MULTI-TASK LEARNING FOR HETEROGENEOUS MULTI-SOURCE BLOCK-WISE MISSING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-task learning (MTL) has emerged as an imperative machine learning tool to solve multiple learning tasks simultaneously and has been successfully applied to healthcare, marketing, and biomedical fields. However, in order to borrow information across different tasks effectively, it is essential to utilize both homogeneous and heterogeneous information. Among the extensive literature on MTL, various forms of heterogeneity are presented in MTL problems, such as block-wise, distribution, and posterior heterogeneity. Existing methods, however, struggle to tackle these forms of heterogeneity simultaneously in a unified framework. In this paper, we propose a two-step learning strategy for MTL which addresses the aforementioned heterogeneity. First, we impute the missing blocks using shared representations extracted from homogeneous source across different tasks. Next, we disentangle the mappings between input features and responses into a shared component and a task-specific component, respectively, thereby enabling information borrowing through the shared component. Our numerical experiments and real-data analysis from the ADNI database demonstrate the superior MTL performance of the proposed method compared to a single task learning and other competing methods.

1 INTRODUCTION

Motivation. Many datasets for specific scientific tasks lack sufficient samples to train an accurate machine learning model. In recent decades, multi-task learning (MTL) has become a powerful tool to borrow information across related tasks for improved learning capacity. In addition, data collected for each task might come from multiple sources; for example, clinic notes, medical images, and lab tests are collected for medical diagnosis. The multi-source data brings richer information for each task, potentially enhancing the MTL. However, this also imposes several key challenges. First of all, it is common that observed data sources for each task are heterogeneous, so some blocks (certain data sources for certain tasks) could be entirely missing, termed as a block-wise missing structure in the literature. Second, even if the observed data sources are aligned across tasks, the distribution of the same data source could be heterogeneous, referred to as distribution heterogeneity. Furthermore, the associations between features and responses could vary due to distinct scientific goals or other factors, which we refer to as posterior heterogeneity. In the following, we provide concrete motivating examples to illustrate these challenges in different problems.

Example 1: Medical multi-source datasets. Multi-source data are widely observed in medical applications and offer more comprehensive information than single-source data. For example, the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset includes medical imaging, biosamples, gene expression, and demographic information (Mueller et al., 2005a;b). However, entire blocks of data are often missing when certain sources become unnecessary or infeasible to collect due to known factors or patient conditions (Madden et al., 2016).

Example 2: Single-cell multi-omics datasets. Data from different experimental batches often exhibit distribution heterogeneity across various omics measurements. For instance, transcriptome data collected from different batches can display varying patterns due to differences in experimental conditions or technical variability (Cao et al., 2022a;b). In multi-omics datasets, sequencing data distributions also differ across various cancer types (Subramanian et al., 2020).

Example 3: Combining randomized controlled trials (RCTs) and observational data. Combining RCTs and observational data has become effective for deriving causal effects due to the high costs and limited participant numbers in RCTs (Colnet et al., 2024). However, RCTs and observational data often exhibit posterior heterogeneity (Li et al., 2024a); for instance, causal effects in RCTs may differ from associations in observational data due to the controlled conditions of RCTs (Imbens & Rubin, 2015).

Challenges. The challenge in MTL is to incorporate various forms of heterogeneity, each introducing a unique challenge. Block-wise heterogeneity complicates the integration of data as missing patterns vary across tasks, making it difficult to leverage shared information efficiently. For example, in the ADNI dataset, imaging features are present in all datasets, but genetic information is available only in specific subsets (Xue & Qu, 2021). In addition, distribution heterogeneity can also lead to biased or misleading scientific conclusions if not addressed properly. For instance, in multi-omics datasets, sequencing data vary significantly across different cancer types (Subramanian et al., 2020). Lastly, posterior heterogeneity affects the accuracy of predictions. For example, the relationships identified in RCTs often do not align with those observational data collected in real-life settings (Kent et al., 2018; 2020). While each type of heterogeneity imposes its own challenge, addressing all three challenges simultaneously under a unified framework presents significant obstacles, and to our best knowledge, current MTL methods are not equipped to handle these intricate dilemmas.

Contributions. In this work, we propose a unified MTL framework to address three types of heterogeneity in MTL. There are three key contributions: First, we propose a novel block-wise missing imputation method which effectively handles distribution heterogeneity by learning both shared and task-specific representations, uncovering complex structures between sources, and enabling better generalization during imputation. Second, we disentangle the associations between all input features and responses into shared and task-specific components, allowing for the effective integration of information while adapting to differences across tasks. Third, we propose an MTL architecture consisting of two parts to construct these associations. The first part builds heterogeneous feature spaces, while the second part learns responses, jointly addressing both distribution and posterior heterogeneity. We validate the proposed framework on synthetic and real-world datasets, demonstrating its superior performance in handling block-wise missing data and various levels of heterogeneity.

2 Related Work

Multi-source data integration. Several related works on multi-source data collected for the same set of samples fall within the Joint and Individual Variation Explained (JIVE) framework. These methods are classified as unsupervised or supervised JIVE, depending on the presence of responses. Unsupervised JIVE and its variants learn joint, individual, and partially shared structures from multiple data matrices through low-rank approximations (Lock et al., 2013; Feng et al., 2018; Gaynanova & Li, 2019; Choi & Jung, 2022; Yi et al., 2023; James et al., 2024). Supervised JIVE, on the other hand, focuses on regression for multi-source data (Gao et al., 2021; Palzer et al., 2022; Zhang & Gaynanova, 2022; Wang & Lock, 2024). Similarly, factor models have been applied to multi-source data in a supervised setting (Shu et al., 2020; Li & Li, 2022; Anceschi et al., 2024). While these methods can effectively address distribution heterogeneity across different sources in linear settings, they are limited in scope, as they capture only simple data structures within a single task.

Multi-source block-wise missing data integration. Recently, several methods have been developed to address block-wise missing data. These methods can be divided into two categories based on whether imputation is involved. Imputation-based methods assume consistent correlations between different sources across datasets, allowing for the imputation of missing blocks (Gao & Lee, 2017; Le Morvan et al., 2021; Xue & Qu, 2021; Xue et al., 2021; Zhou et al., 2021; Ouyang et al., 2024). For example, Xue & Qu (2021) and Xue et al. (2021) construct estimating equations using all available information and integrate informative estimating functions to achieve efficient estimators. On the other hand, non-imputation-based methods focus on learning the covariance matrices among predictors and between the response and predictors from the observed blocks (Yuan et al., 2012; Xiang et al., 2014; Yu et al., 2020; Li et al., 2024b). While these methods perform well in the absence of distribution shift and posterior shift, effectively utilizing all block-wise missing data, they struggle to handle distribution or posterior heterogeneity.

Multi-task learning (MTL). There is a growing literature on learning multiple tasks simultaneously with a shared model; see Zhang & Yang (2018); Crawshaw (2020); Zhang & Yang (2021) for reviews. Here, we primarily focus on MTL with deep neural networks, as these networks can capture more complex relationships. These methods can be broadly classified into four categories: The first category is balancing individual loss functions for different tasks, which is a common approach to ease multi-task optimization (Du et al., 2018; Gong et al., 2019; Hang et al., 2023; Wu et al., 2024). The second category involves regularization, especially in the form of hard parameter sharing (Subramanian et al., 2018; Liu et al., 2019; Maziarka et al., 2022) and soft parameter sharing (Ullrich et al., 2017; Lee et al., 2018; Han et al., 2024). The third category addresses the challenge of negative transfer, where explicit gradient modulation is used to alleviate conflicts in learning dynamics between tasks (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018; Maninis et al., 2019; Abdollahzadeh et al., 2021; Hu et al., 2022; Wang et al., 2024b). The fourth category uses knowledge distillation to transfer knowledge from single-task networks to a multi-task student network (Rusu et al., 2015; Teh et al., 2017; Clark et al., 2019; D'Eramo et al., 2024). Although MTL can integrate data from multiple tasks, it is limited in addressing different types of heterogeneity and is constrained by the assumption of a fully observed setting.

Most related work focuses on addressing a single challenge, such as posterior heterogeneity or the missing data problem, but typically fails to address all challenges simultaneously. In contrast, our proposed method extends these approaches by tackling both distribution and posterior heterogeneity in a block-wise missing setting. This enables a more comprehensive integration of data across tasks, resulting in improved performance in MTL.

3 Two-step MTL for heterogeneous multi-source block-wise missing data

Notation. We introduce the notations used in this paper. Vectors and matrices are denoted by $\boldsymbol{x}$ and $\boldsymbol{X}$, respectively. The ℓ_1 and ℓ_2 norms of a vector $\boldsymbol{x}$ are $\|\boldsymbol{x}\|_1$ and $\|\boldsymbol{x}\|_2$, and the Frobenius norm of a matrix $\boldsymbol{X}$ is $\|\boldsymbol{X}\|_F$. The symbol $|$ represents concatenation. For example, $[\boldsymbol{x}_1|\boldsymbol{x}_2]$ denotes concatenating a $p_1 \times 1$ vector $\boldsymbol{x}_1$ and a $p_2 \times 1$ vector $\boldsymbol{x}_2$ into a $(p_1 + p_2) \times 1$ vector. Similarly, $[\boldsymbol{X}_1|\boldsymbol{X}_2]$ denotes concatenating an $n \times p_1$ matrix $\boldsymbol{X}_1$ and an $n \times p_2$ matrix $\boldsymbol{X}_2$ into an $n \times (p_1 + p_2)$ matrix. We define $[r] = \{1, 2, \ldots, r\}$ as the set of integers from 1 to r.

Problem Description. Suppose we have data from T tasks, with features collected from $T + 1$ sources. For all tasks, we assume that a common source, called the anchoring source, is observed. Additionally, each task has its own task-specific source, denoted as $\boldsymbol{x}_s^t$ for the s-th source in the t-th task. Specifically, $\boldsymbol{x}_0^t$ represents the anchoring source observed in the t-th task, and $\boldsymbol{x}_t^t$ denotes the task-specific source for the t-th task, while $\{\boldsymbol{x}_s^t\}_{s \neq 0,t}$ are missing. For the t-th task, we observe n_t samples $\{[\boldsymbol{x}_{0,i}^t|\boldsymbol{x}_{t,i}^t], y_i^t\}_{i=1}^{n_t}$. This block-wise missing pattern is common in real-world applications. For example, in biomedical data, some measurements (data sources) are widely observed for all subjects, while some measurements are only collected to a subgroup of subjects due to various reasons. Concretely, in the ADNI data that we analyzed in Section 4.3, MRI is crucial to monitor the cognitive impairment development of Alzheimer's patients, so it is measured for all subjects, while gene expression and PET images are less crucial and are only observed for two subgroups separately. Another example is the split questionnaire design, which aims to reduce respondent fatigue and improve response rates by assigning different subsets of the questionnaire to different sampled respondents (Lin et al., 2023). In Figure 1, we provide an example of a block-wise missing pattern for 4 tasks and 5 sources, where the blue source $\boldsymbol{x}_0^t$ for $t \in [4]$ represents the anchoring source observed by all four tasks, and each task also has a uniquely observed specific source $\boldsymbol{x}_t^t$ for $t \in [4]$. Our goal is to perform MTL on these tasks with block-wise missing data.

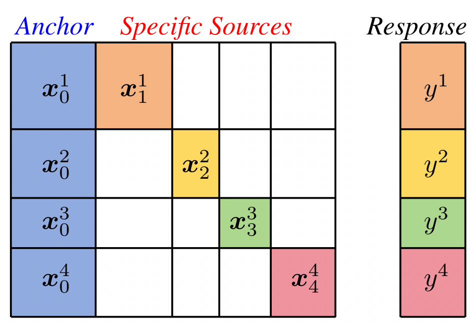


Figure 1: Block-wise missing pattern for 4 tasks and 5 sources, including an anchoring source and task-specific sources.

Figure 1 illustrates one of the challenges in MTL. Each task has different missing blocks; for example, in the first task, sources 2, 3, and 4 are missing, while in the second task, sources 1, 3, and 4 are missing. Furthermore, both distribution and posterior heterogeneity across tasks complicate the application of standard imputation methods (Nair et al., 2019; He et al., 2024a;b) and MTL methods (Kouw & Loog, 2018; Lee et al., 2024; Maity et al., 2024).

