Adaptive Group Personalization for Federated Mutual Transfer Learning

Haoqing Xu [1] **Dian Shen** [1 2] **Meng Wang** [3] **Beilun Wang** [1 *]

Abstract

Mutual transfer learning aims to improve prediction with knowledge from related domains. Recently, federated learning is applied in this field to address the communication and privacy concerns. However, previous clustered federated learning (CFL) solutions lack theoretical guarantee of learnability recovery and require time-consuming hyper-parameter tuning, while centralized mutual transfer learning methods lack adaptability to concept drifts. In this paper, we propose the Adaptive Group Personalization method (**AdaGrP**) to overcome these challenges. We adaptively decide the recovery threshold with a non-parametric method, *adaptive threshold correction*, for tuning-free solution with relaxed condition. Theoretical results guarantee the perfect learnability recovery with the corrected threshold. Empirical results show AdaGrP achieves 16.9% average improvement in learnability structure recovery compared with state-of-the-art CFL baselines.

1. Introduction

Mutual transfer learning (Cheng et al., 2020; Xu et al., 2022) is a learning paradigm in big data analysis. It aims to improve prediction performance by transferring useful knowledge among related domains. It is assumed in mutual transfer learning that data domains are clustered into subgroups, in which the domains share knowledge more efficiently, as a *learnability structure*. For example, in climate analysis (Vose et al., 2014), climate zones are formed by multiple climate divisions as a learnability structure. Specifically, assuming domain $\mathcal{D}_i$ has subgroup label k_i, i.e., $\mathcal{D}_i \in \mathcal{S}_{k_i}$, the response $\boldsymbol{y}_i$ of sample size n_i sampled from $\mathcal{D}_i$ follows a two-layer linear mixed-effects model as

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{\theta}_i + \boldsymbol{\varepsilon}_i, \\ \text{s.t. } \boldsymbol{\theta}_i &= \boldsymbol{\alpha}_{k_i} + \boldsymbol{u}_i, \ \ \mathcal{D}_i \in \mathcal{S}_{k_i}, \end{aligned} \tag{1}$$

where $\boldsymbol{X}_i \in \mathbb{R}^{n_i \times p}$, $\boldsymbol{Z}_i \in \mathbb{R}^{n_i \times q}$ are *global features* and *heterogeneous features*, and they correspond to parameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}_i$. Global parameter $\boldsymbol{\beta}$ is shared among all the domains, while heterogeneous parameter $\boldsymbol{\theta}_i$ contains two components: $\boldsymbol{\alpha}_k$ is only shared within the subgroup $\mathcal{S}_k$, and $\boldsymbol{u}_i$ represents the domain-specific random effects that cannot be transferred. The learnability structure $\mathcal{S} = \{\mathcal{S}_k, \forall k\}$ therefore reveals the underlying relationships between domains and helps accurate prediction. Typical applications include climate analysis (Salam & Salam, 2020; Cheng et al., 2020), healthcare analysis (Zhang et al., 2021; Li et al., 2022), and longitudinal data analyses (Hu et al., 2022).

The learnability structure $\mathcal{S}$ is usually unknown *a priori* in the aforementioned applications. In order to recovery the learnability structure $\mathcal{S}$ and improve prediction performance, parameters should be estimated jointly with data in different domains. Previous mutual transfer learning methods mainly focus on centralized computing scenario. Data of each domain is collected and sent to a central server for estimation. However, in the climate analysis applications (Vose et al., 2014), the data will contain more than 4 billion samples if collected per second. It is unaffordable to transfer such large data to a central server. In the healthcare analysis (Li et al., 2022; Luo et al., 2022), the data usually contain sensitive information of patients and clinics (Bonomi & Jiang, 2018; Janmey & Elkin, 2018). Leakage of these private data may cause serious ethic problems. Therefore, *Communication bottleneck* and *Privacy concerns* significantly hinder mutual transfer learning from being widely adopted in emerging applications.

