsymbol{\delta}} - \boldsymbol{\beta}^{(0)}\right\|_2 \lesssim \sqrt{\frac{s\log p}{N}} + \sqrt{\sqrt{\frac{\log p}{n_T}}\bar{h}}.$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$. The proof is then completed.

A.3 Proof of Theorem 3

We first show a lemma discussing the bias and variance components in (59), whose proof is based on the results in Javanmard and Montanari (2014) but taking into account that $\boldsymbol{\beta}^{(k)}$ is not exactly s-sparse (due to the contrast term $\boldsymbol{\delta}^{(k)}$).

Lemma 6. *Under Assumption 1 and 2 and* $\frac{s\log p}{n_S} = o(1)$, *if we construct* $\{\hat{\boldsymbol{\beta}}^{(k)}_{LASSO}\}_{k=1,\dots,K}$ *through (57) and* $\{\hat{\boldsymbol{\Theta}}^{(k)}\}_{k=1,\dots,K}$ *using (58), with parameters* $\tilde{\lambda}_k = \mu_k = c_0\sqrt{\frac{\log p}{n_S}}$ *for some universal constant* c_0, *then we have that for* $k = 1,\dots,K$,

$$\frac{1}{n_S}\left\|\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}\right\|_\infty \lesssim \sqrt{\frac{\log p}{n_S}} \tag{32}$$

and

$$\left\|\boldsymbol{b}^{(k)}\right\|_\infty := \left\|\left(\hat{\boldsymbol{\Theta}}^{(k)}\hat{\boldsymbol{\Sigma}}^{(k)} - \boldsymbol{I}\right)\left(\hat{\boldsymbol{\beta}}^{(k)}_{\text{LASSO}} - \boldsymbol{\beta}^{(k)}\right)\right\|_\infty \lesssim \frac{s\log p}{n_S} + h_k\sqrt{\frac{\log p}{n_S}}, \tag{33}$$

with probability larger than $1 - c_1\exp(-c_2\log p)$.

Now we proceed to the proof of the theorem. We first show that by reparametrization, problem (57) is essentially a special case of problem (2). Then we apply techniques similar to those used in Theorem 1 to prove the results.

Following the arguments in (35), we may reformulate problem (57) into a generalized LASSO problem:

$$\tilde{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\left\{\tilde{L}(\boldsymbol{\theta}) + \lambda_0\mathcal{R}(\boldsymbol{\theta})\right\} := \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\left\{\frac{1}{2N}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\theta}\right\|_2^2 + \lambda_0\mathcal{R}(\boldsymbol{\theta})\right\} \tag{34}$$

where

$$\tilde{\mathbf{y}} = \begin{pmatrix} \sqrt{n_S}\tilde{\boldsymbol{\beta}}^{(1)} \\ \sqrt{n_S}\tilde{\boldsymbol{\beta}}^{(2)} \\ \vdots \\ \sqrt{n_S}\tilde{\boldsymbol{\beta}}^{(K)} \\ \mathbf{y}^{(0)} \end{pmatrix}, \quad \tilde{\mathbf{X}} = \begin{pmatrix} \sqrt{n_S}\boldsymbol{I}_p & 0 & \cdots & 0 & \sqrt{n_S}\boldsymbol{I}_p \\ 0 & \sqrt{n_S}\boldsymbol{I}_p & \cdots & 0 & \sqrt{n_S}\boldsymbol{I}_p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sqrt{n_S}\boldsymbol{I}_p & \sqrt{n_S}\boldsymbol{I}_p \\ 0 & 0 & \cdots & 0 & \mathbf{X}^{(0)} \end{pmatrix}. \tag{35}$$

Similarly, we can define the random noise $\tilde{\boldsymbol{\epsilon}} = \tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\theta}^*$. The k-th block of $\tilde{\boldsymbol{\epsilon}}$ is given by $\sqrt{n_S}\left(\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}/n_S + \boldsymbol{b}^{(k)}\right)$ for $k = 1, \ldots, K$, whereas the last block is $\boldsymbol{\epsilon}^{(0)}$, the observation noise for the target model.

Following the aforementioned reformulation, we can employ an approach similar to that used in Lemma 2 to establish the result about $\left\langle \nabla\tilde{\mathcal{L}}\left(\boldsymbol{\theta}\right), \boldsymbol{\Delta}\right\rangle$. Define $\delta_k = \frac{s\log p}{N} + \frac{n_S}{N}\sqrt{\frac{\log p}{n_S}}h_k$ and $\delta_0 = \frac{Ks\log p}{N} + \sqrt{\frac{\log p}{n_S}}\bar{h}$, the result is stated as follows:

Lemma 7. *Under assumptions 1 and 2, if* $n_S \gtrsim \log p$, $\lambda_k = c_k\left(\sqrt{\frac{n_S}{N}\frac{\log p}{N}} + \delta_k\right)$ *and* $\lambda_0 = c_0\left(\sqrt{\frac{\log p}{N}} + \delta_0\right)$ *for some appropriate constants* $c_0, \ldots, c_K$*, then we have for any* $\boldsymbol{\Delta} = \left(\left(\boldsymbol{\Delta}^{(1)}\right)^\top, \ldots, \left(\boldsymbol{\Delta}^{(K)}\right)^\top, \left(\boldsymbol{\Delta}^{(0)}\right)^\top\right)^\top \in \mathbb{R}^{(K+1)p}$*,*

$$\left|\left\langle \nabla\tilde{L}\left(\boldsymbol{\theta}\right), \boldsymbol{\Delta}\right\rangle\right| \le \sum_{k=1}^{K}\frac{\lambda_k}{2}\left\|\boldsymbol{\Delta}^{(k)}\right\|_1 + \frac{\lambda_0}{2}\left\|\boldsymbol{\Delta}^{(0)}\right\|_1.$$

with probability larger than $1 - c_1\exp\left(-c_2\log p\right)$.

Notice that the only difference between Lemma 7 and Lemma 2 is the choice of $\{\lambda_k\}_{k=0\le k\le K}$. With this new choice of parameters, we can verify that if further $n_S \gg Ks\log p$, $n_S \gtrsim K^2\log p$, and $h_k \asymp \bar{h}$ for any $1 \le k \le K$, the conditions of Lemma 5 hold. Therefore, we can apply Lemma 5 and obtain

$$\|\hat{\mathbf{w}}_C - \mathbf{w}\|_2 \lesssim \sqrt{s}\lambda_0 + \sqrt{\sum_{k=1}^{K}\lambda_k h_k} \lesssim \sqrt{s}\left(\sqrt{\frac{\log p}{N}} + \delta_0\right) + \sqrt{\sum_{k=1}^{K}\left(\frac{n_S}{N}\sqrt{\frac{\log p}{n_S}} + \delta_k\right)h_k}. \tag{36}$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$. This together with the fact that $\|\mathbf{w} - \boldsymbol{\beta}^{(0)}\|_2^2 \le \epsilon_D^2$ implies the bound (13).

Furthermore, if we assume $h_k \asymp \bar{h} = O(1)$, then we have

$$s\delta_0^2 = \frac{K^2 s^3\log^2 p}{N^2} + \frac{s\log p}{n_S}\bar{h}^2 \lesssim \frac{s\log p}{N} + \sqrt{\frac{\log p}{n_S}}\bar{h} \tag{37}$$

and

$$\sum_{k=1}^{K}\delta_k h_k = \frac{s\log p}{n_S}\bar{h} + \sqrt{\frac{\log p}{n_S}}\sum_{k=1}^{K}\frac{n_S}{N}h_k^2 \lesssim \sqrt{\frac{\log p}{n_S}}\bar{h} \tag{38}$$

based on the condition that $n_S \gg Ks^2\log p$. Therefore, we have the bound (14).