3.1 Heterogeneous Block-wise Imputation

In this section, we propose the first step, Heterogeneous Block-wise Imputation (HBI) for imputing the missing blocks while leveraging distribution heterogeneity across tasks. HBI extracts disentangled hidden representations from the anchoring source $\boldsymbol{x}_0$, including a shared representation across tasks and a task-specific representation for each task. The shared representation is then used to impute the missing blocks, improving generalization across tasks.

For T tasks and $T+1$ sources, we impute the task-specific sources in a parallel fashion. For each task-specific source $s \neq 0$, we utilize the anchoring source across all tasks and $\boldsymbol{x}_s^s$ to impute the unobserved blocks $\{\boldsymbol{x}_s^t\}_{t\neq s}$. In particular, for the t-th source, only the t-th task has observed values for the features $\boldsymbol{x}_t^t$. The imputation aims to use the observed $\boldsymbol{x}_0^t$ and $\boldsymbol{x}_t^t$ along with $\boldsymbol{x}_0^{-t} = \{\boldsymbol{x}_0^r\}_{r\neq t}$ to estimate the missing features in the t-th source for the other $T-1$ tasks, where $\boldsymbol{x}_t^{-t} = \{\boldsymbol{x}_t^r\}_{r\neq t}$ are unobserved. For example, in Figure 2, we use information from $\boldsymbol{x}_0^1$, $\boldsymbol{x}_1^1$, and $\boldsymbol{x}_0^{-1} = \{\boldsymbol{x}_0^2, \boldsymbol{x}_0^3, \boldsymbol{x}_0^4\}$ to impute the missing blocks $\boldsymbol{x}_1^{-1} = \{\boldsymbol{x}_1^2, \boldsymbol{x}_1^3, \boldsymbol{x}_1^4\}$ for the task 1-specific source.

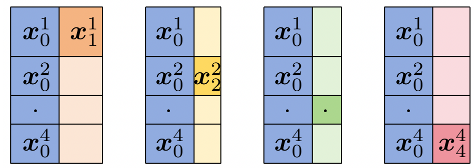


Figure 2: Illustration of parallel imputation for task-specific sources.

This is accomplished by learning a model that exploits both the shared and task-specific information of the data, allowing for accurate prediction of missing values based on the available observed data. To fully integrate multi-source information, we leverage an encoder-decoder framework, which is well-suited for capturing non-linear relationships in data. Let $E_c(\cdot)$ be a common encoder that maps $\{\boldsymbol{x}_0^t, \boldsymbol{x}_0^{-t}\}$ to shared representations $\boldsymbol{f}_c^t = E_c(\boldsymbol{x}_0^t)$ and $\boldsymbol{f}_c^{-t} = E_c(\boldsymbol{x}_0^{-t})$ across all T tasks. Let $E_p^t(\cdot)$ and $E_p^{-t}(\cdot)$ be task-specific encoders that map $\boldsymbol{x}_0^t$ and $\boldsymbol{x}_0^{-t}$ to task-specific representations $\boldsymbol{g}^t = E_p^t(\boldsymbol{x}_0^t)$ and $\boldsymbol{g}^{-t} = E_p^{-t}(\boldsymbol{x}_0^{-t})$. Then, $D(\boldsymbol{f}, \boldsymbol{g})$ serves as a decoder that reconstructs the anchoring source $\boldsymbol{x}_0$ from $\boldsymbol{f}$ and $\boldsymbol{g}$. Finally, $G(\boldsymbol{f})$ is a predictor that maps the shared representation $\boldsymbol{f}$ to the task t-specific source $\boldsymbol{x}_t$. The resulting heterogeneous block-wise imputation model is illustrated in Figure 3.

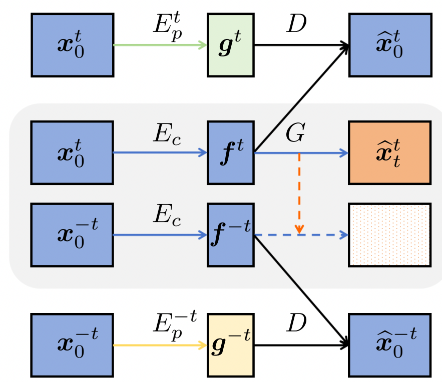


Figure 3: Illustration of HBI for the task t-specific source $\boldsymbol{x}_t$. A common encoder $E_c(\cdot)$ learns to capture representation components that are shared among tasks. Task-specific encoders $E_p(\cdot)$ (one for the t-th task, and one for the other $T-1$ tasks) learn to capture task-specific components of the representations. A decoder learns to reconstruct the anchoring source $\boldsymbol{x}_0$ by using both shared and task-specific representations. The shared part of the relationship between the anchoring source $\boldsymbol{x}_0$ and the task t-specific source $\boldsymbol{x}_t$ can be borrowed through $E_c(\cdot)$ and $G(\cdot)$ for imputation. See the text for more information.

In Figure 3, we assume that the relationship between the anchoring source $\boldsymbol{x}_0$ and the task t-specific source $\boldsymbol{x}_t$ can be borrowed through the shared representations $\boldsymbol{f}$, the common encoder $E_c(\cdot)$, and the decoder $G(\cdot)$. This allows us to utilize the shared information (reflected in $\boldsymbol{f}^t$ and $\boldsymbol{f}^{-t}$) for imputation, while also accounting for the heterogeneity between $\boldsymbol{x}_0^t$ and $\boldsymbol{x}_0^{-t}$ (reflected in $\boldsymbol{g}^t$ and $\boldsymbol{g}^{-t}$). Existing imputation methods often learn the relationship between $\boldsymbol{x}_0$ and $\boldsymbol{x}_t$ within the t-th task and apply the relationship to other tasks, overlooking distribution heterogeneity (Xue et al., 2021; Zhou et al., 2021). Moreover, common imputation methods rely on parametric models which fail to capture complex relationships in missing data (Xue & Qu, 2021; Li et al., 2023). However, our HBI method effectively overcomes these obstacles. Notably, HBI's extraction of the common components in the relationships between sources across different tasks shares similarities with domain adaptation (Mansour et al., 2008; Bousmalis et al., 2016; Tzeng et al., 2017; Farahani et al., 2021) but focuses on completing block-wise missing data. This architecture effectively models complex data structures and interactions, providing a robust tool for understanding intricate patterns. The resulting optimization can be formulated as:

$$(\widehat{E}_c(\cdot), \widehat{E}_p^t(\cdot), \widehat{E}_p^{-t}(\cdot), \widehat{D}(\cdot), \widehat{G}(\cdot)) = \arg\min\{\mathcal{L}_{\text{pre}} + \mathcal{L}_{\text{recon}}\}, \tag{1}$$

In (1), the prediction loss $\mathcal{L}_{\text{pre}}$ trains the model to predict $\boldsymbol{x}_t^t$, the target of interest, which is applied only to the t-th task. We use the following loss function:

$$\mathcal{L}_{\text{pre}} = \sum_{i=1}^{n_t} l(\boldsymbol{x}_{t,i}^t, G(E_c(\boldsymbol{x}_{0,i}^t))),$$

where $\boldsymbol{x}_i$ denotes the observed sample, and $l(\cdot,\cdot)$ can be the mean squared error for continuous outcomes or cross-entropy for binary outcomes (this applies similarly to the following symbols). For the reconstruction loss in (1),

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{n_t} l(\boldsymbol{x}_{0,i}^t, D(E_c(\boldsymbol{x}_{0,i}^t), E_p^t(\boldsymbol{x}_{0,i}^t))) + \sum_{i=1}^{n_{-t}} l(\boldsymbol{x}_{0,i}^{-r}, D(E_c(\boldsymbol{x}_{0,i}^{-t}), E_p^{-t}(\boldsymbol{x}_{0,i}^{-t}))),$$

where $n_{-t} = \sum_{r\neq t} n_r$. Then, we can train (1) to obtain the estimators $\widehat{E}_c(\cdot)$ and $\widehat{G}(\cdot)$. Consequently, we compute $\widehat{\boldsymbol{x}}_t^{-t} = \widehat{G}(\widehat{E}_c(\boldsymbol{x}_0^{-t}))$. Note that (1) is constructed based on task t-specific source imputation. Similarly, we can construct imputations for the other $T-1$ sources. When performing imputation for different sources using HBI, the learned hidden representations and corresponding generative functions adapt dynamically. This adaptation is crucial as it allows the model to accommodate the unique information of each source. The complete algorithm for parallel heterogeneous imputation is provided in Appendix A.4.

Our proposed HBI method ensures that the imputation model leverages common information across tasks while incorporating the heterogeneity of each task. By decomposing the latent space into shared and task-specific components, we gain a nuanced understanding of how input features from different sources interact, thereby enhancing imputation accuracy.

3.2 Heterogeneous Multi-task Learning

In this section, we propose our MTL framework to accommodate distribution and posterior heterogeneity given the imputed blocks from HBI. Similar to the disentangled representations for features, we also model the association between features and responses as two components: a shared function mapping and a task-specific function mapping. Specifically, for the t-th task, we assume that the relationship between the response y^t and the features $[\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\cdots|\boldsymbol{x}_T^t]$ is given by:

$$y^t = \psi_c([\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\cdots|\boldsymbol{x}_T^t]) + \psi_p^t([\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\cdots|\boldsymbol{x}_T^t]), \tag{2}$$

where $[\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\cdots|\boldsymbol{x}_T^t]$ influence y^t through a shared mapping $\psi_c(\cdot)$ and a task-specific mapping $\psi_p^t(\cdot)$. Equation 2 extends traditional meta-analysis, which often assumes a linear relationship in the t-th task as $y^t = [\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\cdots|\boldsymbol{x}_T^t]^\top \boldsymbol{\beta}^t + \varepsilon$, where $\boldsymbol{\beta}^t$ includes a common component $\boldsymbol{\mu}$ shared across all T tasks and a unique component $\boldsymbol{\alpha}^t$ for each task (Chen et al., 2021; Cai et al., 2022; Maity et al., 2022). Traditional meta-analysis is incapable of accommodating non-linear relationships or varying effects. In contrast, we propose a flexible framework which accommodates non-linearities and integrates task-specific information.

To construct the shared mapping $\psi_c(\cdot)$ and the task-specific mappings $\{\psi_p^t\}_{t=1}^T$ jointly, we consider an MTL architecture comprising two parts. The first part builds heterogeneous feature spaces, while the second part learns responses for all T tasks. Specifically, for the t-th task, following HBI in Section 3.1, we obtain samples with reconstructed features $\{(\boldsymbol{x}_{0,i}^t, \ldots, \widehat{\boldsymbol{x}}_{t-1,i}^t, \boldsymbol{x}_{t,i}^t, \widehat{\boldsymbol{x}}_{t+1,i}^t, \ldots, \widehat{\boldsymbol{x}}_{T,i}^t), y_i^t\}_{i=1}^{n_t}$. These features can then be integrated to capture both shared and task-specific representations, enabling the utilization of the combined data while addressing task-specific heterogeneity. During HBI, the components $\{\widehat{\boldsymbol{x}}_s^t\}_{s\neq 0,t}$ are primarily predicted using the anchoring source $\boldsymbol{x}_0^t$, indicating that $\boldsymbol{x}_0^t$ serves as a common basis. To prevent redundancy, we extract shared representations solely from the anchoring source $\boldsymbol{x}_0^t$. Specifically, we define

$$\boldsymbol{h}^t = \phi_c(\boldsymbol{x}_0^t), \tag{3}$$

where $\phi_c(\cdot)$ is a shared encoder that learns hidden information from the anchoring source for all tasks. Meanwhile, task heterogeneity is captured by extracting representations from all features, creating a framework in which shared representations provide a common foundation, while task-specific details can still be preserved. For the t-th task, we define:

$$\boldsymbol{k}^t = \phi_p^t([\boldsymbol{x}_0^t|\cdots|\widehat{\boldsymbol{x}}_{t-1}^t|\boldsymbol{x}_t^t|\widehat{\boldsymbol{x}}_{t+1}^t|\cdots|\widehat{\boldsymbol{x}}_T^t]), \tag{4}$$

where ϕ_p^t is a task-specific encoder which maps the unique information within the t-th task. In (3) and (4), the heterogeneous feature spaces are fully captured using all data information. The task-specific representations $\boldsymbol{k}$ capture complex interactions between different sources unique to each task, aided by HBI in Section 3.1. In practice, such interactions are crucial. For example, in the ADNI dataset, there are intricate relationships between images and gene expression. Equation 4 accounts for this heterogeneous information. However, previous work (Moon & Carbonell, 2017; Bica & van der Schaar, 2022) often oversimplifies these interactions by focusing only on task-specific sources, neglecting a wealth of shared information from other tasks.