Federated learning (FL) (McMahan et al., 2017; Konečnỳ et al., 2016) provides a reliable distributed learning framework to address these concerns. By transmitting parameter updates only, the server eliminates the heavy burden of communicating all the data. Meanwhile, privacy is preserved since raw data are not transmitted. Unfortunately, two challenges emerge in Federated Mutual Transfer Learning:

(1) Learnability Heterogeneity: Learnability structure not only reveals the relationship between domains but also im-

[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China [2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China [3]College of Design and Innovation, Tongji University, Shanghai, China. * Correspondence to: Beilun Wang <beilun@seu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

proves prediction performance. However, it is impossible to benefit from such structure if single-model FL methods is applied. Solution to estimate learnability structure $\mathcal{S}$ and parameters $\boldsymbol{\beta}$, $\boldsymbol{\alpha}_k$ simultaneously within the FL framework remains unexplored.

(2) Concept Drift: With FL framework applied, it is possible for mutual transfer learning to apply to much larger applications. Since FL framwork communicates with clients among a long range of time, *concept drift* problems (Hsieh et al., 2022) would arise, where the data distribution of domains $\mathcal{D}_i$ could have been changing due to user activities, especially in longitudinal data analyses (Hu et al., 2022). The previous stable distribution assumption in centralized mutual transfer learning would therefore no longer hold.

Possible straightforward solution of applying FL to mutual transfer learning is either to adapt previous centralized mutual transfer learning methods to an FL framework, or to apply previous work in FL with similar assumptions. Centralized mutual transfer learning methods estimate learnability structure $\mathcal{S}$ by evaluating the distances of domains. As a typical method, DiffS (Xu et al., 2022) recovers learnability structure via complete-linkage clustering with a fixed threshold based on their proposed domain metric. However, it cannot deal with concept drift challenge well due to its fixed threshold in the clustering. When the underlying distribution shifts, the previous threshold would possibly be either too large or too small for new data. Previous work in FL with similar assumptions includes Clustered Federated Learning (CFL) methods (Sattler et al., 2020a) from Personalized Federated Learning (Tan et al., 2022). These methods assume clients are partitioned into clusters. For example, IFCA (Ghosh et al., 2020) cluster clients by their local losses on different cluster models. FedDrift (Jothimurugesan et al., 2023) further addresses concept drift problems by adaptively creating and merging clusters. However, CFL methods focus on more general models so that they require tedious hyper-parameter tuning and lack theoretical guarantee in learnability structure recovery.

In this paper, we propose an Adaptive Group Personalization method for Federated Mutual Transfer Learning (**AdaGrP**) to overcome the above two challenges. We designed a nonparametric algorithm to correct the threshold of learnability structure recovery in each round, called *adaptive threshold correction*, in order to fit in with concept drifts. Based on such, AdaGrP utilizes a group personalization framework to aggregate the heterogeneous parameters from different subgroups and the global parameters simultaneously. We theoretically prove that the recovered learnability structure is perfect under a relaxed condition compared with previous work. Numerical results also indicate the outperformance of AdaGrP under concept drift environment. Our contribution is summarized as follows:

- **Novel method for federated mutual transfer learning:** We propose a novel method, AdaGrP, for mutual transfer learning under federated setting. We introduce the proposed *adaptive threshold correction* algorithm, to handle concept drift and leverage CFL framework to update the global and heterogeneous parameters simultaneously.
- **Tuning-free solution:** AdaGrP eliminates hyperparameter tuning work by utilizing the nonparametric adaptive threshold correction compared with common methods. It saves much time and effort when dealing with newly incoming data especially under concept drift setting.
- **Theoretical analysis:** We theoretically analyze the proposed AdaGrP in federated scenario. Results show that AdaGrP is able to perfectly recover the learnability structure in each communication round under a relaxed condition compared with previous work. AdaGrP is therefore ensured that it can keep capturing the drifting concepts during the federated learning procedure.
- **Synthetic and real-world experiments:** We conduct both synthetic and real-world experiments to compare the proposed AdaGrP with state-of-the-art baselines. Results show that AdaGrP outperforms significantly with about 96.32% learnability structure recovery accuracy compared with 91.52% of the best baseline. Results in NOAA nClimDiv dataset also show the utility of AdaGrP in real-world data.