A.4 Proof of Theorem 4

We use a similar line of arguments as the proof of Theorem 2. Recall that we choose

$$\lambda_k = c_0(8 \vee \frac{\bar{h}}{h_k})\left(\sqrt{\frac{n_S}{N}\frac{\log p}{N}} + \delta_k\right), \delta_k = \frac{s\log p}{N} + \frac{n_S}{N}\sqrt{\frac{\log p}{n_S}}h_k, \tag{39}$$

$$\lambda_0 = c_0\left(\sqrt{\frac{\log p}{N}}\mathbb{1}_A + \sqrt{\frac{\log p}{n_S}}\mathbb{1}_{A^c} + \delta_0\right), \delta_0 = \frac{Ks\log p}{N} + \sum_{k=1}^{K}\frac{n_S}{N}\sqrt{\frac{\log p}{n_S}}h_k. \tag{40}$$

We can verify that if $n_S \gg Ks\log p$, $n_S \gtrsim K^2 \log p$ and $h_k \asymp \bar{h}$ for any $1 \le k \le K$, this choice of parameters satisfies the conditions in Lemma 5. Therefore we can apply Lemma 5 and obtain

$$\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_2 \lesssim \sqrt{s}\lambda_0 + \sqrt{\sum_{k=1}^{K} \lambda_k h_k} \lesssim \sqrt{\frac{s\log p}{N}}\mathbb{1}_A + \sqrt{\frac{s\log p}{n_S}}\mathbb{1}_{A^c} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}} + \sqrt{s}\delta_0 + \sqrt{\sum_{k=1}^{K} \delta_k h_k} \tag{41}$$

$$\begin{aligned}\|\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}}\|_1 &\lesssim s\lambda_0 + \sqrt{s}\sqrt{\sum_{k=1}^{K} \lambda_k h_k} + \frac{\sum_{k=1}^{K}\lambda_k}{\lambda_0}\bar{h} \\ &\lesssim \sqrt{\frac{s^2\log p}{N}}\mathbb{1}_A + \sqrt{\frac{s^2\log p}{n_S}}\mathbb{1}_{A^c} + \sqrt{\sqrt{\frac{\log p}{n_S}}s\bar{h}} + s\delta_0 + \sqrt{\sum_{k=1}^{K} s\delta_k h_k} + \sqrt{K}\bar{h}\mathbb{1}_A + \bar{h}\mathbb{1}_{A^c}\end{aligned} \tag{42}$$

by plugging in the choice of λ_0 and λ_ks and using the fact that $\sum_{k=1}^{K} \delta_k = \delta_0$.

To prove the theorem, it suffices to show that in the bounds (41) and (42), the terms involving δ_0 and δ_k are of orders that align with some other terms, then we can follow exactly the same proof in Theorem 2 to prove the result. To show this, we can use the results in (37) and (38), which gives us

$$\sqrt{s}\delta_0 + \sqrt{\sum_{k=1}^{K} \delta_k h_k} \lesssim \sqrt{\frac{s\log p}{N}} + \sqrt{\sqrt{\frac{\log p}{n_S}}\bar{h}}. \tag{43}$$

With this, the proof is completed.

B Proof of Technical Lemmas

B.1 Proof of Lemma 2

By definition, $-\nabla\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \left(\frac{1}{N}\left(\mathbf{X}^{(1)}\right)^\top \boldsymbol{\epsilon}^{(1)}, \ldots, \frac{1}{N}\left(\mathbf{X}^{(K)}\right)^\top \boldsymbol{\epsilon}^{(K)}, \frac{1}{N}\sum_{k=0}^{K}\left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}\right)^\top$.

Therefore, by Hölder's inequality, we have

$$\begin{aligned}|\langle \nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\Delta}\rangle| &= \sum_{k=1}^{K}\left|\left\langle \frac{1}{N}\left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}, \boldsymbol{\Delta}^{(k)}\right\rangle\right| + \left|\left\langle \frac{1}{N}\sum_{k=0}^{K}\left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}, \boldsymbol{\Delta}^{(0)}\right\rangle\right| \\ &\le \sum_{k=1}^{K}\left\|\frac{1}{N}\left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}\right\|_\infty \left\|\boldsymbol{\Delta}^{(k)}\right\|_1 + \left\|\frac{1}{N}\sum_{k=0}^{K}\left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}\right\|_\infty \left\|\boldsymbol{\Delta}^{(0)}\right\|_1.\end{aligned}$$

Recall that we define $n_k = n_S$ for $1 \le k \le K$ and $n_k = n_T$ for $k = 0$. Define

$$\mathcal{A}^{(k)} = \left\{\max_{1\le j\le p}\left\{\frac{1}{n_k}\sum_{i=1}^{n_k}\left(x_{ij}^{(k)}\right)^2\right\} \le 2\max_{1\le j\le p} E\left(x_{ij}^{(k)}\right)^2\right\},$$

and

$$\mathcal{A} = \left\{\max_{1\le j\le p}\left\{\frac{1}{N}\sum_{k=0}^{K}\sum_{i=1}^{n_k}\left(x_{ij}^{(k)}\right)^2\right\} \le 2\max_{1\le k\le K, 1\le j\le p} E\left(x_{ij}^{(k)}\right)^2\right\}.$$

Since $\mathbf{X}^{(k)}$ is sub-Gaussian with uniformly bounded second moment and $n_S \gtrsim \log p$, we have $P\left(\overline{\mathcal{A}^{(k)}}\right) \le c_1\exp(-c_2 n_S)$ for $1 \le k \le K$ and $P\left(\overline{\mathcal{A}}\right) \le c_1\exp(-c_2 N)$ for some universal constants c_1 and c_2.

In addition, as $\boldsymbol{\epsilon}^{(k)} \sim N\left(0, \sigma_k^2\boldsymbol{I}\right)$ for some finite σ_k, by Proposition 5.10 in Vershynin (2010), we can establish that with some universal constant c_3, for $1 \le k \le K$,

$$P\left(\max_{1\le j\le p}\left|\frac{1}{n_S}\sum_{i=1}^{n_S}\epsilon_i^{(k)}x_{ij}^{(k)}\right| \ge t\right) \le P\left(\max_{1\le j\le p}\left|\frac{1}{n_S}\sum_{i=1}^{n_S}\epsilon_i^{(k)}x_{ij}^{(k)}\right| \ge t \;\middle|\; \mathcal{A}^{(k)}\right) + P(\overline{\mathcal{A}^{(k)}})$$

$$\le p\cdot e\cdot\exp\left(-\frac{c_3 n_S t^2}{4\sigma_k^2\max_{1\le j\le p}E\left(x_{ij}^{(k)}\right)^2}\right) + c_1\exp\left(-c_2 n_S\right).$$

Since by Assumption 1, there exist a constant c such that $\max_{1\le k\le K}\Lambda_{\max}(\boldsymbol{\Sigma}^{(k)}) \le c$, so $\max_{1\le j\le p}E\left(x_{ij}^{(k)}\right)^2$ is uniformly bounded above. Therefore for $1 \le k \le K$, by choosing $t = \sqrt{c_4\log p/n_S}$ for some constant c_4, with probability larger than $1 - c_1\exp\left(-c_2\log p\right)$, we have

$$\left\|\frac{1}{N}\left(\mathbf{X}^{(k)}\right)^\top\boldsymbol{\epsilon}^{(k)}\right\|_\infty \lesssim \frac{n_S}{N}\sqrt{\frac{\log p}{n_S}} = \sqrt{\frac{n_S}{N}}\sqrt{\frac{\log p}{N}}.$$

Similarly, we have

$$P\left(\max_{1\le j\le p}\left|\frac{1}{N}\sum_{k=0}^{K}\sum_{i=1}^{n_k}\epsilon_i^{(k)}x_{ij}^{(k)}\right| \ge t\right) \le P\left(\max_{1\le j\le p}\left|\frac{1}{N}\sum_{k=0}^{K}\sum_{i=1}^{n_k}\epsilon_i^{(k)}x_{ij}^{(k)}\right| \ge t \;\middle|\; \mathcal{A}\right) + P(\overline{\mathcal{A}})$$

$$\le p\cdot e\cdot\exp\left(-\frac{c_4 N t^2}{4\max_{0\le k\le K,1\le j\le p}\sigma_k^2 E\left(x_{ij}^{(k)}\right)^2}\right) + c_1\exp\left(-c_2 N\right)$$

So we have with probability larger than $1 - c_1\exp\left(-c_2\log p\right)$,

$$\left\|\sum_{k=0}^{K}\frac{1}{N}\left(\mathbf{X}^{(k)}\right)^\top\boldsymbol{\epsilon}^{(k)}\right\|_\infty \lesssim \sqrt{\frac{\log p}{N}}$$

Therefore, by choosing $\lambda_k = a_k\lambda_k = c_0\sqrt{\frac{n_S}{N}}\sqrt{\frac{\log p}{N}}$ and $\lambda_0 = c_0\sqrt{\frac{\log p}{N}}$ for some sufficiently large constant c_0, we have the desired result.