For the second part, we consider a network architecture for learning responses in all T tasks, consisting of L layers with both shared and task-specific subspaces (Ruder et al., 2019; Curth & Van der Schaar, 2021; Bica & van der Schaar, 2022). For simplicity, in the t-th task, let $\bar{\boldsymbol{k}}_l^t$ and $\bar{\boldsymbol{h}}_l^t$ represent the inputs, and $\boldsymbol{k}_l^t$ and $\boldsymbol{h}_l^t$ the outputs of the l-th layer. For $l = 1$, set $\bar{\boldsymbol{k}}_1^t = [\boldsymbol{h}^t|\boldsymbol{k}^t]$ and $\bar{\boldsymbol{h}}_1^t = \bar{\boldsymbol{k}}_1^t$. For $l > 1$, the inputs to the $(l+1)$-th layer are given by $\bar{\boldsymbol{k}}_{l+1}^t = [\boldsymbol{h}_l^t|\boldsymbol{k}_l^t]$ and $\bar{\boldsymbol{h}}_l^t = [\boldsymbol{h}_l^t]$. Let $g^t(\cdot)$ be the association function in the t-th task, defined as $g^t([\boldsymbol{h}_L^t|\boldsymbol{k}_L^t]) = \psi_c([\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\cdots|\boldsymbol{x}_T^t]) + \psi_p^t([\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\cdots|\boldsymbol{x}_T^t])$, where $g^t(\cdot)$ is a linear function for continuous outcomes and a sigmoid function for binary ones.

Figure 4 illustrates the construction of the shared mapping $\psi_c(\cdot)$ and the task-specific mappings $\psi_p^1(\cdot)$ and $\psi_p^2(\cdot)$ for two tasks. For task 1, the input features consist of $[\boldsymbol{x}_0^1|\boldsymbol{x}_1^1|\widehat{\boldsymbol{x}}_2^1]$, where $\widehat{\boldsymbol{x}}_2^1$ represents the imputed source. In the first part, we build the heterogeneous feature space by extracting the shared representation $\boldsymbol{h}^1 = \phi_c(\boldsymbol{x}_0^1)$ and the task-specific representation $\boldsymbol{k}^1 = \phi_p^1([\boldsymbol{x}_0^1|\boldsymbol{x}_1^1|\widehat{\boldsymbol{x}}_2^1])$. Similarly, for task 2, we extract $\boldsymbol{h}^2 = \phi_c(\boldsymbol{x}_0^2)$ and $\boldsymbol{k}^2 = \phi_p^2([\boldsymbol{x}_0^2|\widehat{\boldsymbol{x}}_1^2|\boldsymbol{x}_2^2])$. Next, we utilize the pairs $\{\boldsymbol{h}^1, \boldsymbol{k}^1\}$ and $\{\boldsymbol{h}^2, \boldsymbol{k}^2\}$ to model the responses y^1 and y^2, respectively. In Figure 4, the blue mapping illustrates the shared mapping $\psi_c(\cdot)$, while the orange and yellow mappings represent the task-specific mappings $\psi_p^1(\cdot)$ and $\psi_p^2(\cdot)$ for tasks 1 and 2, respectively.

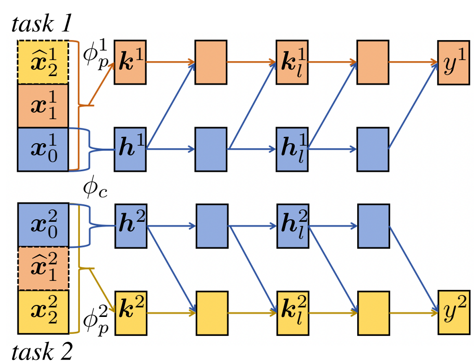


Figure 4: Illustration of the construction of shared mapping and task-specific mappings for two tasks.

The above construction allows us to define the following integrated loss across all T tasks:

$$\mathcal{L}_{\text{integ}} = \sum_{t=1}^{T}\sum_{i=1}^{n_t} l(y_i^t, g^t([\boldsymbol{h}_{L,i}^t|\boldsymbol{k}_{L,i}^t])). \tag{5}$$

Similar to Bousmalis et al. (2016), we also incorporate an orthogonality regularizer, defined as:

$$\mathcal{R}_{\text{orth}} = \sum_{t=1}^{T} \|(\boldsymbol{H}^t)^\top \boldsymbol{K}^t\|_F^2, \tag{6}$$

where $\boldsymbol{H}^t$ and $\boldsymbol{K}^t$ are matrices whose rows are the hidden representations $\boldsymbol{h}^t$ and $\boldsymbol{k}^t$, respectively. Furthermore, in (4), the input is $[\boldsymbol{x}_0^t \,|\, \cdots \,|\, \widehat{\boldsymbol{x}}_{t-1}^t \,|\, \boldsymbol{x}_t^t \,|\, \widehat{\boldsymbol{x}}_{t+1}^t \,|\, \cdots \,|\, \widehat{\boldsymbol{x}}_T^t]$, where $\boldsymbol{x}_0^t$ and $\boldsymbol{x}_t^t$ are the observed data, and $\{\widehat{\boldsymbol{x}}_s^t\}_{s\neq 0,t}$ are obtained through imputation. Since imputation can introduce errors, we also downweight the imputed data $\{\widehat{\boldsymbol{x}}_s^t\}_{s\neq 0,t}$ compared to observed data for learning $\boldsymbol{k}^t$ by applying a regularizer to the parameters of the first layer of the encoder $\phi_p^t(\cdot)$, defined as:

$$\mathcal{R}_{\text{imp}} = \sum_{t=1}^{T} \sum_{s\neq 0,t} \|\mathbf{\Theta}_{s,p,1}^t\|_F^2, \tag{7}$$

where $\mathbf{\Theta}_{s,p,1}^t$ are the parameters of the first layer of $\phi_p^t(\cdot)$ corresponding to $\{\widehat{\boldsymbol{x}}_s^t\}_{s\neq 0,t}$. This regularizer downweights potentially less accurate imputed features by penalizing the magnitude of the encoder parameters, fostering a model more robust to imputation errors. To further reduce redundancy between the shared and task-specific layers, we introduce an orthogonal regularizer (Ruder et al., 2019; Bica & van der Schaar, 2022). Let $d_{c,l-1}^t$ and $d_{p,l-1}^t$ be the dimensions of $\boldsymbol{h}_{l-1}^t$ and $\boldsymbol{k}_{l-1}^t$, the outputs of the $(l-1)$-th layer. Denote the weights in the l-th layer as $\mathbf{\Theta}_{c,l}^t \in \mathbb{R}^{d_{c,l-1}^t \times d_{c,l}^t}$ and $\mathbf{\Theta}_{p,l}^t \in \mathbb{R}^{(d_{c,l-1}^t + d_{p,l-1}^t)\times d_{p,l}^t}$. We apply the following regularizer:

$$\mathcal{R}_{\text{dr}} = \sum_{t=1}^{T} \sum_{l=1}^{L} \|(\mathbf{\Theta}_{c,l}^t)^\top \mathbf{\Theta}_{p,l,1:d_{c,l-1}^t}^t\|_F^2. \tag{8}$$

By combining the losses from (5), (6), (7), and (8), we train the set of all parameters $\mathbf{\Theta}$ using the following integrated loss function:

$$(\widehat{\psi}_c, \{\widehat{\psi}_p^t\}_{1\leq t\leq T}) = \arg\min_{\mathbf{\Theta}} \{\mathcal{L}_{\text{integ}} + \gamma\mathcal{R}_{\text{orth}} + \delta\mathcal{R}_{\text{imp}} + \kappa\mathcal{R}_{\text{dr}}\},$$

where γ, δ, and κ are weights controlling the balance among different terms. The more detailed algorithm for heterogeneous MTL is provided in Appendix A.4.

4 Experiments

In this section, we conduct extensive numerical experiments, including two-task MTL, multi-task MTL with more than two tasks, and an application to the ADNI real dataset. The numerical experiments demonstrate that our proposed two-step MTL method effectively aggregates information in the presence of block-wise, distribution, and posterior heterogeneity.

4.1 MTL for Two tasks

We address a common real-world scenario involving MTL with two tasks for illustration. The data generation process (DGP) is as follows:

DGP: For Task 1, the features are denoted as $\boldsymbol{x}^1 = [\boldsymbol{x}_0^1|\boldsymbol{x}_1^1|\boldsymbol{x}_2^1]$ and follow a Gaussian distribution with mean $\mathbf{0}$ and an exchangeable covariance matrix. The variance is fixed at 1, and the covariance structure is specified as $(\rho_1)^{0.01|i-j|}$. We randomly generate n_1 samples, with the third block $\boldsymbol{x}_2^1$ missing in Task 1. For Task 2, the features are denoted as $\boldsymbol{x}^2 = [\boldsymbol{x}_0^2|\boldsymbol{x}_1^2|\boldsymbol{x}_2^2]$, where $\boldsymbol{x}^2$ follows a Gaussian distribution with mean $\mathbf{0}$, variance 1, and covariance structure $(\rho_2)^{0.01|i-j|}$. In this task, we generate n_2 samples, with the second block $\boldsymbol{x}_1^2$ missing. The responses are defined as:

$$y^1 = \alpha \sum_{d=1}^{p} v_{c,d}(x_d^1)^2/p + (1-\alpha)\sum_{d=1}^{p} v_{1,d}x_d^1/p + \varepsilon_1,$$
$$y^2 = \alpha \sum_{d=1}^{p} v_{c,d}(x_d^2)^2/p + (1-\alpha)\sum_{d=1}^{p} v_{2,d}x_d^2/p + \varepsilon_2,$$

where $p = \sum_{s=0}^{2} p_s$, with p_s being the dimension of the s-th source, and the subscript d denotes the d-th element of a vector (this notation applies to subsequent symbols as well). The parameters v_c, v_1, and v_2 are sampled from $N(-10, 10^2)$, and the noise terms $\varepsilon_1 \sim N(0, \sigma_1^2)$ and $\varepsilon_2 \sim N(0, \sigma_2^2)$. The parameter α controls the level of sharing across tasks. Additionally, our DGP accounts for nonlinear relationships by element-wise square, further increasing the complexity of MTL. For evaluation, we calculate the average root-mean-squared error (RMSE) on the testing data, as defined in Appendix A.2. We conduct experiments under various settings to compare the proposed MTL for heterogeneous multi-source block-wise missing data (MTL-HMB) against existing methods, including

Single Task Learning (STL) and Transfer Learning for Heterogeneous Data (HTL) (Bica & van der Schaar, 2022).

Setting A: Effect of covariance parameters. We set $n_1 = n_2 = 300$, $p_0 = 100$, $p_1 = p_2 = 25$, $\alpha = 0.3$, and $\sigma_1 = \sigma_2 = 0.1$. To examine the impact of the covariance parameters ρ_1 and ρ_2, we set $\rho_1 = \rho_2$ and vary them from 0.5 to 0.95, assuming no distribution heterogeneity across datasets. As shown in Figure 5(a), increasing correlation improves prediction accuracy across all methods. The proposed MTL-HMB is the best performer. Specifically, at $\rho = 0.95$, it outperforms the others by more than $28.33\,\%$. Even at $\rho = 0.5$, despite imputation errors, our method maintains an advantage. This demonstrates that imputation enhances prediction, especially when distribution heterogeneity is absent.

Setting B: Effect of heterogeneous covariance parameters. We set $n_1 = n_2 = 300$, $p_0 = 100$, $p_1 = p_2 = 25$, $\alpha = 0.3$, and $\sigma_1 = \sigma_2 = 0.1$. To assess the impact of heterogeneous covariance, we fix $\rho_1 = 0.95$ and vary ρ_2 from 0.5 to 0.9. Smaller ρ_2 indicates greater heterogeneity and weaker correlations in Task 2, making predictions more challenging. Figure 5(b) shows that as ρ_2 increases, all methods improve, and our approach consistently leads. At the highest level of heterogeneity, MTL-HMB outperforms HTL by over $20.91\,\%$. Moreover, HTL shows no advantage over STL, indicating that transfer learning struggles with distribution heterogeneity. In contrast, the proposed method effectively solves the heterogeneity challenge through imputation, achieving better predictive accuracy.