Notations: Let bold lowercase characters like $\boldsymbol{\alpha}$ be vectors, bold uppercase characters like $\boldsymbol{A}$ be matrices, and calligraphic characters like $\mathcal{A}$ be sets. $[N] = \{1, \ldots, N\}$ is the set of natural numbers less than N. $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$. $\|\cdot\|$ is the ℓ_2 norm. The squared root $\boldsymbol{A}^{1/2}$ of a positive semi-definite matrix $\boldsymbol{A}$ is defined as $\boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2} = \boldsymbol{A}$. $\mathbf{1}_{p\times q}, \mathbf{0}_{p\times q} \in \mathbb{R}^{p\times q}$ denote $p \times q$ matrices with all entries filled with 1, or 0. $\boldsymbol{I}_q \in \mathbb{R}^{q\times q}$ denotes the $q \times q$ identity matrix. $\mathcal{I}(\cdot)$ refers to the indicator function that $\mathcal{I}(A) = 1$ iff A is true, otherwise 0.

2. Background

2.1. Problem Formulation

Consider a Federated Mutual Transfer Learning server with M clients which correspond to M domains $\{\mathcal{D}_i^{(\tau)}, i \in [M]\}$ respectively. At time step $\tau \in \mathbb{N}_+$, client i samples its private dataset $\{\boldsymbol{X}_i^{(\tau)}, \boldsymbol{Z}_i^{(\tau)}, \boldsymbol{y}_i^{(\tau)}\}$ from domain distribution $\mathcal{P}_{\mathcal{D}_i}^{(\tau)}$. Then the response $\boldsymbol{y}_i^{(\tau)}$ follows model (1) with an unknown learnability structure $\mathcal{S}^{(\tau)} = \{\mathcal{S}_k^{(\tau)}, k \in [K]\}$. Here, the domain-specific random effect $\boldsymbol{u}_i^{(\tau)} \sim \mathcal{N}(\mathbf{0}, \sigma_{u,(\tau)}^2 \boldsymbol{I})$,

Algorithm 1 Learnability Structure Recovery with Difference Standardization $\Psi(\boldsymbol{\theta}; \lambda)$

Input: Heterogeneous parameters $\boldsymbol{\theta}_i, \forall i$; Threshold λ;
$\boldsymbol{\Delta} \leftarrow (\Delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j))_{M\times M} := (\Delta_{ij})_{M\times M}$;
Initialize subgroups $\mathcal{S} = \{\{1\}, \{2\}, \ldots, \{M\}\}$;
for $m = 1$ to $M-1$ **do**
  if $\min_{i\neq j} \Delta_{ij} > \lambda$ **then**
    break;
  end if
  $u \leftarrow \arg\min_i \sum_{j\neq i} \mathcal{I}(\Delta_{ij} \leqslant \lambda)$ s.t. $\exists j \neq i, \Delta_{ij} \leqslant \lambda$;
  $v \leftarrow \arg\min_j \Delta_{uj}$;
  Add $\mathcal{S}_t = \mathcal{S}_u \cup \mathcal{S}_v$ to $\mathcal{S}$ and remove $\mathcal{S}_u, \mathcal{S}_v$;
  Insert a new row and column into $\boldsymbol{\Delta}$ indexed with t, where $\Delta_{t,j} = \max(\Delta_{u,j}, \Delta_{v,j}),\ \forall j \neq u, v, t$;
  Remove the rows and columns in $\boldsymbol{\Delta}$ of $\mathcal{S}_u, \mathcal{S}_v$;
end for
return $\mathcal{S}$;

and the observation noise $\boldsymbol{\varepsilon}_i^{(\tau)} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\varepsilon,(\tau)}\boldsymbol{I})$. Following previous work (Cheng et al., 2020; Xu et al., 2022), we further assume that σ_u and σ_ε are known here with consistent estimation of Restricted Maximum Likelihood method (Richardson & Welsh, 1994). Federated mutual transfer learning solves the generalized least square problem