B.2 Proof of Lemma 3

Define S is the support set of $\boldsymbol{\theta}^{(0)} = \boldsymbol{\beta}^{(0)}$, and S^c as its complement. Then we have $\|\boldsymbol{\theta}_S^*\|_0 = s$ and $\|\boldsymbol{\theta}_{S^c}^*\|_1 \le \sum_{k=1}^K h_k \le (K+1)\bar{h}$ as $n_S \ge n_T$.

We define $F : \mathbb{R}^{(K+1)p} \to \mathbb{R}$ as

$$F(\boldsymbol{\Delta}) = \mathcal{L}\left(\boldsymbol{\theta}^* + \boldsymbol{\Delta}\right) - \mathcal{L}\left(\boldsymbol{\theta}^*\right) + \lambda_0\mathcal{R}\left(\boldsymbol{\theta}^* + \boldsymbol{\Delta}\right) - \lambda_0\mathcal{R}\left(\boldsymbol{\theta}^*\right),$$

and $\hat{\boldsymbol{\theta}}$ as the solution to the problem (17). We then have $\hat{\boldsymbol{\Delta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* = \underset{\boldsymbol{\Delta}}{\operatorname{argmin}}F(\boldsymbol{\Delta})$ and $F(0) = 0$. Consequently, it follows that $F(\hat{\boldsymbol{\Delta}}) \le 0$.

Since $\mathcal{L}$ is a convex function, by Lemma 2, we can choose $\lambda_k = c_0\sqrt{\frac{n_S}{N}}\sqrt{\frac{\log p}{N}}$ and $\lambda_0 = c_0\sqrt{\frac{\log p}{N}}$ so that

$$\mathcal{L}\left(\boldsymbol{\theta}^* + \hat{\boldsymbol{\Delta}}\right) - \mathcal{L}\left(\boldsymbol{\theta}^*\right) \ge \left\langle\nabla\mathcal{L}\left(\boldsymbol{\theta}^*\right), \hat{\boldsymbol{\Delta}}\right\rangle \ge -\sum_{k=1}^{K}\frac{\lambda_k}{2}\left\|\hat{\boldsymbol{\Delta}}^{(k)}\right\|_1 - \frac{\lambda_0}{2}\left\|\hat{\boldsymbol{\Delta}}^{(0)}\right\|_1 \tag{44}$$

with probability larger than $1 - c_1\exp(c_2\log p)$.

Since the ℓ_1-norm function is decomposable and $\|\boldsymbol{\theta}_{S^c}^{(0)}\|_1 = 0$, by triangle inequality we have

$$
\begin{aligned}
\lambda_0 \mathcal{R}\left(\boldsymbol{\theta}^* + \boldsymbol{\Delta}\right) - \lambda_0 \mathcal{R}\left(\boldsymbol{\theta}^*\right) &= \sum_{k=1}^{K} \lambda_k \left(\left\| \boldsymbol{\theta}^{(k)} + \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 - \left\| \boldsymbol{\theta}^{(k)} \right\|_1 \right) + \lambda_0 \left(\left\| \boldsymbol{\theta}^{(0)} + \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1 - \left\| \boldsymbol{\theta}^{(0)} \right\|_1 \right) \\
&\geq \sum_{k=1}^{K} \lambda_k \left(\left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 - 2 \left\| \boldsymbol{\theta}^{(k)} \right\|_1 \right) + \lambda_0 \left(\left\| \hat{\boldsymbol{\Delta}}_{S^c}^{(0)} \right\|_1 - \left\| \hat{\boldsymbol{\Delta}}_{S}^{(0)} \right\|_1 - 2 \left\| \boldsymbol{\theta}_{S^c}^{(0)} \right\|_1 \right) \\
&\geq \sum_{k=1}^{K} \lambda_k \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 - 2 \sum_{k=1}^{K} \lambda_k h_k + \lambda_0 \left(\left\| \hat{\boldsymbol{\Delta}}_{S^c}^{(0)} \right\|_1 - \left\| \hat{\boldsymbol{\Delta}}_{S}^{(0)} \right\|_1 \right). \qquad (45)
\end{aligned}
$$

Combining (44) and (45) yields

$$
0 \geq F(\hat{\boldsymbol{\Delta}}) \geq \sum_{k=1}^{K} \frac{\lambda_k}{2} \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 - 2 \sum_{k=1}^{K} \lambda_k h_k + \frac{\lambda_0}{2} \left(\left\| \hat{\boldsymbol{\Delta}}_{S^c}^{(0)} \right\|_1 - 3 \left\| \hat{\boldsymbol{\Delta}}_{S}^{(0)} \right\|_1 \right) \qquad (46)
$$

which leads to the following inequality:

$$
\sum_{k=0}^{K} \lambda_k \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 + \lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1 \leq 4\lambda_0 \left\| \hat{\boldsymbol{\Delta}}_{S}^{(0)} \right\|_1 + 4 \sum_{k=1}^{K} \lambda_k h_k.
$$

Recalling that we define $\hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} = \sum_{k=1}^{K} \frac{n_S}{N} \hat{\boldsymbol{\Delta}}^{(k)} + \hat{\boldsymbol{\Delta}}^{(0)}$, it follows that

$$
\begin{aligned}
\sum_{k=1}^{K} \lambda_k \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 + \lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1 &\leq 4\lambda_0 \left\| \hat{\boldsymbol{\Delta}}_{S}^{(0)} \right\|_1 + 4 \sum_{k=1}^{K} \lambda_k h_k \\
&\leq 4\lambda_0 \left\| \hat{\boldsymbol{\Delta}}_{S}^{\boldsymbol{w}} \right\|_1 + 4 \sum_{k=1}^{K} \lambda_0 \frac{n_S}{N} \left\| \hat{\boldsymbol{\Delta}}_{S}^{(k)} \right\|_1 + 4 \sum_{k=1}^{K} \lambda_k h_k \qquad (47)
\end{aligned}
$$

Since by the choice of parameters, we have $\frac{\lambda_k}{2} \geq 4\lambda_0 \frac{n_S}{N}$, so reorganizing (47) yields

$$
\sum_{k=1}^{K} \lambda_k \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 + 2\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1 \leq 8\lambda_0 \left\| \hat{\boldsymbol{\Delta}}_{S}^{\boldsymbol{w}} \right\|_1 + 8 \sum_{k=1}^{K} \lambda_k h_k \qquad (48)
$$

which is as desired.

B.3 Proof of Lemma 4

Applying Lemma 1 and Jensen's inequality, along with some algebra, we have

$$\begin{aligned}
&\mathcal{L}\left(\boldsymbol{\theta}^* + \hat{\boldsymbol{\Delta}}\right) - \mathcal{L}\left(\boldsymbol{\theta}^*\right) - \left\langle \nabla \mathcal{L}\left(\boldsymbol{\theta}^*\right), \hat{\boldsymbol{\Delta}} \right\rangle \\
&= \hat{\boldsymbol{\Delta}}^\top \nabla^2 \mathcal{L}\left(\boldsymbol{\theta}^* + \gamma \hat{\boldsymbol{\Delta}}\right) \hat{\boldsymbol{\Delta}} \quad (\gamma \in (0,1)) \\
&= \sum_{k=1}^K \frac{n_S}{N} \left(\hat{\boldsymbol{\Delta}}^{(k)}\right)^\top \hat{\boldsymbol{\Sigma}}^{(k)} \hat{\boldsymbol{\Delta}}^{(k)} + 2 \sum_{k=1}^K \frac{n_S}{N} \left(\hat{\boldsymbol{\Delta}}^{(k)}\right)^\top \hat{\boldsymbol{\Sigma}}^{(k)} \hat{\boldsymbol{\Delta}}^{(0)} + \left(\hat{\boldsymbol{\Delta}}^{(0)}\right)^\top \left(\sum_{k=1}^K \frac{n_S}{N} \hat{\boldsymbol{\Sigma}}^{(k)} + \frac{n_T}{N} \hat{\boldsymbol{\Sigma}}^{(0)}\right) \hat{\boldsymbol{\Delta}}^{(0)} \\
&= \sum_{k=1}^K \frac{n_S}{N} \left(\hat{\boldsymbol{\Delta}}^{(k)} + \hat{\boldsymbol{\Delta}}^{(0)}\right)^\top \hat{\boldsymbol{\Sigma}}^{(k)} \left(\hat{\boldsymbol{\Delta}}^{(k)} + \hat{\boldsymbol{\Delta}}^{(0)}\right) + \frac{n_T}{N} \left(\hat{\boldsymbol{\Delta}}^{(0)}\right)^\top \hat{\boldsymbol{\Sigma}}^{(0)} \hat{\boldsymbol{\Delta}}^{(0)} \\
&\geq \sum_{k=1}^K \frac{n_S \alpha_k}{N} \left\| \hat{\boldsymbol{\Delta}}^{(k)} + \hat{\boldsymbol{\Delta}}^{(0)} \right\|_2^2 + \frac{n_T \alpha_0}{N} \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_2^2 - \mathcal{R}\left(\hat{\boldsymbol{\Delta}}\right) \\
&\geq \alpha_{\min} \left\| \sum_{k=1}^K \frac{n_S}{N} \hat{\boldsymbol{\Delta}}^{(k)} + \hat{\boldsymbol{\Delta}}^{(0)} \right\|_2^2 - \mathcal{R}\left(\hat{\boldsymbol{\Delta}}\right) \\
&= \alpha_{\min} \left\| \hat{\boldsymbol{\Delta}}^{\boldsymbol{w}} \right\|_2^2 - \mathcal{R}\left(\hat{\boldsymbol{\Delta}}\right) \qquad (49)
\end{aligned}$$