Setting C: Effect of heterogeneous mappings. We set $n_1 = n_2 = 300$, $p_0 = 100$, $p_1 = p_2 = 25$, $\rho_1 = \rho_2 = 0.8$, and $\sigma_1 = \sigma_2 = 0.1$. The parameter α is varied to control the level of information sharing in the mappings to the response. A larger α indicates more shared information. With $\rho_1 = \rho_2$ fixed, heterogeneity is governed solely by α. Figure 5(c) shows that as α increases, the magnitude of y also increases, resulting in higher average RMSEs. Except at $\alpha = 0.1$, HTL consistently outperforms STL, indicating its advantage in incorporating posterior shift. Overall, the proposed MTL-HMB performs the best across all settings, even in the absence of distribution heterogeneity.

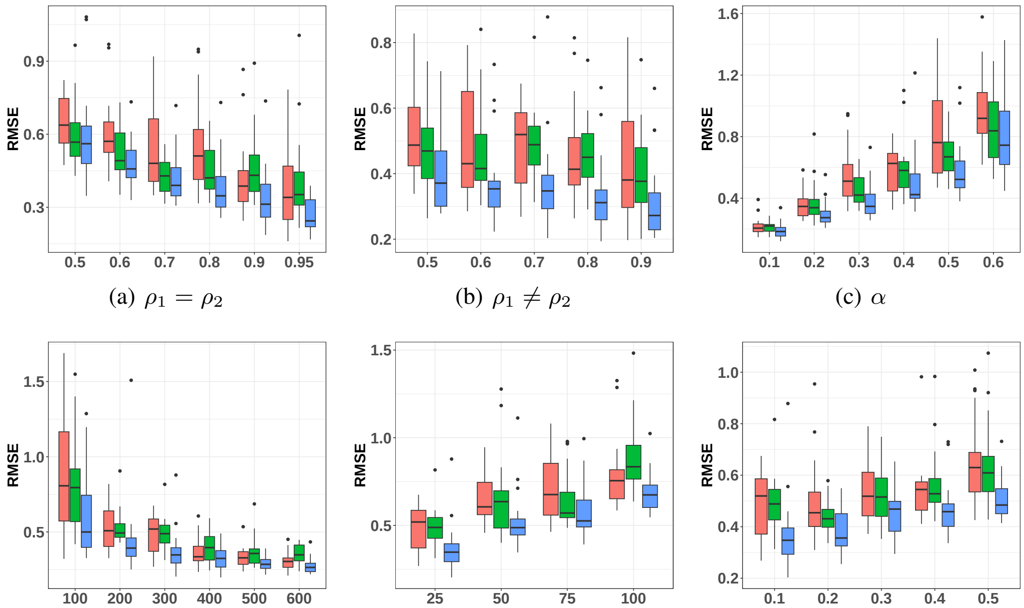
Setting D: Effect of sample sizes. We set $p_0 = 100$, $p_1 = p_2 = 25$, $\rho_1 = 0.95$, $\rho_2 = 0.7$, $\alpha = 0.3$, and $\sigma_1 = \sigma_2 = 0.1$. The sample sizes n_1 and n_2 vary as $n_1 = n_2 = k \times 100$ for $k = 1, \ldots, 6$. Figure 5(d) shows that as the sample size increases, average RMSEs decrease, and the corresponding variability of estimator is reduced across all methods. Our method consistently performs best, with an improvement of over $18.13\,\%$ compared to HTL. This is particularly notable at smaller sample sizes such as 100, where MTL-HMB outperforms HTL and STL by $37.13\,\%$ and $38.06\,\%$, respectively. Additionally, HTL does not significantly outperform STL, indicating difficulty in handling distribution heterogeneity.

Setting E: Effect of dimensions. We set $n_1 = n_2 = 300$, $\rho_1 = 0.95$, $\rho_2 = 0.7$, $\alpha = 0.3$, and $\sigma_1 = \sigma_2 = 0.1$. To assess the impact of dimensions p_1, p_2, and p_3, we fix $p_1 = 100$ and vary $p_2 = p_3 = k \times 25$ for $k = 1, \ldots, 4$. Figure 5(e) shows that increasing dimensions make prediction more challenging, leading to higher average RMSEs for all methods. Our method consistently outperforms the others, with at least $7.88\,\%$ and $10.73\,\%$ improvements over HTL and STL, respectively. Moreover, it exhibits greater stability, refleced by lower RMSEs at both the 75th and 25th percentiles.

Setting F: Effect of heterogeneous noise levels. We set $n_1 = n_2 = 300$, $p_0 = 100$, $p_1 = p_2 = 25$, $\rho_1 = 0.95$, $\rho_2 = 0.7$. By fixing $\sigma_2 = 0.1$ and varying σ_1 from 0.1 to 0.5, we assess the impact of different noise levels. Figure 5(f) shows that HTL lacks a clear advantage over STL, indicating that distribution heterogeneity leads to degenerated HTL's performance. Our MTL-HMB consistently outperforms the competing methods, demonstrating robustness in addressing both distribution and posterior heterogeneity. Furthermore, MTL-HMB excels at lower prediction levels, with neither STL nor HTL matching its performance at the 25th percentile.

4.2 MTL for Multiple tasks

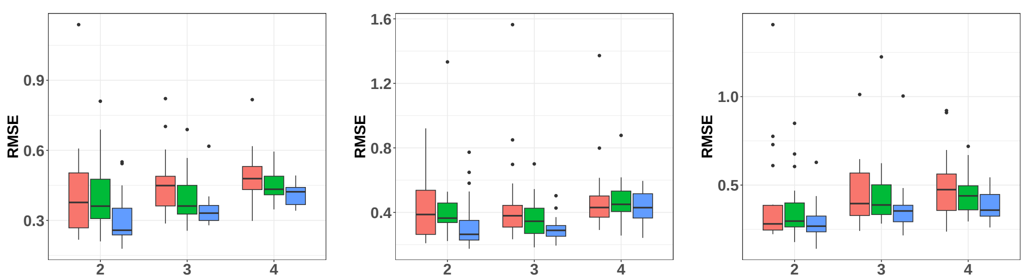
To save space, the DGP for number of tasks greater than 2 is detailed in Appendix A.2. We select $T = 2$, $T = 3$, and $T = 4$, with the prediction results summarized in Figure 6. As shown in Figure 6(a), increasing the number of heterogeneous tasks makes prediction more challenging, resulting in higher average RMSEs. This underscores the complexity of integrating diverse data. Nevertheless,



(a) $\rho_1 = \rho_2$ (b) $\rho_1 \neq \rho_2$ (c) α

(d) $n_1 = n_2$ (e) $p_1 = p_2$ (f) $\sigma_1 \neq \sigma_2$

Figure 5: Boxplots of average RMSEs under **Settings A** to **F** for the three methods. The methods are distinguished by color: orange for STL, green for HTL, and blue for the proposed MTL-HMB.



(a) Average (b) "Easy" (c) "Difficult"

Figure 6: Boxplots of average RMSEs under multiple settings across three methods. The methods are differentiated by color: orange for STL, green for HTL, and blue for our proposed MTL-HMB.

our MTL-HMB consistently outperforms the other methods and shows the smallest RMSE standard deviation, indicating greater robustness and reliability. We focus on the first task, considering it the "easiest" due to the highest observed correlations. Figure 6(b) shows that for this "easy" task, as more tasks are integrated, the improvement of our method decreases due to increasing heterogeneity. Additionally, we observe that in two-task learning, the second task is the most challenging; in three-task learning, it is the third task; and in four-task learning, it is the fourth task. This pattern indicates that as the number of integrated tasks increases, the complexity of learning escalates, particularly for the most recently added task. To quantify these challenges, we compile the RMSEs for the most difficult tasks in Figure 6(c), which shows that our MTL-HMB excels in these challenging tasks, consistently outperforming the other methods. For example, in the four-task integration, our method achieves over $18.22\,\%$ improvement compared to the HTL.

4.3 ADNI Real Data Application

We perform MTL using the ADNI database. The first task has 72 samples with features from MRI and PET sources, denoted as $\boldsymbol{X}_0^1$ and $\boldsymbol{X}_1^1$. The second task has 69 samples with features from MRI and GENE sources, denoted as $\boldsymbol{X}_0^2$ and $\boldsymbol{X}_2^2$. The MRI source includes 267 features, PET includes 113, and GENE includes 300. For the response variable, we use the Mini-Mental State Examination (MMSE), which measures cognitive impairment and serves as a diagnostic indicator of Alzheimer's disease (Tombaugh & McIntyre, 1992). We provide a detailed description of the ADNI database in Appendix A.5.1

Although both tasks share the MRI source, significant heterogeneity may still exist between the two datasets. To quantitatively assess this heterogeneity, we calculate the Maximum Mean Discrepancy (MMD) distance between $\boldsymbol{X}_0^1$ and $\boldsymbol{X}_0^2$. Additionally, a permutation test is conducted to determine whether the differences between these sample sets are statistically significant. The test yields a p-value of 1×10^{-6}, indicating significant differences between $\boldsymbol{X}_0^1$ and $\boldsymbol{X}_0^2$ and therefore a necessity of incorporating heterogeneity among homogeneous source in MTL. Furthermore, the small sample sizes in both tasks impose challenges for prediction, where MTL can potentially enhance performance. For both datasets, we use 60% of the samples for training, 20% for model selection and early stopping, and calculate RMSE on the remaining 20% for testing. The experiment is repeated 30 times, and the results are summarized in Table 1. Our MTL-HMB yields lower prediction errors in both tasks, particularly in Task 2, where it improves performance by at least 17.28% compared to the other two methods, despite the small sample sizes. HTL performs worse than STL due to ignoring the significant heterogeneity between $\boldsymbol{X}_0^1$ and $\boldsymbol{X}_0^2$.

Method	Task 1	Task 2
STL	2.74(0.87)	4.57(1.15)
HTL	2.86(0.75)	4.34(1.47)
Ours	**2.66(0.59)**	**3.59(0.98)**

Table 1: Prediction accuracy on testing data, measured by RMSE.

Figure 7 presents the t-SNE visualization of the latent representations obtained from a single training session, where the proposed MTL-HMB method effectively captures both shared and task-specific representations. Notably, the shared representations of the two tasks form a single cluster, while the task-specific representations of the two tasks exhibit significant differences in their distributions. This indicates that the datasets for the two tasks share certain commonalities while also displaying clear heterogeneity, which requires careful consideration during integration.

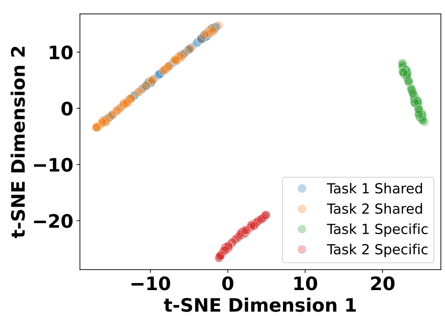


Figure 7: The t-SNE visualization of the learned task-specific and shared representations.

5 Discussion

In this paper, we propose a novel two-step strategy for effective MTL in the context of block-wise missing data in conjunction with different types of heterogeneity. The first step addresses distribution heterogeneity using integrated imputation, while the second step integrates learning to overcome distribution and posterior heterogeneity. We conduct extensive numerical experiments to validate the superiority of the proposed method across various levels of heterogeneity. Additionally, in the ADNI real-world dataset, our approach achieves significant improvements in both tasks. In the following, we provide the limitations and outline directions for future work, primarily focusing on transforming the two-step process into a single-step approach. In this unified method, the shared and task-specific hidden representations can be used for both imputing missing data and posterior learning simultaneously, as detailed in Appendix A.6.

References

Milad Abdollahzadeh, Touba Malekzadeh, and Ngai-Man Man Cheung. Revisit multimodal meta-learning through the lens of multi-task learning. *Advances in Neural Information Processing Systems*, 34:14632–14644, 2021.

Niccolo Anceschi, Federico Ferrari, David B Dunson, and Himel Mallick. Bayesian joint additive factor models for multiview learning. *arXiv preprint arXiv:2406.00778*, 2024.