$$\begin{aligned}\min_{\mathcal{S},\boldsymbol{\beta},\boldsymbol{\alpha}} \ \mathcal{L}^{(\tau)}(\mathcal{S}, \boldsymbol{\beta}, \boldsymbol{\alpha}) &= \sum_{i=1}^{M} \frac{n_i}{N} \ell_i^{(\tau)}(\boldsymbol{\beta}, \boldsymbol{\theta}_i), \\ &= \sum_{i=1}^{M} \frac{n_i}{N} \boldsymbol{r}_i^{(\tau)\top} \boldsymbol{W}_i^{(\tau)} \boldsymbol{r}_i^{(\tau)}, \\ \text{s.t. } \boldsymbol{\theta}_i = \boldsymbol{\alpha}_{k_i},\ &\text{where } \mathcal{D}_i^{(\tau)} \in \mathcal{S}_{k_i}, \forall i \in [M],\end{aligned} \tag{2}$$

where $\boldsymbol{r}_i = \boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{Z}_i\boldsymbol{\theta}_i$ is the residual of domain $\mathcal{D}_i$. $\boldsymbol{W}_i = (\sigma_\varepsilon^2\boldsymbol{I} + \sigma_u^2\boldsymbol{Z}_i\boldsymbol{Z}_i^\top)^{-1}$ is the inverse covariance of $\boldsymbol{y}_i$ given $\boldsymbol{X}_i$, $\boldsymbol{Z}_i$. Following (Jothimurugesan et al., 2023), we define that a *concept drift* occurs at time τ and at domain $\mathcal{D}_i^{(\tau)}$ when $\mathcal{P}_{\mathcal{D}_i}^{(\tau)} \neq \mathcal{P}_{\mathcal{D}_i}^{(\tau-1)}$. As a result, the global and heterogeneous parameters $\boldsymbol{\beta}^{(\tau)}$, $\boldsymbol{\alpha}_k^{(\tau)}$, and the learnability structure $\mathcal{S}^{(\tau)}$ would possibly change over time. The goal of federated mutual transfer learning is to estimate these during the communication rounds within time step τ.

2.2. Learnability Structure Recovery with Difference Standardization

Under the centralized assumption that the server has full accessibility to all the data, (Xu et al., 2022) proposes a difference standardization method, called DiffS, for fast and accurate estimation. The authors start with the minimizer of the local loss $\ell_i(\boldsymbol{\beta}, \boldsymbol{\theta})$ as

$$(\boldsymbol{\beta}_i^{\mathcal{D}}, \boldsymbol{\theta}_i^{\mathcal{D}})^\top = [\boldsymbol{G}_i^\top \boldsymbol{W}_i \boldsymbol{G}_i]^{-1} \boldsymbol{G}_i^\top \boldsymbol{W}_i \boldsymbol{y}_i \tag{3}$$

Here, $\boldsymbol{G}_i = (\boldsymbol{X}_i, \boldsymbol{Z}_i)$ is the concatenation of the feature matrices for notation simplification. The key idea of DiffS is to recover the learnability structure $\mathcal{S}$ with the initial estimator $\boldsymbol{\theta}_i^{\mathcal{D}}$ of heterogeneous parameter via the proposed complete-linkage algorithm Ψ as

$$\widehat{\mathcal{S}} = \Psi(\boldsymbol{\theta}^{\mathcal{D}}; \lambda), \tag{4}$$

as presented by Algorithm 1, where $\boldsymbol{\theta}^{\mathcal{D}} = \{\boldsymbol{\theta}_i^{\mathcal{D}}, \forall i\}$. Since $\boldsymbol{\theta}_i^{\mathcal{D}}$ follows a normal distribution asymptotically with center $\boldsymbol{\alpha}_{k_i}$ and random covariance, they argue that learnability structure could be recovered easier by *standardizing* the distances between $\boldsymbol{\theta}_i^{\mathcal{D}}$ as

$$\begin{aligned}\Delta(\boldsymbol{\theta}_i^{\mathcal{D}}, \boldsymbol{\theta}_j^{\mathcal{D}}) &= (\boldsymbol{\theta}_i^{\mathcal{D}} - \boldsymbol{\theta}_j^{\mathcal{D}})^\top (\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j)^{-1} (\boldsymbol{\theta}_i^{\mathcal{D}} - \boldsymbol{\theta}_j^{\mathcal{D}}), \\ &\sim \begin{cases} \chi_q^2, & k_i = k_j \\ \chi_q'^2(\|\boldsymbol{\mu}_{ij}\|^2), & k_i \neq k_j, \end{cases}\end{aligned} \tag{5}$$