with probability larger than $1 - c_1 \exp(-c_2 n_T)$, where we define $\alpha_{\min} := \min_{0 \leq k \leq K} \alpha_k$ and

$$\mathcal{R}\left(\hat{\boldsymbol{\Delta}}\right) := \sum_{k=1}^K \frac{n_S \beta_k}{N} \frac{\log p}{n_S} \left\| \hat{\boldsymbol{\Delta}}^{(k)} + \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1^2 + \frac{n_T \beta_0}{N} \frac{\log p}{n_T} \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1^2 .$$

Notice that by triangle inequality,

$$\begin{aligned}
\mathcal{R}\left(\hat{\boldsymbol{\Delta}}\right) &= \sum_{k=1}^K \frac{\beta_k \log p}{N} \left\| \hat{\boldsymbol{\Delta}}^{(k)} + \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1^2 + \frac{\beta_0 \log p}{N} \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1^2 \\
&\leq \sum_{k=1}^K \frac{2\beta_k \log p}{N} \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1^2 + \sum_{k=0}^K \frac{2\beta_k \log p}{N} \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1^2
\end{aligned}$$

According to the restricted set of directions outlined in (48), it holds that

$$\sum_{k=1}^K \lambda_k \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 + \lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1 \leq \sum_{k=1}^K \lambda_k \left\| \hat{\boldsymbol{\Delta}}^{(k)} \right\|_1 + 2\lambda_0 \left\| \hat{\boldsymbol{\Delta}}^{(0)} \right\|_1 \leq 8\lambda_0 \left\| \hat{\boldsymbol{\Delta}}_S^{\boldsymbol{w}} \right\|_1 + 8 \sum_{k=1}^K \lambda_k h_k .$$

with probability larger than $1 - c_1 \exp(-c_2 \log p)$.

So if we define $\beta_{\max} = \max_{0\le k\le K}\beta_k$, by triangle inequality and the fact that $|S| = s$, we then have

$$\begin{aligned}
\mathcal{R}\left(\hat{\mathbf{\Delta}}\right) &\le \frac{2\beta_{\max}\log p}{N}\left(\sum_{k=1}^{K}\frac{\lambda_k^2}{\lambda_k^2}\left\|\hat{\mathbf{\Delta}}^{(k)}\right\|_1^2 + (K+1)\frac{\lambda_0^2}{\lambda_0^2}\left\|\hat{\mathbf{\Delta}}^{(0)}\right\|_1^2\right)\\
&\le \frac{2\beta_{\max}}{\lambda_k^2\wedge(\lambda_0^2/(K+1))}\frac{\log p}{N}\left(\sum_{k=1}^{K}\lambda_k^2\left\|\hat{\mathbf{\Delta}}^{(k)}\right\|_1^2 + \lambda_0^2\left\|\hat{\mathbf{\Delta}}^{(0)}\right\|_1^2\right)\\
&\le \frac{2\beta_{\max}}{\lambda_k^2\wedge(\lambda_0^2/(K+1))}\frac{\log p}{N}\left(\sum_{k=1}^{K}\lambda_k\left\|\hat{\mathbf{\Delta}}^{(k)}\right\|_1 + \lambda_0\left\|\hat{\mathbf{\Delta}}^{(0)}\right\|_1\right)^2\\
&\le \frac{2\beta_{\max}}{\lambda_k^2\wedge(\lambda_0^2/(K+1))}\frac{\log p}{N}\left(8\lambda_0\left\|\hat{\mathbf{\Delta}}_S^{\boldsymbol{w}}\right\|_1 + 8\sum_{k=1}^{K}\lambda_k h_k\right)^2\\
&\le \frac{2\beta_{\max}}{\lambda_k^2\wedge(\lambda_0^2/(K+1))}\frac{\log p}{N}\left(128\lambda_0^2\left\|\hat{\mathbf{\Delta}}_S^{\boldsymbol{w}}\right\|_1^2 + 128\left(\sum_{k=1}^{K}\lambda_k h_k\right)^2\right)\\
&\le \frac{2\beta_{\max}}{\lambda_k^2\wedge(\lambda_0^2/(K+1))}\frac{\log p}{N}\left(128s\lambda_0^2\left\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\right\|_2^2 + 128\left(\sum_{k=1}^{K}\lambda_k h_k\right)^2\right)
\end{aligned}$$

Recall that we introduce the shorthand $u_n = \frac{256\beta_{\max}\lambda_0^2}{\alpha_{\min}\lambda_k^2\wedge(\lambda_0^2/(K+1))}\frac{s\log p}{N}$, and $v_n = \frac{256\beta_{\max}}{\lambda_k^2\wedge(\lambda_0^2/(K+1))}\frac{\log p}{N}\left(\sum_{k=1}^{K}\lambda_k h_k\right)$, combining the above argument with (49) leads to

$$\mathcal{L}\left(\boldsymbol{\theta}^* + \hat{\mathbf{\Delta}}\right) - \mathcal{L}\left(\boldsymbol{\theta}^*\right) - \left\langle\nabla\mathcal{L}\left(\boldsymbol{\theta}^*\right), \hat{\mathbf{\Delta}}\right\rangle \ge (1-u_n)\alpha_{\min}\left\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\right\|_2^2 - v_n\sum_{k=1}^{K}\lambda_k h_k \tag{50}$$

with probability larger than $1 - c_1\exp(-c_2 n_T) - c_3\exp(-c_4\log p)$, which finishes the proof.

B.4 Proof of Lemma 5

The result in ℓ_2-norm follows directly from the proof of Theorem 1. Here we investigate the upper bound for the estimation error in ℓ_1-norm, i.e., $\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\|_1$. We categorize the problem into two cases based on the relationship between $\frac{1}{4}\lambda_0\left\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\right\|_1$ and $2\sum_{k=1}^{K}\lambda_k h_k$, and discuss by cases.

Starting with the first scenario where $\frac{1}{4}\lambda_0\left\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\right\|_1 > 2\sum_{k=1}^{K}\lambda_k h_k$, in such case (22) implies

$$0 \ge \hat{\mathbf{\Delta}}^\top\hat{\mathbf{\Sigma}}\hat{\mathbf{\Delta}} - \frac{7}{4}\lambda_0\left\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\right\|_1 + 2\lambda_0\left\|\hat{\mathbf{\Delta}}_{S^c}^{\boldsymbol{w}}\right\|_1 \ge -\frac{7}{4}\lambda_0\left\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\right\|_1 + 2\lambda_0\left\|\hat{\mathbf{\Delta}}_{S^c}^{\boldsymbol{w}}\right\|_1,$$

which implies $\frac{1}{4}\|\hat{\mathbf{\Delta}}_{S^c}^{\boldsymbol{w}}\|_1 \le \frac{7}{4}\|\hat{\mathbf{\Delta}}_S^{\boldsymbol{w}}\|_1$, and $\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\|_1 \le 8\|\hat{\mathbf{\Delta}}_S^{\boldsymbol{w}}\|_1 \le 8\sqrt{s}\|\hat{\mathbf{\Delta}}_S^{\boldsymbol{w}}\|_2 \le 8\sqrt{s}\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\|_2$. Therefore, in this case, we obtain

$$\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\|_1 \lesssim s\lambda_0 + \sqrt{s}\sqrt{\sum_{k=1}^{K}\lambda_k h_k}$$

with probability larger than $1 - c_1\exp(c_2 n_T) - c_3\exp(c_4\log p)$.