Ting Bai, Yudong Xiao, Bin Wu, Guojun Yang, Hongyong Yu, and Jian-Yun Nie. A contrastive sharing model for multi-task recommendation. In *Proceedings of the ACM Web Conference 2022*, pp. 3239–3247, 2022.

Jose Bernal, Kaisar Kushibar, Daniel S Asfaw, Sergi Valverde, Arnau Oliver, Robert Martí, and Xavier Lladó. Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. *Artificial intelligence in medicine*, 95:64–81, 2019.

Ioana Bica and Mihaela van der Schaar. Transfer learning on heterogeneous feature spaces for treatment effects estimation. *Advances in Neural Information Processing Systems*, 35:37184–37198, 2022.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

Changxiao Cai, T Tony Cai, and Hongzhe Li. Transfer learning for contextual multi-armed bandits. *The Annals of Statistics*, 52(1):207–232, 2024.

T Tony Cai and Hongming Pu. Transfer learning for nonparametric regression: Non-asymptotic minimax analysis and adaptive procedure. *arXiv preprint arXiv:2401.12272*, 2024.

Tianxi Cai, Molei Liu, and Yin Xia. Individual data protected integrative regression analysis of high-dimensional heterogeneous data. *Journal of the American Statistical Association*, 117(540): 2105–2119, 2022.

Kai Cao, Qiyu Gong, Yiguang Hong, and Lin Wan. A unified computational framework for single-cell data integration with optimal transport. *Nature Communications*, 13(1):7419, 2022a.

Kai Cao, Yiguang Hong, and Lin Wan. Manifold alignment for heterogeneous single-cell multi-omics data integration using pamona. *Bioinformatics*, 38(1):211–219, 2022b.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Jingxiang Chen, Quoc Tran-Dinh, Michael R Kosorok, and Yufeng Liu. Identifying heterogeneous effect using latent supervised clustering with adaptive fusion. *Journal of Computational and Graphical Statistics*, 30(1):43–54, 2021.

SeoWon Gabriel Choi and Sungkyu Jung. Integrative decomposition of multi-source data by identifying partially-joint score subspaces. *arXiv preprint arXiv:2203.14041*, 2022.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829*, 2019.

Noah Cohen Kalafut, Xiang Huang, and Daifeng Wang. Joint variational autoencoders for multi-modal imputation and embedding. *Nature Machine Intelligence*, 5(6):631–642, 2023.

Bénédicte Colnet, Imke Mayer, Guanhua Chen, Awa Dieng, Ruohong Li, Gaël Varoquaux, Jean-Philippe Vert, Julie Josse, and Shu Yang. Causal inference methods for combining randomized trials and observational studies: a review. *Statistical science*, 39(1):165–191, 2024.

Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

Alicia Curth and Mihaela Van der Schaar. On inductive biases for heterogeneous treatment effect estimation. *Advances in Neural Information Processing Systems*, 34:15883–15894, 2021.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Sharing knowledge in multi-task deep reinforcement learning. *arXiv preprint arXiv:2401.09561*, 2024.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

Yaqi Duan and Kaizheng Wang. Adaptive and robust multi-task learning. *The Annals of Statistics*, 51(5):2015–2039, 2023.

Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

Qing Feng, Meilei Jiang, Jan Hannig, and JS Marron. Angle-based joint and individual variation explained. *Journal of multivariate analysis*, 166:241–265, 2018.

Qi Gao and Thomas CM Lee. High-dimensional variable selection in regression and classification with missing data. *Signal Processing*, 131:1–7, 2017.

Xing Gao, Sungwon Lee, Gen Li, and Sungkyu Jung. Covariate-driven factorization by thresholding for multiblock data. *Biometrics*, 77(3):1011–1023, 2021.

Irina Gaynanova and Gen Li. Structural learning and integrative decomposition of multi-view data. *Biometrics*, 75(4):1121–1132, 2019.

Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol. A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632, 2019.

Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023.

Zelin He, Ying Sun, and Runze Li. Transfusion: Covariate-shift robust transfer learning for high-dimensional regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 703–711. PMLR, 2024a.

Zelin He, Ying Sun, Jingyuan Liu, and Runze Li. Adatrans: Feature-wise and sample-wise adaptive transfer learning for high-dimensional regression. *arXiv preprint arXiv:2403.13565*, 2024b.

Ziniu Hu, Zhe Zhao, Xinyang Yi, Tiansheng Yao, Lichan Hong, Yizhou Sun, and Ed Chi. Improving multi-task generalization via regularizing spurious correlation. *Advances in Neural Information Processing Systems*, 35:11450–11466, 2022.

Jian Huang, Yuling Jiao, Wei Wang, Xiaodong Yan, and Liping Zhu. Integrative analysis for high-dimensional stratified models. *Statistica Sinica*, 33, 2023a.

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*, 2023b.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

Carson James, Dongbang Yuan, Irina Gaynanova, and Jesús Arroyo. Learning joint and individual structure in network data with covariates. *arXiv preprint arXiv:2406.08776*, 2024.

Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE transactions on image processing*, 26(9): 4509–4522, 2017.

David M Kent, Ewout Steyerberg, and David Van Klaveren. Personalized evidence based medicine: predictive approaches to heterogeneous treatment effects. *Bmj*, 363, 2018.

David M Kent, Jessica K Paulus, David Van Klaveren, Ralph D'Agostino, Steve Goodman, Rodney Hayward, John PA Ioannidis, Bray Patrick-Lake, Sally Morton, Michael Pencina, et al. The predictive approaches to treatment effect heterogeneity (path) statement. *Annals of internal medicine*, 172(1):35–45, 2020.

Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

Henry Lam, Kaizheng Wang, Yuhang Wu, and Yichen Zhang. Adaptive data fusion for multi-task non-smooth optimization. *arXiv preprint arXiv:2210.12334*, 2022.

Marine Le Morvan, Julie Josse, Erwan Scornet, and Gaël Varoquaux. What'sa good imputation to predict with missing values? *Advances in Neural Information Processing Systems*, 34:11530–11540, 2021.

Hyungi Lee, Giung Nam, Edwin Fong, and Juho Lee. Enhancing transfer learning with flexible nonparametric posterior sampling. *arXiv preprint arXiv:2403.07282*, 2024.

Jia-Hong Lee, Yi-Ming Chan, Ting-Yen Chen, and Chu-Song Chen. Joint estimation of age and gender from unconstrained face images using lightweight multi-task cnn for mobile applications. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 162–165. IEEE, 2018.

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

Jingmao Li, Qingzhao Zhang, Song Chen, and Kuangnan Fang. Weighted multiple blockwise imputation method for high-dimensional regression with blockwise missing data. *Journal of Statistical Computation and Simulation*, 93(3):459–474, 2023.

Quefeng Li and Lexin Li. Integrative factor regression and its inference for multimodal data analysis. *Journal of the American Statistical Association*, 117(540):2207–2221, 2022.

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.

Ting Li, Chengchun Shi, Qianglin Wen, Yang Sui, Yongli Qin, Chunbo Lai, and Hongtu Zhu. Combining experimental and historical data for policy evaluation. *arXiv preprint arXiv:2406.00317*, 2024a.

Yiming Li, Xuehan Yang, Ying Wei, and Molei Liu. Adaptive and efficient learning with blockwise missing and semi-supervised data. *arXiv preprint arXiv:2405.18722*, 2024b.

Cunjie Lin, Jingfu Peng, Yichen Qin, Yang Li, and Yuhong Yang. Optimal integrating learning for split questionnaire design type data. *Journal of Computational and Graphical Statistics*, 32(3): 1009–1023, 2023.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4487–4496, 2019.

Eric F Lock, Katherine A Hoadley, James Stephen Marron, and Andrew B Nobel. Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics*, 7(1):523, 2013.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Shujie Ma and Jian Huang. A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association*, 112(517):410–423, 2017.

Jeanne M Madden, Matthew D Lakoma, Donna Rusinak, Christine Y Lu, and Stephen B Soumerai. Missing clinical and behavioral health data in a large electronic health record (ehr) system. *Journal of the American Medical Informatics Association*, 23(6):1143–1149, 2016.

Subha Maity, Yuekai Sun, and Moulinath Banerjee. Meta-analysis of heterogeneous data: integrative sparse regression in high-dimensions. *Journal of Machine Learning Research*, 23(198):1–50, 2022.

Subha Maity, Diptavo Dutta, Jonathan Terhorst, Yuekai Sun, and Moulinath Banerjee. A linear adjustment-based approach to posterior drift in transfer learning. *Biometrika*, 111(1):31–50, 2024.

Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1851–1860, 2019.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21, 2008.

Łukasz Maziarka, Aleksandra Nowak, Maciej Wołczyk, and Andrzej Bedychaj. On the relationship between disentanglement and multi-task learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 625–641. Springer, 2022.

Nicolai Meinshausen and Peter Bühlmann. Maximin effects in inhomogeneous large-scale data. *The Annals of Statistics*, 43(4):1801–1830, 2015.

Seungwhan Moon and Jaime G Carbonell. Completely heterogeneous transfer learning with attention-what and what not to transfer. In *IJCAI*, volume 1, pp. 1–2, 2017.

Susanne G Mueller, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett. The alzheimer's disease neuroimaging initiative. *Neuroimaging Clinics of North America*, 15(4):869, 2005a.

Susanne G Mueller, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford R Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett. Ways toward an early diagnosis in alzheimer's disease: the alzheimer's disease neuroimaging initiative (adni). *Alzheimer's & Dementia*, 1(1):55–66, 2005b.

Nimisha G Nair, Pallavi Satpathy, Jabez Christopher, et al. Covariate shift: A review and analysis on classifiers. In *2019 Global Conference for Advancement in Technology (GCAT)*, pp. 1–6. IEEE, 2019.

Zhongzhe Ouyang, Lu Wang, and Alzheimer's Disease Neuroimaging Initiative†. Imputation-based variable selection method for block-wise missing data when integrating multiple longitudinal studies. *Mathematics*, 12(7):951, 2024.

Elise F Palzer, Christine H Wendt, Russell P Bowler, Craig P Hersh, Sandra E Safo, and Eric F Lock. sjive: Supervised joint and individual variation explained. *Computational statistics & data analysis*, 175:107547, 2022.

Josh Patterson and Adam Gibson. *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc.", 2017.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4822–4829, 2019.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

Hai Shu, Xiao Wang, and Hongtu Zhu. D-cca: A decomposition-based canonical correlation analysis for high-dimensional datasets. *Journal of the American Statistical Association*, 115(529): 292–306, 2020.

Shanshan Song, Yuanyuan Lin, and Yong Zhou. Semi-supervised inference for block-wise missing data without imputation. *Journal of Machine Learning Research*, 25(99):1–36, 2024.

Indhupriya Subramanian, Srikant Verma, Shiva Kumar, Abhay Jere, and Krishanpal Anamika. Multi-omics data integration, interpretation, and its application. *Bioinformatics and biology insights*, 14:1177932219899051, 2020.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*, 2018.

Lu Tang and Peter XK Song. Fused lasso approach in regression coefficients clustering–learning parameter heterogeneity in data integration. *Journal of Machine Learning Research*, 17(113): 1–23, 2016.

Xiwei Tang, Fei Xue, and Annie Qu. Individualized multidirectional variable selection. *Journal of the American Statistical Association*, 116(535):1280–1296, 2021.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Ye Tian and Yang Feng. Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, 118(544):2684–2697, 2023.

Ye Tian, Haolei Weng, and Yang Feng. Unsupervised multi-task and transfer learning on gaussian mixture models. *arXiv preprint arXiv:2209.15224*, 2022.

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

Tom N Tombaugh and Nancy J McIntyre. The mini-mental state examination: a comprehensive review. *Journal of the American Geriatrics Society*, 40(9):922–935, 1992.

Xinming Tu, Zhi-Jie Cao, Sara Mostafavi, Ge Gao, et al. Cross-linked unified embedding for cross-modality representation learning. *Advances in Neural Information Processing Systems*, 35: 15942–15955, 2022.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

Karen Ullrich, Edward Meeds, and Max Welling. Soft weight-sharing for neural network compression. *arXiv preprint arXiv:1702.04008*, 2017.

Haodong Wang, Quefeng Li, and Yufeng Liu. Multi-response regression for block-missing multimodal data without imputation. *Statistica Sinica*, 34(2):527, 2024a.

Jiuzhou Wang and Eric F Lock. Multiple augmented reduced rank regression for pan-cancer analysis. *Biometrics*, 80(1):ujad002, 2024.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024b.