where $\boldsymbol{\Sigma}_i = (\mathbf{0}_{q\times p}, \boldsymbol{I}_q)[\boldsymbol{G}_i^\top \boldsymbol{W}_i \boldsymbol{G}_i]^{-1}(\mathbf{0}_{q\times p}, \boldsymbol{I}_q)^\top$ is the covariance of $\boldsymbol{\theta}_i^{\mathcal{D}}$. χ_q^2, $\chi_q'^2(\mu)$ refer to centralized and non-centralized χ^2-distribution with q degrees of freedom and non-centrality parameter μ. $\boldsymbol{\mu}_{ij} = (\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j)^{-1/2}(\boldsymbol{\alpha}_{k_i} - \boldsymbol{\alpha}_{k_j})$ where k_i, k_j are the true subgroup labels of $\mathcal{D}_i, \mathcal{D}_j$.

Δ is then called the *standardized domain distance* and used in Ψ instead of the raw distances of $\boldsymbol{\theta}_i^{\mathcal{D}}$. We denote $\Delta_{ij}^{\mathcal{D}} = \Delta(\boldsymbol{\theta}_i^{\mathcal{D}}, \boldsymbol{\theta}_j^{\mathcal{D}})$ and similar notation is defined similarly hereafter. The threshold λ chosen in DiffS is *fixed* as $F_q^{-1}(0.99)$ where $F_q^{-1}(\cdot)$ is the inverse CDF of χ_q^2 to cover 99% of the $\Delta_{ij}^{\mathcal{D}}$ that $k_i = k_j$.

The authors proved theoretically that Algorithm 1 with standardized domain distance is able to perfectly recover the true learnability structure. However, according to their discussion in the paper, λ should be larger when dealing with large M datasets due to more possible outliers in the subgroups. Similarly, λ should be smaller with smaller M to avoid including too many inter-group $\Delta_{ij}^{\mathcal{D}}$ under the threshold. This indicates that the *fixed threshold* **cannot** well adapt to different data distribution, especially under concept drift problems that distribution may change rapidly.

3. Methodology

3.1. Adaptive Group Personalization for Federated Mutual Transfer Learning

Combine the ideas of CFL and centralized mutual transfer learning methods, we propose the Adaptive Group Personalization method for federated mutual transfer learning (**AdaGrP**). Assume there are R communication rounds in each time step, AdaGrP considers the optimization problem at the r-th communication round in time step τ as

$$\begin{aligned}\min_{\mathcal{S}^r,\boldsymbol{\beta}^r,\boldsymbol{\alpha}^r} \ &\mathcal{L}^{(\tau)}(\mathcal{S}^r, \boldsymbol{\beta}^r, \boldsymbol{\alpha}^r), \\ \text{s.t. } &\mathcal{S}^r = \Psi\left(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r)\right),\end{aligned} \tag{6}$$