We now transit to the second scenario when $\frac{1}{4}\lambda_0\left\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\right\|_1 \le 2\sum_{k=1}^{K}\lambda_k h_k$. In this instance, we directly obtain

$$\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\|_1 \le \frac{8\sum_{k=1}^{K}\lambda_k h_k}{\lambda_0}$$

Taking into account both the discussed scenarios, we can conclude that

$$\|\hat{\mathbf{\Delta}}^{\boldsymbol{w}}\|_1 \lesssim s\lambda_0 + \sqrt{s}\sqrt{\sum_{k=1}^{K}\lambda_k h_k} + \frac{\sum_{k=1}^{K}\lambda_k h_k}{\lambda_0} \tag{51}$$

with probability larger than $1 - c_1 \exp(c_2 n_T) - c_3 \exp(c_4 \log p)$, which is as desired.

B.5 Proof of Lemma 6

We start with the first term, $\frac{1}{n_S} \hat{\mathbf{\Theta}}^{(k)} \left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}$. Let $\left(\boldsymbol{a}_j^{(k)}\right)^\top := \boldsymbol{e}_j^\top \hat{\mathbf{\Theta}}^{(k)} \left(\mathbf{X}^{(k)}\right)^\top$. Similar to the proof of Lemma 2, by Vershynin (2010), Proposition 5.10 and a union bound,

$$P\left(\max_{1\leq j\leq p}\left|\frac{1}{n_S}\left(\boldsymbol{a}_j^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)}\right| > t \;\middle|\; \left\{\boldsymbol{a}_j^{(k)}\right\}_{1\leq j\leq p}\right) \leq p \exp\left(-\frac{c n_S^2 t^2}{\sigma_k^2 \max_{1\leq j\leq p}\left\|\boldsymbol{a}_j^{(k)}\right\|_2^2}\right)$$

for some universal constant $c > 0$. Therefore, to prove (32), it suffices to bound

$$c_\Omega := \frac{1}{n_S}\max_{1\leq j\leq p}\left\|\boldsymbol{a}_j^{(k)}\right\|_2^2 = \max_{1\leq j\leq p}\left(\hat{\mathbf{\Theta}}^{(k)}\hat{\mathbf{\Sigma}}^{(k)}\left(\hat{\mathbf{\Theta}}^{(k)}\right)^\top\right)_{j,j}.$$

In order to accomplish this, we employ Lemma 23 from Lee et al. (2017), which is formulated as follows.

Lemma 8. *Under the conditions of Lemma 6,*

$$P\left(\max_{1\leq j\leq p}\left(\mathbf{\Sigma}^{(k)}\right)_j^{-1}\hat{\mathbf{\Sigma}}^{(k)}\left(\mathbf{\Sigma}^{(k)}\right)_j^{-1} > 2\max_{1\leq j\leq p}\left(\mathbf{\Sigma}^{(k)}\right)_{j,j}^{-1}\right) \leq 2pe^{-c_1 n_S}$$

for some universal constant $c_1 > 0$.

Since $\Theta^{(k)}$ is the solution of problem (58), combining the optimally condition with Lemma 8 implies

$$\max_{1\leq j\leq p}\left(\hat{\mathbf{\Theta}}^{(k)}\hat{\mathbf{\Sigma}}^{(k)}\left(\hat{\mathbf{\Theta}}^{(k)}\right)^\top\right)_{j,j} \leq \max_{1\leq j\leq p}\left(\mathbf{\Sigma}^{(k)}\right)_j^{-1}\hat{\mathbf{\Sigma}}^{(k)}\left(\mathbf{\Sigma}^{(k)}\right)_j^{-1} \leq 2\max_{1\leq j\leq p}\left(\mathbf{\Sigma}^{(k)}\right)_{j,j}^{-1}$$

with probability at least $1 - 2pe^{-c_1 n_S}$. By assumption 1, $\left(\mathbf{\Sigma}^{(k)}\right)_{j,j}^{-1}$ is bounded above. Therefore (32) holds.

Next, we aim at the second term, $\boldsymbol{b}^{(k)}$. Applying Hölder's inequality to each component we obtain

$$\begin{aligned}\left\|\boldsymbol{b}^{(k)}\right\|_\infty &= \left\|\left(\hat{\mathbf{\Theta}}^{(k)}\hat{\mathbf{\Sigma}}^{(k)} - \boldsymbol{I}\right)\left(\hat{\boldsymbol{\beta}}_{\text{LASSO}}^{(k)} - \boldsymbol{\beta}^{(k)}\right)\right\|_\infty \\ &\leqslant \max_{1\leq j\leq p}\left\|\hat{\mathbf{\Theta}}_j^{(k)}\hat{\mathbf{\Sigma}}^{(k)} - \boldsymbol{e}_j^\top\right\|_\infty\left\|\hat{\boldsymbol{\beta}}_{\text{LASSO}}^{(k)} - \boldsymbol{\beta}^{(k)}\right\|_1\end{aligned}$$

where recall that $\hat{\mathbf{\Theta}}_j^{(k)}$ denotes the jth row of $\hat{\mathbf{\Theta}}^{(k)}$. By the optimality condition of (2) and the choice of μ_k, we have

$$\max_{1\leq j\leq p}\left\|\hat{\mathbf{\Theta}}_j^{(k)}\hat{\mathbf{\Sigma}}^{(k)} - \boldsymbol{e}_j^\top\right\|_\infty \lesssim \sqrt{\frac{\log p}{n_S}}.$$

In addition, recall that $\left\|\boldsymbol{\beta}^{(0)}\right\|_0 = s$ and $\left\|\boldsymbol{\beta}^{(0)} - \boldsymbol{\beta}^{(k)}\right\|_1 \leq h_k$, so we can show the following Lemma:

Lemma 9. *Under the condition of Lemma 6, we have*

$$\left\|\hat{\boldsymbol{\beta}}_{\text{LASSO}}^{(k)} - \boldsymbol{\beta}^{(k)}\right\|_1 \leqslant s\sqrt{\frac{\log p}{n_S}} + h_k$$

with probability at least $1 - c_1 p^{-c_2}$.

The proof of Lemma 9 comes from a direct application of Lemma 1 in Li et al. (2022). Integrating the above arguments leads to $\left\|\boldsymbol{b}^{(k)}\right\|_\infty \lesssim_P \frac{s\log p}{n_S} + h_k\sqrt{\frac{\log p}{n_S}}$ with probability larger than $1 - c_1\exp(c_2\log p)$. This finishes the proof.