Peiyao Wang, Yufeng Liu, and Dinggang Shen. Flexible locally weighted penalized regression with applications on prediction of alzheimer's disease neuroimaging initiative's clinical scores. *IEEE transactions on medical imaging*, 38(6):1398–1408, 2018.

Peiyao Wang, Quefeng Li, Dinggang Shen, and Yufeng Liu. High-dimensional factor regression for heterogeneous subpopulations. *Statistica Sinica*, 33(1):27, 2023a.

Peiyao Wang, Haodong Wang, Quefeng Li, Dinggang Shen, and Yufeng Liu. Joint and individual component regression. *Journal of Computational and Graphical Statistics*, pp. 1–16, 2023b.

Huangyuan Wu, Bin Li, Lianfang Tian, Junjian Feng, and Chao Dong. An adaptive loss weighting multi-task network with attention-guide proposal generation for small size defect inspection. *The Visual Computer*, 40(2):681–698, 2024.

Shuo Xiang, Lei Yuan, Wei Fan, Yalin Wang, Paul M Thompson, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. Bi-level multi-source learning for heterogeneous block-wise missing data. *NeuroImage*, 102:192–206, 2014.

Fei Xue and Annie Qu. Integrating multisource block-wise missing data in model selection. *Journal of the American Statistical Association*, 116(536):1914–1927, 2021.

Fei Xue, Rong Ma, and Hongzhe Li. Statistical inference for high-dimensional linear regression with blockwise missing data. *arXiv preprint arXiv:2106.03344*, 2021.

Sangyoon Yi, Raymond Ka Wai Wong, and Irina Gaynanova. Hierarchical nuclear norm penalization for multi-view data integration. *Biometrics*, 79(4):2933–2946, 2023.

Guan Yu, Quefeng Li, Dinggang Shen, and Yufeng Liu. Optimal sparse linear prediction for block-missing multi-modality data without imputation. *Journal of the American Statistical Association*, 115(531):1406–1419, 2020.

Lei Yuan, Yalin Wang, Paul M Thompson, Vaibhav A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage*, 61(3):622–632, 2012.

Ruqian Zhang, Yijiao Zhang, Annie Qu, Zhongyi Zhu, and Juan Shen. Concert: Covariate-elaborated robust local information transfer with conditional spike-and-slab prior. *arXiv preprint arXiv:2404.03764*, 2024a.

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*, 119(545):485–497, 2024b.

Yijiao Zhang and Zhongyi Zhu. Transfer learning for high-dimensional quantile regression via convolution smoothing. *arXiv preprint arXiv:2212.00428*, 2022.

Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1): 30–43, 2018.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

Yunfeng Zhang and Irina Gaynanova. Joint association and classification analysis of multi-view data. *Biometrics*, 78(4):1614–1625, 2022.

Tianqi Zhao, Guang Cheng, and Han Liu. A partially linear framework for massive heterogeneous data. *Annals of statistics*, 44(4):1400, 2016.

Doudou Zhou, Tianxi Cai, and Junwei Lu. Multi-source learning via completion of block-wise overlapping noisy matrices. *arXiv preprint arXiv:2105.10360*, 2021.

Doudou Zhou, Tianxi Cai, and Junwei Lu. Multi-source learning via completion of block-wise overlapping noisy matrices. *Journal of Machine Learning Research*, 24(221):1–43, 2023.

A APPENDIX

In Appendix A.1, we further expand on the related works described in Section 2.

In Appendix A.2, we provide the detailed DGP used in Section 4.2.

In Appendix A.3, we conduct ablation experiments to demonstrate the individual roles of the two steps in our proposed method.

In Appendix A.4, we provide the pseudo-code for the proposed MTL-HMB.

In Appendix A.5, we include detailed experimental information, including the real data description and implementation details.

In Appendix A.6, we discuss the limitations of our work and potential future research directions.

In Appendix A.7, we display all qualitative results in Section 4 and compare with additional statistical method.

A.1 EXPANDED RELATED WORKS

Multi-group data integration. Multi-group data integration and MTL share the common goal of learning from multiple datasets or tasks simultaneously. The input features and response of a single task can be viewed as a separate group. There are several existing methods in the statistical literature for multi-group data analysis, which can be broadly classified into three categories. The first category designs specialized regression models (Meinshausen & Bühlmann, 2015; Zhao et al., 2016; Wang et al., 2018; Huang et al., 2023a;b) or factor regression models (Wang et al., 2023a;b) to handle large-scale heterogeneous data and identify group-specific structures. The second category employs specified parameter space constraints, such as fused penalties, to estimate regression coefficients that capture subgroup structures (Tang & Song, 2016; Ma & Huang, 2017; Chen et al., 2021; Li & Sang, 2019; Tang et al., 2021; Lam et al., 2022; Duan & Wang, 2023; Zhang et al., 2024b). The third category involves transfer learning, which borrows information from source data to target data (Li et al., 2022; Tian et al., 2022; Zhang & Zhu, 2022; Tian & Feng, 2023; Cai & Pu, 2024; Cai et al., 2024; He et al., 2024a; Zhang et al., 2024a). The aforementioned multi-group data integration approaches address distribution and posterior heterogeneity but overlook block-wise missing issues. Additionally, most methods rely on structured model assumptions, such as linearity, limiting their capacity to capture complex relationships.

Heterogeneous feature spaces. Existing transfer learning methods mainly addressed either distribution shift or posterior shift separately, with fewer studies considering both types of shifts simultaneously. For instance, Moon & Carbonell (2017) investigated scenarios with both heterogeneous feature and label spaces in the context of natural language processing. They proposed a method that learned a common embedding for the features and labels and then established a mapping between them. Similarly, Bica & van der Schaar (2022) focused on a shared label space but assumed that all tasks had a common source, utilizing the same encoder to extract shared representations. However, this assumption was often unrealistic in practice. Even when sources were identical, different tasks could exhibit significant heterogeneity due to variations in subjects, locations, and experimental settings. For example, in our ADNI real data (Section 4.3), tasks sharing MRI features might still differ due to varying experimental conditions. A key distinction in our method is that we treat this problem as a block-wise missing data issue rather than simply considering each task to have only the observed features. This perspective aligns more closely with the reality of medical data, where missing problem is common, and these missing features can also influence the response. Additionally, we focus on MTL, which is designed for numerous small-sample and challenging tasks. In contrast, transfer learning often assumed the existence of a large-scale dataset to support a smaller-sample task. For example, in the experiments conducted by Bica & van der Schaar (2022), the source domain's sample size was typically more than ten times that of the target domain. However, in real-world scenarios, it is more common for all tasks to have relatively small and limited sample sizes. Our method aims to provide a more comprehensive and robust learning framework by integrating heterogeneous information across these small-sample tasks.

Block-wise Statistical Methods. Numerous statistical methods for block-wise missing data have been proposed, and we provide a more detailed discussion here. Yu et al. (2020); Wang et al.

(2024a) learn linear predictors through covariance matrix and cross-covariance vector, which can be estimated with block-missing data without imputation. Xue & Qu (2021); Xue et al. (2021) propose a multiple block-wise imputation (MBI) approach to construct estimating equations based on all available information and integrate estimating functions to achieve efficient estimators. Li et al. (2024b) leverages block-wise missing labeled samples and further enhances estimation efficiency by incorporating large unlabeled samples through imputation and projection. Their method is robust to model misspecification on the missing covariates. Song et al. (2024) address a similar problem under the semi-supervised learning setting, employing a double debiased procedure without relying on imputation. Zhou et al. (2023) develop an efficient block-wise overlapping noisy matrix integration algorithm to obtain multi-source embeddings. These methods have demonstrated strong performance in various real-world applications. For instance, Zhou et al. (2023); Li et al. (2024b) validated their methods on electronic health record (EHR) data, demonstrating their effectiveness in real-world applications. However, all the aforementioned methods suffer from several limitations. First, they primarily capture linear relationships and struggle to effectively learn nonlinear patterns. Many real-world datasets, such as multi-modal single-cell data (Tu et al., 2022; Cohen Kalafut et al., 2023) and imaging data (Jin et al., 2017; Bernal et al., 2019), exhibit complexities that further limit the applicability of these methods. This limitation underscores the motivation for adopting an encoder-decoder framework in our work. Second, these methods consider the homogeneous model setup for different tasks, for instance, assuming the same regression coefficients are applied to all tasks. However, data heterogeneity across tasks or sources are ubiquitous in real applications, either marginal distribution of sources or conditional distribution among sources can be distinct, which complicates the modeling procedure. This is another motivation of our project, to effectively handle multiple types of heterogeneity simultaneously.

A.2 Data Generation Process in Section 4.2

We consider MTL for multiple tasks. The DGP is similar to that in Section 4.1 but is extended to accommodate more tasks. For three-task learning, the features for the t-th task are denoted as $\boldsymbol{x}^t = [\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\boldsymbol{x}_2^t|\boldsymbol{x}_3^t]$ and follow a Gaussian distribution with mean $\mathbf{0}$ and an exchangeable covariance matrix. The variance is fixed at 1, and the covariance structure is determined by $(\rho_t)^{0.01|i-j|}$. We randomly generate n_t samples, with only $\boldsymbol{x}_0^t$ and $\boldsymbol{x}_t^t$ being observed. The response y^t is given by:

$$y^t = \alpha \sum_{d=1}^{p} v_{c,d}(x_d^t)^2/p + (1-\alpha)\sum_{d=1}^{p} v_{t,d}x_d^t/p + \varepsilon_r, \quad \forall r \in [3].$$

where $p = \sum_{s=0}^{3} p_s$. For three-task learning, we choose the following parameters: $n_1 = n_2 = n_3 = 300$, $p_0 = 125$, $p_1 = p_2 = p_3 = 25$, $\rho_1 = 0.95$, $\rho_2 = \rho_3 = 0.9$, $\alpha = 0.3$, $v_c, v_t \sim N(-10, 10^2)$, and $\varepsilon_t \sim N(0, 0.01)$ for $t \in [3]$.

For four-task learning, the features for the t-th task are denoted as $\boldsymbol{x}^t = [\boldsymbol{x}_0^t|\boldsymbol{x}_1^t|\boldsymbol{x}_2^t|\boldsymbol{x}_3^t|\boldsymbol{x}_4^t]$ and follow a Gaussian distribution with mean $\mathbf{0}$ and an exchangeable covariance matrix. The variance is fixed at 1, and the covariance structure is determined by $(\rho_t)^{0.01|i-j|}$. We randomly generate n_t samples, with only $\boldsymbol{x}_0^t$ and $\boldsymbol{x}_t^t$ being observed. The response y^t is given by:

$$y^t = \alpha \sum_{d=1}^{p} v_{c,d}(x_d^t)^2/p + (1-\alpha)\sum_{d=1}^{p} v_{t,d}x_d^t/p + \varepsilon_r, \quad \forall r \in [4].$$

where $p = \sum_{s=0}^{4} p_s$. For four-task learning, we choose the following parameters: $n_1 = n_2 = n_3 = n_4 = 300$, $p_0 = 125$, $p_1 = p_2 = p_3 = p_4 = 25$, $\rho_1 = 0.95$, $\rho_2 = \rho_3 = \rho_4 = 0.9$, $\alpha = 0.3$, $v_c, v_t \sim N(-10, 10^2)$, and $\varepsilon_t \sim N(0, 0.01)$ for $t \in [4]$.

For evaluation, we focus on the average RMSE across all tasks in the testing data, defined as follows:

$$\text{RMSE} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{n_{t,\text{test}}}\sum_{i=1}^{n_{t,\text{test}}}(\widehat{y}_i^t - y_i^t)^2}.$$

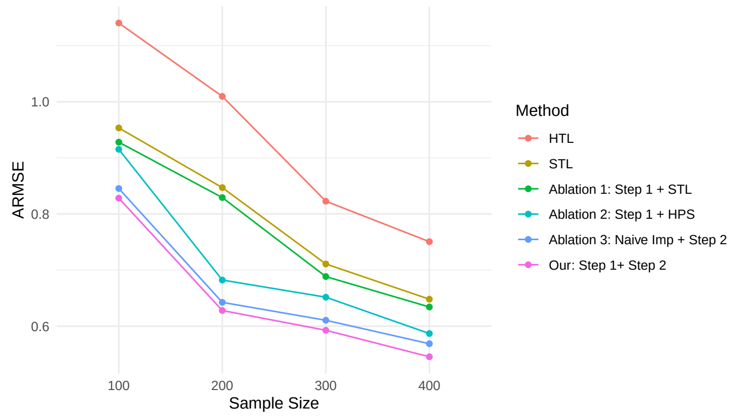


Figure 8: The average RMSEs of all methods across different n_2 sample sizes. HPS refers to hard parameter sharing, and Imp refers to imputation.