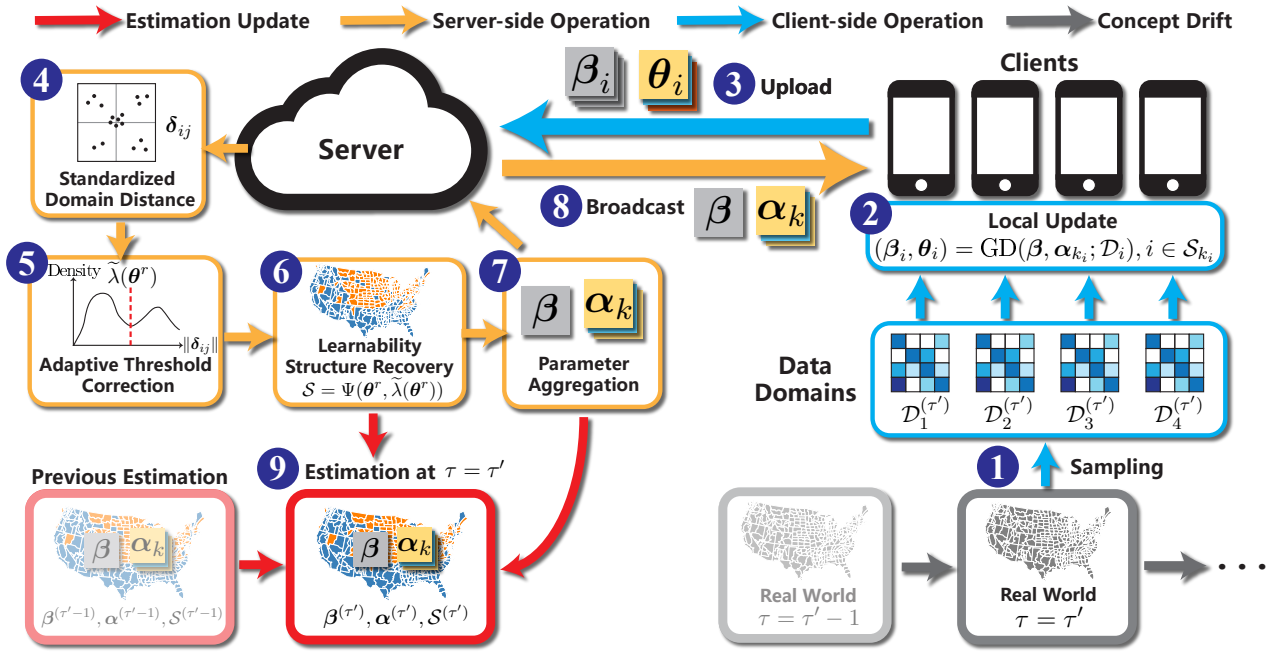


Figure 1. An illustration of the proposed AdaGrP. Generally, there are 8 steps in each communication round: (1) sample from corresponding data domains; (2) conduct local updates; (3) upload to server; (4) compute standardized domain distances; (5) **adaptive threshold correction** (Section 3.2); (6) learnability structure recovery; (7) parameter aggregation; (8) broadcast new parameters. These steps repeat for R times as R communication rounds in one time step. At the last round of a time step, AdaGrP proceeds to (9) estimation output.

where $\mathcal{S}^r$ denotes the learnability structure recovered by Algorithm 1 with adaptive threshold $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ in the r-th round. $\boldsymbol{\theta}^r = \{\boldsymbol{\theta}_i^r, \forall i\}$ and $\boldsymbol{\theta}_i^r$ denotes the $\boldsymbol{\theta}_i$ updated by client i at r-th round. To solve this problem, we illustrate the main steps of AdaGrP in Figure 1. Generally, AdaGrP follows the basic federated learning framework (step 1–3 & 8): clients conduct local updates with private data and only transfer parameters in communications. On server side, we divide the solution to problem (6) into two components: **1) learnability recovery** (step 4–6) and **2) parameter estimation** (step 7). For the learnability recovery part, we propose the **Adaptive Threshold Correction** (step 5) method to adaptively decide the threshold $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ based on $\boldsymbol{\theta}^r$. We present the detailed description of adaptive threshold correction in Section 3.2. For the parameter estimation part, we construct the whole **Group Personalization** algorithm of AdaGrP in Section 3.3 inspired by Group Personalized FL method (Liu et al., 2022b) that clients contribute their updates to their own subgroup.