B.6 Proof of Lemma 7

Following the proof of Lemma 2, by applying Hölder's inequality, we obtain

$$\begin{aligned}
|\langle \nabla \mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\Delta}\rangle| &= \sum_{k=1}^{K} \left|\left\langle \frac{\sqrt{n_S}}{N}\tilde{\boldsymbol{\epsilon}}^{(k)}, \boldsymbol{\Delta}^{(k)}\right\rangle\right| + \left|\left\langle \sum_{k=1}^{K} \frac{\sqrt{n_S}}{N}\tilde{\boldsymbol{\epsilon}}^{(k)} + \frac{1}{N}\left(\mathbf{X}^{(0)}\right)^{\top}\boldsymbol{\epsilon}^{(0)}, \boldsymbol{\Delta}^{(0)}\right\rangle\right| \\
&\le \sum_{k=1}^{K} \left\|\frac{\sqrt{n_S}}{N}\tilde{\boldsymbol{\epsilon}}^{(k)}\right\|_{\infty}\left\|\boldsymbol{\Delta}^{(k)}\right\|_{1} + \left\|\sum_{k=1}^{K} \frac{\sqrt{n_S}}{N}\tilde{\boldsymbol{\epsilon}}^{(k)} + \frac{1}{N}\left(\mathbf{X}^{(0)}\right)^{\top}\boldsymbol{\epsilon}^{(0)}\right\|_{\infty}\left\|\boldsymbol{\Delta}^{(0)}\right\|_{1} \\
&\le \sum_{k=1}^{K} \left\|\frac{1}{N}\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^{\top}\boldsymbol{\epsilon}^{(k)} + \frac{n_S}{N}\boldsymbol{b}^{(k)}\right\|_{\infty}\left\|\boldsymbol{\Delta}^{(k)}\right\|_{1} \\
&\quad + \left\|\frac{1}{N}\left(\sum_{k=1}^{K}\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^{\top}\boldsymbol{\epsilon}^{(k)} + \left(\mathbf{X}^{(0)}\right)^{\top}\boldsymbol{\epsilon}^{(0)}\right) + \sum_{k=1}^{K}\frac{n_S}{N}\boldsymbol{b}^{(k)}\right\|_{\infty}\left\|\boldsymbol{\Delta}^{(0)}\right\|_{1}
\end{aligned}$$

We start with the first term on the right-hand side of the inequality. In Lemma 6 we have shown that with probability larger than $1 - c_1\exp(-c_2\log p)$,

$$\frac{1}{n_S}\left\|\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^{\top}\boldsymbol{\epsilon}^{(k)}\right\|_{\infty} \lesssim \sqrt{\frac{\log p}{n_S}} \quad \text{and} \quad \left\|\boldsymbol{b}^{(k)}\right\|_{\infty} \lesssim \frac{s\log p}{n_S} + h_k\sqrt{\frac{\log p}{n_S}}. \tag{52}$$

Thus in order to guarantee

$$\lambda_k \ge \left\|\frac{1}{N}\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^{\top}\boldsymbol{\epsilon}^{(k)} + \frac{n_S}{N}\boldsymbol{b}^{(k)}\right\|_{\infty},$$

it suffices to choose $\lambda_k = c_k\left(\sqrt{\frac{n_S}{N}\frac{\log p}{N}} + \frac{s\log p}{N} + \frac{n_S}{N}\sqrt{\frac{\log p}{n_S}}h_k\right)$ for some sufficiently large constant c_k.

Next, we shift our focus to the second term. Similar to the arguments in Proposition 6, we denote $\left(\boldsymbol{a}_j^{(k)}\right)^{\top} := \boldsymbol{e}_j^{\top}\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^{\top}$ for $k = 1, \ldots, K$ and $\left(\boldsymbol{a}_j^{(0)}\right)^{\top} := \boldsymbol{e}_j^{\top}\left(\mathbf{X}^{(0)}\right)^{\top}$. By Vershynin (2010), Proposition 5.10 and a union bound,

$$P\left(\max_{1\le j\le p}\left|\frac{1}{N}\sum_{k=0}^{K}\left(\boldsymbol{a}_j^{(k)}\right)^{\top}\boldsymbol{\epsilon}^{(k)}\right| > t \;\middle|\; \left\{\boldsymbol{a}_j^{(k)}\right\}_{1\le j\le p, 0\le k\le K}\right) \le p\exp\left(-\frac{cN^2t^2}{\max_{1\le j\le p, 0\le k\le K}\sigma_k^2\left\|\boldsymbol{a}_j^{(k)}\right\|_2^2}\right)$$

According to Proposition 6 and Assumption 1, $\max_{0\le k\le K, 1\le j\le p}\left\|\boldsymbol{a}_j^{(k)}\right\|_2^2$ is bounded above with probability larger than $1 - c_1\exp(-c_2\log p)$. This result together with (52) indicates that

$$\left\|\frac{1}{N}\left(\sum_{k=1}^{K}\hat{\boldsymbol{\Theta}}^{(k)}\left(\mathbf{X}^{(k)}\right)^{\top}\boldsymbol{\epsilon}^{(k)} + \left(\mathbf{X}^{(0)}\right)^{\top}\boldsymbol{\epsilon}^{(0)}\right) + \sum_{k=1}^{K}\frac{n_S}{N}\boldsymbol{b}^{(k)}\right\|_{\infty} \lesssim \sqrt{\frac{\log p}{N}} + \frac{Ks\log p}{N} + \sum_{k=1}^{K}\frac{n_S}{N}\sqrt{\frac{\log p}{n_S}}h_k$$

so it suffices to choose $\lambda_0 = c_0\left(\sqrt{\frac{\log p}{N}} + \frac{Ks\log p}{N} + \sum_{k=1}^{K}\frac{n_S}{N}\sqrt{\frac{\log p}{n_S}}h_k\right)$ for some constant c_0. This finishes the proof.

C Impact of Covariate Shift on C_Σ

Recall constant C_Σ defined as

$$C_\Sigma := 1 + \max_{j\le p}\max_{k}\left\|e_j^{\top}\left(\boldsymbol{\Sigma}^{(k)} - \boldsymbol{\Sigma}^{(0)}\right)\left(\sum_{1\le k\le K}\frac{1}{K}\boldsymbol{\Sigma}^{(k)}\right)^{-1}\right\|_1.$$

Clearly, C_Σ depends on the difference between the source and target covariance matrices. In the following, we provide an example where C_Σ diverges with p.

Let $K = 1$, i.e., there is only one source dataset, and the target data covariance matrix $\mathbf{\Sigma}^{(0)} = \mathbf{I}$. The source data covariance $\mathbf{\Sigma}^{(k)}$ is constructed as $\mathbf{\Sigma}^{(k)} = \alpha\mathbf{A} + (1-\alpha)\mathbf{I}$, where $\mathbf{A}$ is an arrowhead matrix with all nonzero elements equal to one. Next, we provide the choice of α such that the eigenvalues of $\mathbf{\Sigma}^{(k)}$ are bounded, thus satisfying Assumption 1. To this end, we first provide an expression of the eigenvalues of $\mathbf{A}$. Applying Cauchy's interlace theorem, we have $\Lambda_{\min}(\mathbf{A}) \le 1$, $\Lambda_{\max}(\mathbf{A}) \ge 1$, and all of the rest eigenvalues equal to one. Further using the fact that $\mathrm{Tr}(\mathbf{A}) = p$ and $\det(\mathbf{A}) = -(p-2)$ we conclude $\Lambda_{\min}(\mathbf{A}) = 1 - \sqrt{p-1}$ and $\Lambda_{\max}(\mathbf{A}) = 1 + \sqrt{p-1}$. Based on the eigenvalues we set $\alpha = c/\sqrt{p-1}$ with constant $c \in (0,1)$, which gives $\Lambda_{\min}(\mathbf{\Sigma}^{(k)}) = 1 - c$ and $\Lambda_{\max}(\mathbf{\Sigma}^{(k)}) = 1 + c$.

We then compute C_Σ under the above setting. Using the formula for the inverse of arrowhead matrices provided in Salkuyeh and Beik (2018), we obtain

$$\left(\mathbf{\Sigma}^{(k)} - \mathbf{\Sigma}^{(0)}\right)\left(\sum_{1\le k\le K}\frac{1}{K}\mathbf{\Sigma}^{(k)}\right)^{-1} = \left(\begin{array}{c|c} 1-\beta^{-1} & \alpha\beta^{-1}\cdots\alpha\beta^{-1} \\ \hline \alpha\beta^{-1} & \\ \vdots & -\alpha^2\beta^{-1} \\ \alpha\beta^{-1} & \end{array}\right)_{p\times p}, \tag{53}$$

where $\beta := 1 - (p-1)\alpha^2$. Substituting $\alpha = c/\sqrt{p-1}$ reveals $C_\Sigma = O(\sqrt{p})$.

D Implementation of TransFusion

In this section, we show that how the first step of TransFusion method can be implemented using the Proximal Gradient Descent (PGD) algorithm via a change of variables. The second de-bias step (5) is a standard LASSO problem and there is a rich literature on efficient numerical solutions, see Li et al. (2018) and the references therein.