A.3 Ablation Experiments

We propose an MTL framework that involves two steps: Step 1 for HBI (see Section 3.1) and Step 2 for heterogeneous MTL (see Section 3.2). To assess the independent effect of each step, we design ablation experiments. In addition to comparing with STL and HTL, we consider three new ablation experiments. The first is Step 1 + STL, which applies HBI followed by STL to evaluate the effect of Step 2 and is denoted as Ablation 1. The second approach is Step 1 combined with a common MTL framework (Ablation 2). Specifically, we adopt the hard parameter sharing (HPS) framework, which shares the main layers across tasks while differentiating in the final layer, and is widely used in MTL (Liu et al., 2019; Bai et al., 2022). However, hard parameter sharing struggles to address distribution heterogeneity due to the shared structure in the first $L-1$ layers. The third is naive imputation + Step 2, where we ignore distribution heterogeneity to analyze the impact of disregarding heterogeneity in imputation, denoted as Ablation 3.

The data generation process is consistent with Section 4.1, but we adopt a more challenging setting. Specifically, we set $p_1 = 100$, $p_2 = 25$, $p_3 = 25$, $\rho_1 = 0.8$, $\rho_2 = 0.6$, $\alpha = 0.3$, and $\sigma_1 = \sigma_2 = 0.1$. We analyze the impact of sample sizes n_1 and n_2 on three methods by fixing $n_1 = 300$ and varying n_2 as $n_2 = k \times 100$ for $k = 1, \ldots, 4$. The experiments are repeated 30 times, and the mean RMSE per task is computed, with the results summarized in Figure 8. It is important to note that, due to the presence of distribution heterogeneity in this setting, HTL performs the worst.

We analyze the ablation results from different perspectives. First, it is evident that both Ablation 3 and our proposed MTL-HMB outperform STL, Ablation 1, and Ablation 2, indicating that Step 2 plays a crucial role in enhancing prediction performance. Second, by comparing Ablation 1 with STL, we observe that Ablation 1 consistently achieves lower loss across different sample sizes, demonstrating that Step 1 improves predictions for a single dataset. Third, when comparing Ablation 3 with our proposed method, Ablation 3 shows higher loss, suggesting that ignoring distribution heterogeneity in imputation negatively impacts performance. Fourth, we compare Ablation 1, Ablation 2, and our proposed MTL-HMB method, all of which incorporate Step 1. The prediction results demonstrate that our method outperforms both Ablation 2 and Ablation 1. This indicates that our MTL framework in Step 2 is more effective than hard parameter sharing, as it accounts for distribution heterogeneity, while hard parameter sharing performs better than STL. Fifth, even when comparing Ablation 2 with Ablation 3—which uses a less effective imputation method—the latter still achieves better predictive performance. This further highlights the advantages of Step 2 over traditional MTL approaches. Overall, the ablation experiments demonstrate that when both distribution and posterior heterogeneity are present, both steps of our proposed framework are crucial.

A.4 Pseudo-code for Our Proposed MTL-HMB

Algorithm 1 provides the pseudo-code for training our proposed MTL method. For simplicity, we set the mini-batch size to be the same across all T datasets: $B^t = B$ for $t \in [T]$. For HBI, we divide the data into training and testing sets and train the parameters on the training set. Early stopping is applied to $\mathcal{L}_{\text{pre}}$ on the t-th dataset's testing data to check for convergence and perform model selection. For heterogeneous MTL, the data is split into training, validation, and testing sets. Parameters are trained on the training set, and the best hyperparameter combination is selected using the validation set. Early stopping is applied to $\mathcal{L}_{\text{integ}}$ on the validation set to check for convergence, and the final prediction metrics are calculated on the test set. In practice, we choose the regularization parameters γ, δ, and κ from the set $[0.01, 0.1, 1]$ for $\mathcal{R}_{\text{orth}}$, $\mathcal{R}_{\text{imp}}$, and $\mathcal{R}_{\text{dr}}$. In our experiments, we found that the selection of γ, δ, and κ is robust, having minimal impact on the final prediction performance.

A.5 Experimental details

A.5.1 Dataset description

In this subsection, we provide a detailed description of the ADNI database used in Section 4.3. The ADNI study (Mueller et al., 2005a) aims to identify biomarkers that track the progression of Alzheimer's disease (AD). The MMSE score, which measures cognitive impairment, is treated as the response variable, and we aim to select biomarkers from three complementary data sources: MRI, PET, and gene expression. Given the sparsity assumption, we use region of interest (ROI) level data rather than raw imaging data, as the latter might not be suitable for our method. MRI variables include volumes, cortical thickness, and surface areas, while PET features represent standard uptake value ratios (SUVR) of different ROIs. Gene expression variables are derived from blood samples and represent expression levels at different gene probes. To reduce the number of gene expression variables, we apply sure independence screening (SIS), narrowing it down to 300 variables. This results in a total of 680 features, including 267 MRI features and 113 PET features. The data is sourced from ADNI-2 at month 48, where block-wise missingness occurs due to factors such as low-quality images or patient dropout. Using visit codes, we align MMSE with the imaging data to ensure they are measured within the same month. Ultimately, we obtained two datasets: dataset 1 contains only MRI and PET sources, while dataset 2 includes MRI and gene expression sources. Both datasets have relatively small sample sizes, underscoring the importance of effectively using incomplete observations in the analysis.

A.5.2 Implementation details and hyperparameter tuning

In Section 4, we compare our proposed method (MTL-HMB) with Single Task Learning (STL) and Heterogeneous Transfer Learning (HTL). Here, we provide the implementation details of these three methods. For STL, we use standard deep neural networks to train each dataset individually. In contrast, HTL assumes no heterogeneity in the anchoring source and extracts task-shared representations from it, while task-specific representations are derived from task-specific sources.

STL. For STL, we use standard deep neural networks to train each dataset individually. Each dataset is split into 60% for training, 20% for validation, and 20% for testing. On the training set, we perform hyperparameter tuning, including network width from $\{32, 64, 128\}$, depth from $\{2, 3, 4, 5\}$, and batch size from $\{8, 16, 32\}$ (with 8 included due to the smaller sample size in the ADNI database). Additionally, we set the learning rate to 0.001 and the early-stopping patience to 30. To stabilize the optimization during iterations, we use the exponential scheduler (Patterson & Gibson, 2017), which decays the learning rate by a constant per epoch. In all numerical tasks, we set the decay constant to 0.95, applied every 200 iterations. We tune the hyperparameters and select the best model on the validation set. Finally, the tuned hyperparameters are used to compute the prediction loss on the testing set.

HTL. For HTL, we adapt the network architecture from Bica & van der Schaar (2022) and modify it for our setting. Following their approach, the framework for handling heterogeneous feature spaces consists of a common encoder for shared source and task-specific encoders for task-specific sources, implemented using deep neural networks. The network widths are selected from $\{32, 64, 128\}$ and depths from $\{2, 3, 4\}$. The remaining components are incorporated into an MTL network architecture, similar to the structure described in Section 3.2, where shared and task-specific pathways have depths chosen from $\{2, 3, 4\}$. The output dimensions of the first $L-1$ layers are selected from $\{32, 64, 128\}$, with the final layer predicting the corresponding response. The batch size is chosen from $\{8, 16, 32\}$, and the learning rate is set to 0.001. To remain consistent with Bica & van der Schaar (2022), we train the prediction loss on the training set, along with regularization terms $\mathcal{R}_{\text{orth}}$ and $\mathcal{R}_{\text{dr}}$. Early stopping and hyperparameter tuning are performed based on the sum of the prediction losses across all datasets on the validation set, with an early-stopping patience of 30. Finally, the tuned hyperparameters are used to compute the prediction loss on the testing set.

MTL-HMB. For the proposed MTL-HMB, we describe the method in two steps: Step 1 and Step 2. **Step 1:** In HBI, the common encoder, task-specific encoders, decoder, and predictor use network architectures with widths selected from $\{8, 16, 32\}$ and depths from $\{1, 2, 3\}$. The batch size is chosen from $\{8, 16, 32\}$, and the learning rate is set to 0.001. Notably, since the features in our simulated data exhibit relatively simple linear relationships, we include smaller network widths and depths in our tuning. **Step 2:** To construct task-shared and task-specific mappings, the network architecture for the shared encoder ϕ_c and the task-specific encoders ϕ_p^t have widths selected from $\{32, 64, 128\}$ and depths from $\{2, 3, 4\}$. The output dimensions are also chosen from $\{32, 64, 128\}$. For the prediction function, both shared and task-specific pathways have depths chosen from $\{2, 3, 4\}$, with the output dimensions of the first $L-1$ layers selected from $\{32, 64, 128\}$. The final layer predicts the corresponding response. The batch size is chosen from $\{8, 16, 32\}$, and the learning rate is set to 0.001. Early stopping and hyperparameter tuning are conducted based on the sum of the prediction losses across all datasets on the validation set, using an early-stopping patience of 30. Finally, the tuned hyperparameters are applied to compute the prediction loss on the testing set.

A.6 Discussion about limitations and future work

First, our proposed method essentially assumes that there is common information across all tasks that can be fused, which implies a relatively strong shared structure. For example, in Section 3.2, we assume the existence of a common mapping, ψ_c, between input features and responses for all $r \in [R]$. However, in reality, when there is strong heterogeneity across multiple datasets, the shared structure is often only partial. For instance, in three datasets, only two may share the common ψ_c, while the third task may be too heterogeneous to fuse with the first two. In such cases, an adaptive approach for MTL is needed, one that explores partially shared information among tasks while preserving the uniqueness of the highly heterogeneous task. Currently, some studies have considered adaptive MTL in relatively simple settings, such as linear cases (Duan & Wang, 2023; Tian et al., 2023). However, adaptive MTL in the presence of block-wise, distribution, and posterior heterogeneity remains unexplored, making it a meaningful direction for future research.

Moreover, it is worth noting that in both Section 3.1 (HBI) and Section 3.2 (MTL), the hidden representations in each dataset are learned in two steps: the first step for imputation and the second step for learning the response. This process introduces some computational redundancy. A possible improvement would be to combine these two steps into one, unifying multiple tasks to learn the hidden representations for each task, which can then be used for both imputation and response learning. However, this approach poses computational challenges, such as how to balance different loss functions to achieve both accurate imputation and prediction. Thus, this remains a future research direction worth exploring.

A.7 Qualitative Results

As suggested by the reviewers, we display all qualitative results from Section 4 in this subsection. Specifically, we report the means and standard deviations of the average RMSEs under different experimental settings. Additionally, we apply the statistical block-wise imputation method (MBI) proposed by Xue & Qu (2021); Xue et al. (2021) to various simulation settings and the ADNI real dataset. In particular, MBI does not account for distribution or posterior heterogeneity. It assumes that the relationships among all sources across different tasks are consistent, as well as the relationships between the sources and the response. The method first imputes all missing blocks and then constructs estimating equations based on the available information. These estimating equations are subsequently integrated to achieve efficient estimators. We used the R package `BlockMissingData` to conduct the experiments, with the tuning parameters set to their default values. The RMSE was computed on a 20% testing set.

Tables 2 to 7 present the prediction results under settings A to F in Section 4.1, while Table 8 corresponds to the results in Section 4.2. We have included the prediction results of the MBI method for comparison. The performance of STL, HTL, and the proposed MTL-HMB methods is extensively discussed in Section 4, where it is evident that MBI significantly underperforms compared to these three methods. For example, as shown in Table 2, the prediction error of MBI is several times higher than that of the proposed MTL-HMB method. This poor performance can be attributed to several factors. First, MBI cannot handle nonlinear relationships and is limited to modeling linear relationships between sources and the response, which severely restricts its learning capacity. These findings underscore the substantial benefits of leveraging the encoder-decoder framework. Second, MBI is unable to address distribution or posterior heterogeneity.

To ensure a fair comparison, we reconsidered a linear data-generating process (DGP). Specifically, we modified the nonlinear DGP described in Section 4.1 to a simpler linear DGP as follows:

$$y^1 = \alpha \sum_{d=1}^{p} v_{c,d} x_d^1 / p + (1-\alpha) \sum_{d=1}^{p} v_{1,d} x_d^1 / p + \varepsilon_1,$$
$$y^2 = \alpha \sum_{d=1}^{p} v_{c,d} x_d^2 / p + (1-\alpha) \sum_{d=1}^{p} v_{2,d} x_d^2 / p + \varepsilon_2,$$

Where other parameters and settings remain unchanged, we evaluated the prediction performance of the four methods under this linear DGP. The results, presented in Table 9, show that MTL-HMB still achieves the best performance, followed by STL. HTL is constrained by distribution heterogeneity, while MBI, despite being designed for linear cases, suffers significant errors starting from the imputation step due to its assumption of no distribution or posterior heterogeneity. Consequently, its final predictions are notably poor.

Additionally, we evaluated MBI on the ADNI real dataset. The prediction results for Task 1 and Task 2 were $9.847, (3.516)$ and $10.272, (3.448)$, respectively. These findings further demonstrate the significant improvements brought by the encoder-decoder framework in real-world applications.

Table 2: Average RMSEs under **Setting A**.

$\rho_1 = \rho_2$	STL	HTL	MTL-HMB	MBI
0.5	0.650(0.116)	0.593(0.130)	0.593(0.188)	5.155(0.936)
0.6	0.604(0.150)	0.529(0.114)	0.474(0.098)	5.089(0.936)
0.7	0.535(0.170)	0.434(0.077)	0.421(0.107)	4.921(0.842)
0.8	0.558(0.196)	0.452(0.098)	0.382(0.118)	4.782(0.821)
0.9	0.421(0.155)	0.463(0.138)	0.345(0.125)	4.651(0.779)
0.95	0.376(0.169)	0.413(0.182)	0.270(0.064)	4.516(0.696)

Table 3: Average RMSEs under **Setting B**.

$\rho_1 \neq \rho_2$	STL	HTL	MTL-HMB	MBI
0.5	0.529(0.145)	0.476(0.128)	0.410(0.131)	5.125(0.961)
0.6	0.485(0.170)	0.466(0.140)	0.378(0.128)	5.013(0.849)
0.7	0.489(0.129)	0.485(0.116)	0.375(0.143)	4.966(0.856)
0.8	0.457(0.154)	0.463(0.109)	0.333(0.116)	4.902(0.821)
0.9	0.444(0.200)	0.402(0.127)	0.314(0.126)	4.807(0.806)

Table 4: Average RMSEs under **Setting C**.

α	STL	HTL	MTL-HMB	MBI
0.1	0.218(0.056)	0.214(0.035)	0.191(0.050)	1.825(0.336)
0.2	0.366(0.093)	0.372(0.134)	0.300(0.085)	3.356(0.564)
0.3	0.558(0.196)	0.452(0.098)	0.382(0.118)	4.921(0.842)
0.4	0.585(0.154)	0.587(0.192)	0.511(0.207)	6.551(1.180)
0.5	0.810(0.280)	0.676(0.130)	0.583(0.192)	8.152(1.424)
0.6	0.970(0.246)	0.871(0.237)	0.809(0.266)	9.711(1.668)

Table 5: Average RMSEs under **Setting D**.

$n_1 = n_2$	STL	HTL	MTL-HMB	MBI
100	0.893(0.403)	0.797(0.294)	0.598(0.277)	4.474(1.118)
200	0.526(0.141)	0.532(0.112)	0.452(0.262)	4.362(0.803)
300	0.489(0.129)	0.485(0.116)	0.375(0.143)	4.966(0.856)
400	0.365(0.096)	0.400(0.105)	0.327(0.084)	5.091(0.726)
500	0.332(0.066)	0.367(0.102)	0.288(0.046)	5.140(0.627)
600	0.303(0.060)	0.350(0.064)	0.276(0.055)	4.867(0.693)

Table 6: Average RMSEs under **Setting E**.

$p_1 = p_2$	STL	HTL	MTL-HMB	MBI
25	0.489(0.129)	0.485(0.116)	0.375(0.143)	4.966(0.856)
50	0.660(0.141)	0.658(0.232)	0.531(0.170)	4.414(0.874)
75	0.712(0.175)	0.649(0.168)	0.585(0.161)	4.745(0.548)
100	0.786(0.201)	0.886(0.208)	0.692(0.118)	4.709(0.770)

Table 7: Average RMSEs under **Setting F**.

σ_1	STL	HTL	MTL-HMB	MBI
0.1	0.489(0.129)	0.485(0.116)	0.375(0.143)	4.966(0.856)
0.2	0.499(0.156)	0.438(0.071)	0.389(0.098)	4.961(0.877)
0.3	0.536(0.119)	0.522(0.102)	0.454(0.093)	4.938(0.858)
0.4	0.543(0.120)	0.569(0.131)	0.466(0.105)	4.974(0.872)
0.5	0.663(0.162)	0.647(0.160)	0.511(0.082)	5.002(0.855)

Table 8: Average RMSEs under multiple tasks.

T	SDL	HTL	Proposed	MBI
2	0.429(0.209)	0.398(0.150)	0.310(0.111)	4.776(0.688)
3	0.468(0.129)	0.394(0.109)	0.344(0.074)	5.861(0.878)
4	0.490(0.116)	0.451(0.066)	0.410(0.044)	6.902(0.976)

Table 9: Average RMSEs under linear setting.

STL	HTL	MTL-HMB	MBI
0.295(0.028)	0.765(0.167)	0.274(0.029)	0.525(0.296)

Algorithm 1 Pseudo-code for Our Proposed MTL-HMB.

Input: T datasets denoted by $\{\boldsymbol{x}_i^t, y_i^t\}_{i=1}^{n_t}$, where $\boldsymbol{x}_i^t$ includes two blocks $\boldsymbol{x}_{0,i}^t$ and $\boldsymbol{x}_{t,i}^t$, learning rate η, mini-batch size for the t-th dataset is denoted by B^t.
Step 1: HBI
for $t = 1, \ldots, T$ **do** ▷ Imputation for task t-specific source
  Initialize: $\boldsymbol{\theta}^t$ (all parameters in this step)
  while not converged **do**
    Sample mini-batch of B^t demonstrations from the t-th dataset $\{\boldsymbol{x}_i^t, y_i^t\}_{i=1}^{n_t}$ and mini-batch combination of $B^{-t} = \sum_{s \neq t} B^t$ demonstrations from the rest $T-1$ datasets.
    for $i = 1, \ldots, B^t$ **do** ▷ Process batch from the t-th dataset.
      $\boldsymbol{f}_i^t = E_c(\boldsymbol{x}_{0,i}^t)$, $\boldsymbol{g}_i^t = E_p^t(\boldsymbol{x}_{0,i}^t)$
    end for
    Compute prediction loss $\mathcal{L}_{\text{pre}}^t = \sum_{i=1}^{B^t} l(\boldsymbol{x}_{t,i}^t, G(\boldsymbol{f}_i^t))$
    Compute reconstruction loss $\mathcal{L}_{\text{recon}}^t = \sum_{i=1}^{B^t} l(\boldsymbol{x}_{0,i}^t, D(\boldsymbol{f}_i^t, \boldsymbol{g}_i^t))$
    for $i = 1, \ldots, B^{-t}$ **do** ▷ Process batch from the rest $T-1$ datasets.
      $\boldsymbol{f}_i^{-t} = E_c(\boldsymbol{x}_{0,i}^{-t})$, $\boldsymbol{g}_i^{-t} = E_p^{-t}(\boldsymbol{x}_{0,i}^{-t})$
    end for
    Compute reconstruction loss: $\mathcal{L}_{\text{recon}}^{-t} = \sum_{i=1}^{B^{-t}} l(\boldsymbol{x}_{0,i}^{-t}, D(\boldsymbol{f}_i^{-t}, \boldsymbol{g}_i^{-t}))$
    Parameter update $\boldsymbol{\theta}^t \leftarrow \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t}(\mathcal{L}_{\text{pre}}^t + \mathcal{L}_{\text{recon}}^t + \mathcal{L}_{\text{recon}}^{-t})$
  end while
  for $i = 1, \ldots, B^{-t}$ **do**
    Imputation for task t-specific source: $\widehat{\boldsymbol{x}}_{t,i}^{-t} = \widehat{G}(\widehat{E}_c(\boldsymbol{x}_{0,i}^{-t}))$
  end for
end for
Obtain samples with reconstructed features $\{(\boldsymbol{x}_{0,i}^t, \ldots, \widehat{\boldsymbol{x}}_{t-1,i}^t, \boldsymbol{x}_{t,i}^t, \widehat{\boldsymbol{x}}_{t+1,i}^t, \ldots, \widehat{\boldsymbol{x}}_{T,i}^t), y_i^t)\}_{i=1}^{n_t}$

Step 2: Heterogeneous MTL
Initialize: $\boldsymbol{\Theta}$ (all parameters in this step)
while not converged **do**
  for $t = 1, \ldots, T$ **do**
    for $i = 1 \ldots B^t$ **do** ▷ Process batch from the r-th dataset.
      $\boldsymbol{h}_i^t = \phi_c(\boldsymbol{x}_{0,i}^t)$, $\boldsymbol{k}_i^t = \phi_p^t([\boldsymbol{x}_{0,i}^t | \cdots | \widehat{\boldsymbol{x}}_{t-1,i}^t | \boldsymbol{x}_{t,i}^t | \widehat{\boldsymbol{x}}_{t+1}^t | \cdots | \widehat{\boldsymbol{x}}_{T,i}^t])$
      Set $\boldsymbol{H}^t = [\boldsymbol{h}_i^t \cdots \boldsymbol{h}_{B^t}^t]^\top$, $\boldsymbol{K}^t = [\boldsymbol{k}_i^t \cdots \boldsymbol{k}_{B^t}^t]^\top$
      for $l = 1 \ldots L$ **do**
        if $l == 1$ **then**
          $\bar{\boldsymbol{h}}_{l,i}^t = \boldsymbol{h}_i^t$, $\bar{\boldsymbol{k}}_{l,i}^t = [\boldsymbol{h}_i^t | \boldsymbol{k}_i^t]$
        else
          $\bar{\boldsymbol{h}}_{l,i}^t = \boldsymbol{h}_{l-1,i}^t$, $\bar{\boldsymbol{k}}_{l,i}^t = [\boldsymbol{h}_{l-1,i}^t | \boldsymbol{k}_{l-1,i}^t]$
          $\boldsymbol{h}_{l,i}^t = \text{Shared\_Path}(\bar{\boldsymbol{h}}_{l,i}^t)$, $\boldsymbol{k}_{l,i}^t = \text{Task\_Specific\_Path}^t(\bar{\boldsymbol{k}}_{l,i}^t)$
        end if
      end for
      $\widehat{y}_i^t = g^t([\boldsymbol{h}_{L,i}^t | \boldsymbol{k}_{L,i}^t])$
    end for
  end for
  Compute integration loss: $\mathcal{L}_{\text{integ}} = \sum_{t=1}^{T} \sum_{i=1}^{B^t} l(y_i^t, \widehat{y}_i^t)$
  Compute orthogonal regularizer for features: $\mathcal{R}_{\text{orth}} = \sum_{t=1}^{T} \|(\boldsymbol{H}^t)^\top \boldsymbol{K}^t\|_F^2$
  Compute robust regularizer for imputation: $\mathcal{R}_{\text{imp}} = \sum_{t=1}^{T} \sum_{s \neq 0,t} \|\boldsymbol{\Theta}_{s,p,1}^t\|_F^2$
  Compute regularizater for redundancy: $\mathcal{R}_{\text{dr}} = \sum_{t=1}^{T} \sum_{l=1}^{L} \|(\boldsymbol{\Theta}_{c,l}^t)^\top \boldsymbol{\Theta}_{p,l,1:d_{c,l-1}^r}^t\|_F^2$

  Parameters update:
  $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \eta \nabla_{\boldsymbol{\Theta}}(\mathcal{L}_{\text{integ}} + \mathcal{R}_{\text{orth}} + \mathcal{R}_{\text{imp}} + \mathcal{R}_{\text{dr}})$
end while
Output: Learnt parameters $\boldsymbol{\Theta}$