For the first co-traning step, notice that under the one-to-one variable transformation $\boldsymbol{\theta} := ((\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(0)})^\top, (\boldsymbol{\beta}^{(2)} - \boldsymbol{\beta}^{(0)})^\top, \ldots, (\boldsymbol{\beta}^{(K)} - \boldsymbol{\beta}^{(0)})^\top, \boldsymbol{\beta}^{(0)})$, solving problem (2) is equivalent to solving

$$\hat{\boldsymbol{\theta}} \in \operatorname*{argmin}_{\boldsymbol{\theta}} \left\{ \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda_0 \sum_{k=0}^{K} a_k \left\|\boldsymbol{\theta}^{(k)}\right\|_1 \right\}, \tag{54}$$

where $a_0 = 1$ and $\mathbf{y}$ and $\mathbf{X}$ are defined in (16), recalled below for convenience:

$$\mathbf{y} := \begin{pmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \\ \vdots \\ \mathbf{y}^{(K)} \\ \mathbf{y}^{(0)} \end{pmatrix} \quad \mathbf{X} := \begin{pmatrix} \mathbf{X}^{(1)} & 0 & \cdots & 0 & \mathbf{X}^{(1)} \\ 0 & \mathbf{X}^{(2)} & \cdots & 0 & \mathbf{X}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathbf{X}^{(K)} & \mathbf{X}^{(K)} \\ 0 & 0 & \cdots & 0 & \mathbf{X}^{(0)} \end{pmatrix}.$$

Problem (54) is a weighted LASSO problem (Zou, 2006), and thus the proximal gradient descent algorithm can be directly applied. Given initialization $\boldsymbol{\theta}_0 \in \mathbb{R}^{(K+1)p}$ and proximal parameter $\gamma > 0$, the PGD iteration reads:

$$\boldsymbol{\theta}_{t+1} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^{(K+1)p}} \left\langle \frac{1}{N}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\theta}_t - \mathbf{y}), \boldsymbol{\theta} - \boldsymbol{\theta}_t \right\rangle + \frac{1}{2\gamma}\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2 + \lambda_0 \sum_{k=0}^{K} a_k \left\|\boldsymbol{\theta}^{(k)}\right\|_1. \tag{55}$$

Notably, although calculating the gradient $\frac{1}{N}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\theta}_t - \mathbf{y})$ appears to involve multiplying a $(K+1)p \times (K+1)p$ matrix by a $(K+1)p$-dimensional vector, using the sparse structure of $\mathbf{X}$ it is easy to see it can be obtained via computing quantities $\mathbf{X}^{(k)\top}\mathbf{X}^{(k)}\boldsymbol{\beta}_t^{(k)}$ and $\mathbf{X}^{(k)\top}\mathbf{y}^{(k)}$ for $k = 0, \ldots, K$. Therefore per iteration it only involves the multiplication of a $p \times p$ matrix by a p-dimensional vector, and can be computed efficiently in parallel. The proximal mapping (55) can also be computed in closed form via soft-thresholding.

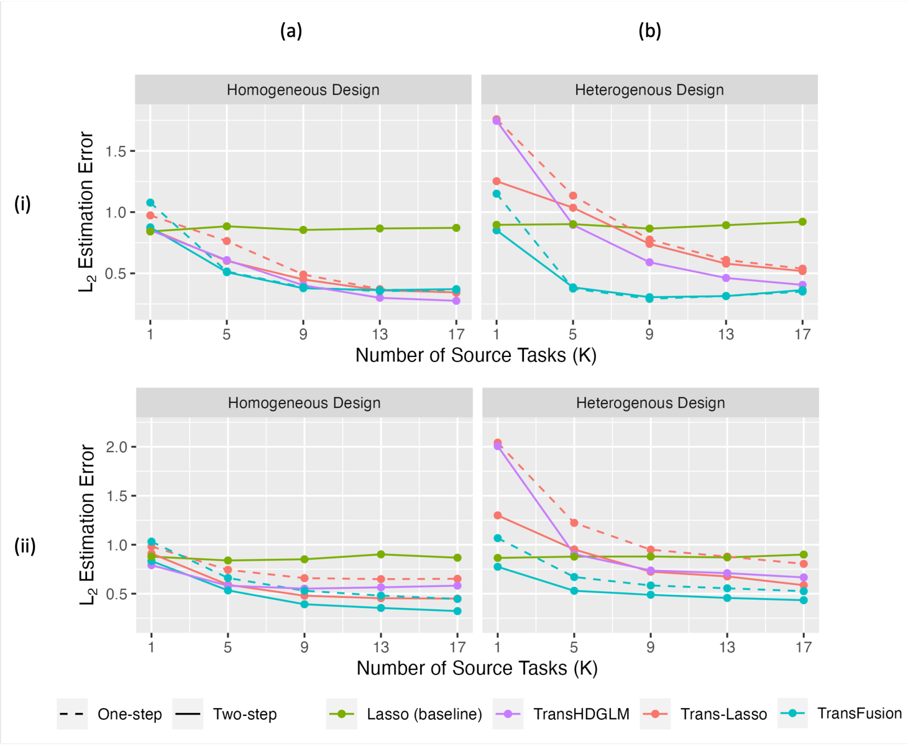

Figure 3: Comparison of estimation errors under (i) diverse and (ii) non-diverse source task settings with (a) homogeneous design and (b) heterogeneous design with a large choice of K.

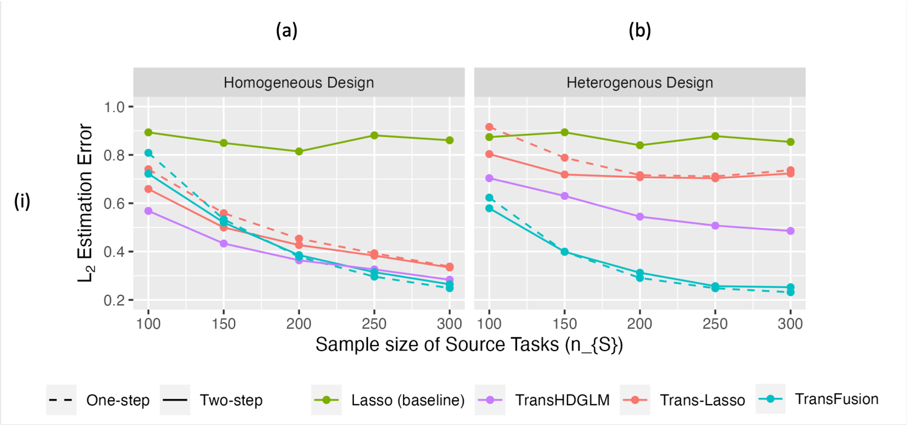

Figure 4: Comparison of estimation errors under (i) diverse source task settings with (a) homogeneous design and (b) heterogeneous design with different choice of n_S and a fixed $K = 10$.

E Additional Simulation Results

Fig. 3 shows the results with the number of source tasks $K \in \{1, 5, 9, 13, 17\}$. The results are in general analogous to those in section 4. One interesting observation is that in the diverse source task setting (i), as the source task number K increases and the source task sample size n_S remains a constant, Fig. 3 (i-a) and (i-b) show diminishing improvement of TransFusion estimation error. This is because as our theoretical results suggest, in such settings, TransFusion achieves an estimation error of order $O(\frac{s \log p}{n_T + K n_S} + (1 + v_n)\bar{h}\sqrt{\frac{\log p}{n_T}})$ with $v_n = \frac{K^2 \log p}{n_S}\bar{h}$. If we fix the source sample size n_S, increasing K only decreases the first term, and the overall sum will be dominated by the second term. The term v_n contributes to the U-shape error curve in the figures. This is the cost of using only local data to estimate the task-specific signal $\boldsymbol{\delta}^{(k)}$ in order to achieve robustness to covariate shifts.

Meanwhile, the theory implies that TransFusion would have a consistent error improvement if we proportionally increase the source sample size, n_S, with K. This is verified in Fig. 4, where, in the same diverse source task setting, a fixed K with a growing n_S results in a faster error reduction for TransFusion compared to other methods, especially in the existence of covariate shifts (Fig. 4 (i-b)).

F Case Study: Handwritten-digit Classification

We consider the problem of handwritten-digit classification based on the MNIST-C (Mu and Gilmer, 2019) dataset. MNIST-C is a comprehensive suite of different corruptions applied to the MNIST dataset, for benchmarking out-of-distribution robustness in computer vision. This setup allows us to evaluate the covariate-shift robustness of the proposed *TransFusion* algorithm.

We choose images corrupted by “brightness”, “fog” and “motion blur” as the source datasets ($K = 3$), drawing n_S source samples from each with $n_S \in \{500, 1000, 1500, 2000\}$. Then we set the original MNIST dataset as the target dataset, from which we collect $n_T = 100$ target samples. We use flattened pixel features of the images as features, amounting to $28 \times 28 = 784$ features per image ($p = 784$). We transform the classification problem into 10 binary classification problems—one for each digit against all others, then evaluate the classification accuracy on 2000 test images from the target dataset.

For this multi-source binary classification task, we employ the *TransFusion* algorithm and compare its performance with the *TransLasso* algorithm and *Lasso (baseline)* algorithm, all based on the logistic regression. Note that although in this paper we mainly focus on the linear regression setting, our theory and methodology can be easily generalized to the logistic regression setting (Friedman et al., 2010; Negahban et al., 2012). The implementation follows a similar manner as discussed in Appendix D.

One of the key challenges in this classification problem is managing the covariate shifts between different tasks. This was evident from the correlation heat maps of flattened pixel features in Figure 5. Compared to the target sample, each source sample exhibits a distinct covariate correlation structure. Such covariate shifts should be carefully handled for effective knowledge transfer.

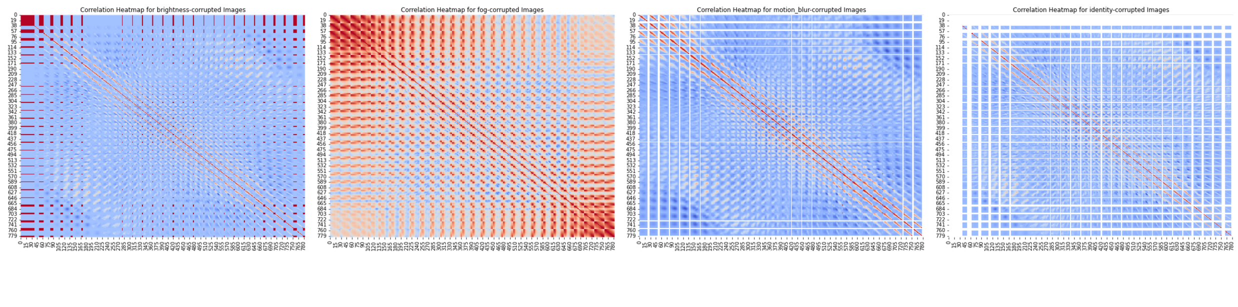

Figure 5: The correlation heatmaps of flattened pixel features for handwritten digit images affected by different types of corruptions. From left to right, the images are subjected to brightness corruption, fog corruption, motion blur corruption, and the original images without corruption (identity corruption).

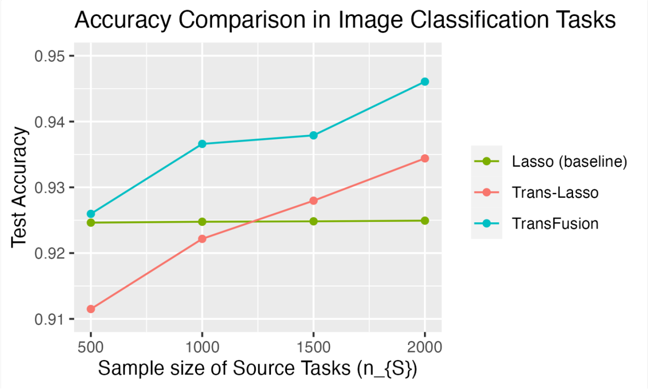

Figure 6: Test accuracy of different methods for the binary digit classification problem averaged over 10 problems, plotted against varying source sample sizes n_S.

Figure 6 shows the average test accuracy across the 10 binary classification problems versus the source sample size n_S. From the figure, we can see that *TransFusion* consistently outperforms other benchmarks. When the source sample size is relatively small, the *Trans-Lasso* method even performs worse than the baseline Lasso method, suggesting that it fails to utilize the transferable knowledge under the covariate shift. In contrast, *TransFusion* method still outperforms the baseline, indicating its robustness against the covariate shift.

G Choice of $\tilde{\boldsymbol{\beta}}^{(k)}$

In this section, we specify how to choose $\tilde{\boldsymbol{\beta}}^{(k)}$. An intuitive option is the LASSO estimator computed based on source sample $(\mathbf{X}^{(k)}, \mathbf{y}^{(k)})$:

$$\hat{\boldsymbol{\beta}}^{(k)}_{\text{LASSO}} \in \underset{\boldsymbol{\beta}\in\mathbb{R}^{(k+1)p}}{\operatorname{argmin}} \left\{ \frac{1}{2n_S} \left\| \mathbf{y}^{(k)} - \mathbf{X}^{(k)}\boldsymbol{\beta} \right\|_2^2 + \tilde{\lambda}_k \|\boldsymbol{\beta}\|_1 \right\}. \tag{56}$$

However, since the LASSO estimator is biased, computing $\hat{\mathbf{w}}_C$ by aggregating the local $\hat{\boldsymbol{\beta}}^{(k)}_C$s can only reduce the variance and has almost no effects on the bias (Mcdonald et al., 2009). To overcome such a drawback, we propose to first "correct" the bias at the local level before transmitting it to the target node for transfer learning. This is achieved by debiasing $\hat{\boldsymbol{\beta}}^{(k)}_{\text{LASSO}}$ using the method proposed in (Javanmard and Montanari, 2014):

$$\tilde{\boldsymbol{\beta}}^{(k)} = \hat{\boldsymbol{\beta}}^{(k)}_{\text{LASSO}} + \frac{1}{n_S}\hat{\boldsymbol{\Theta}}^{(k)} \left(\mathbf{X}^{(k)}\right)^\top \left(\mathbf{y}^{(k)} - \mathbf{X}^{(k)}\hat{\boldsymbol{\beta}}^{(k)}_{\text{LASSO}}\right). \tag{57}$$

Here, $\hat{\boldsymbol{\Theta}}^{(k)}$ serves as an approximation of $\left(\boldsymbol{\Sigma}^{(k)}\right)^{-1}$, whose j-th row is defined as the solution of the following optimization problem

$$\begin{aligned} \underset{\boldsymbol{\theta}_j\in\mathbb{R}^p}{\text{minimize}} \quad & \boldsymbol{\theta}_j^\top \hat{\boldsymbol{\Sigma}}^{(k)} \boldsymbol{\theta}_j \\ \text{subject to} \quad & \left\| \hat{\boldsymbol{\Sigma}}^{(k)}\boldsymbol{\theta}_j - \boldsymbol{e}_j \right\|_\infty \le \mu_k, \end{aligned} \tag{58}$$

with parameter $\mu_k > 0$ properly chosen. The source code for solving the problem can be found at `https://web.stanford.edu/~montanar/sslasso/code.html`.

To understand the choice of $\tilde{\boldsymbol{\beta}}$, we may rewrite (57) by subtracting $\boldsymbol{\beta}^{(k)}$ from both sides to obtain

$$\tilde{\boldsymbol{\beta}}^{(k)} - \boldsymbol{\beta}^{(k)} = \frac{1}{n_S}\hat{\boldsymbol{\Theta}}^{(k)} \left(\mathbf{X}^{(k)}\right)^\top \boldsymbol{\epsilon}^{(k)} \quad - \left(\hat{\boldsymbol{\Theta}}^{(k)}\hat{\boldsymbol{\Sigma}}^{(k)} - \boldsymbol{I}\right)\left(\hat{\boldsymbol{\beta}}^{(k)}_{\text{LASSO}} - \boldsymbol{\beta}^{(k)}\right). \tag{59}$$

The first term on the right-hand side of the equation is associated with the variance of $\tilde{\boldsymbol{\beta}}^{(k)}$. Through (58), we effectively minimize this variance. The second term, $\boldsymbol{b}^{(k)}$, on the other hand, contributes to the bias. By selecting an appropriate constraint parameter μ_k on $\|\hat{\boldsymbol{\Sigma}}^{(k)}\boldsymbol{\theta}_j - \boldsymbol{e}_j\|_\infty$ in (58), we can control the bias term to be comparable or even smaller than the variance term, thereby mitigate the impact of bias on the later aggregation step. Hence, this choice of $\tilde{\boldsymbol{\beta}}$ guarantees that the D-TransFusion could achieve much less communication overhead, while at the same time achieving the minimum performance loss compared to centralized TransFusion.