3.2. Adaptive Threshold Correction

As mentioned in Section 2.2, a *fixed* threshold of Algorithm 1 **cannot** well adapt to concept drift environment. In order to figure out an *adaptive* threshold with theoretical guarantee, we propose the **adaptive threshold correction** algorithm Λ in Algorithm 2 and let the adaptive threshold $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ used in r-th round be

$$\widetilde{\lambda}(\boldsymbol{\theta}^r) = \Lambda(\boldsymbol{\Delta}(\boldsymbol{\theta}^r)), \tag{7}$$

where $\boldsymbol{\Delta}(\boldsymbol{\theta}^r) = \{\Delta(\boldsymbol{\theta}_i^r, \boldsymbol{\theta}_j^r), \forall i < j\} = \{\Delta_{ij}^r, \forall i < j\}$ is the standardized domain distances computed with $\boldsymbol{\theta}^r$. Algorithm 2 starts with a over-estimation of $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ as λ_{sup}. In practice, we set $\lambda_{\text{sup}} = F_q^{-1}(1 - 10^{-3})$ to cover most of the cases. In each iteration, the proportion p_χ of the Δ_{ij}^r less than the current $\widetilde{\lambda}$ is firstly updated. Then, the local density $\mathcal{P}_\Delta(\widetilde{\lambda})$ of the distances $\boldsymbol{\Delta}(\boldsymbol{\theta}^r)$ is estimated by the commonly used k nearest neighbor method (Zhao & Lai, 2022). Once detected that

$$\widehat{\mathcal{P}}_\Delta(\widetilde{\lambda}) < 2p_\chi \chi_q^2(\widetilde{\lambda}), \tag{8}$$

where $\chi_q^2(\cdot)$ is the PDF of χ_q^2, the algorithm returns the current $\widetilde{\lambda}$ as the $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ in this round. Otherwise, it moves $\widetilde{\lambda}$ to a smaller one and continues the iteration. Algorithm stops anyway when $\widetilde{\lambda} \leqslant \lambda_{\text{inf}}$ which is the preset minimum of $\widetilde{\lambda}(\boldsymbol{\theta}^r)$. We set $\lambda_{\text{inf}} = F_q^{-1}(0.9)$ in practice.

The motivation of Λ starts with the distribution of *standardized domain distance*. According to (5), the population of $\Delta_{ij}^{\mathcal{D}}$ follows a mixture of χ^2-distributions as

$$\mathcal{P}_\Delta(x) = p_\chi \chi_q^2(x) + \sum_{i<j} p_{ij} \chi_q'^2(x; \|\boldsymbol{\mu}_{ij}\|^2), \tag{9}$$

Algorithm 2 Adaptive Threshold Correction $\Lambda(\boldsymbol{\Delta})$

Input: Domain distances $\boldsymbol{\Delta} = \{\Delta_{ij}, \forall i < j \in [M]\}$; Maximum and minimum threshold $\lambda_{\sup}$, $\lambda_{\inf}$;
Sort $\boldsymbol{\Delta}$;
$\widetilde{\lambda} \leftarrow \lambda_{\sup}$;
$k_M \leftarrow \lceil \sqrt{M(M-1)/2} \rceil$;
while $\widetilde{\lambda} > \lambda_{\inf}$ **do**
  Update p_χ estimation with threshold $\widetilde{\lambda}$;
  $\widehat{\mathcal{P}}_\Delta(\widetilde{\lambda}) \leftarrow$ K-NEARESTNEIGHBORS$(\widetilde{\lambda}, \boldsymbol{\Delta}; k_M)$;
  if $\widehat{\mathcal{P}}_\Delta(\widetilde{\lambda}) < 2p_\chi \chi_q^2(\widetilde{\lambda})$ **then**
    break;
  else
    $\widetilde{\lambda} \leftarrow \max_{i<j} \Delta_{ij}$ s.t. $\Delta_{ij} < \widetilde{\lambda}$;
  end if
end while
return $\widetilde{\lambda}$.

K-NEARESTNEIGHBORS$(x, \boldsymbol{X}, k)$:
$N =$ # points in $\boldsymbol{X}$;
Find δ that $(x-\delta, x+\delta)$ contains exact k points in $\boldsymbol{X}$;
$\widehat{\mathcal{P}}(x) \leftarrow k/2N\delta$
return $\widehat{\mathcal{P}}(x)$.

where $\chi_q'^2(\cdot;\mu)$ is the PDF of $\chi_q'^2(\mu)$. p_χ, p_{ij} are the proportions of $\Delta_{ij}^{\mathcal{D}}$ that follows χ^2 or $\chi'^2(\|\boldsymbol{\mu}_{ij}\|^2)$), respectively. In fact, the threshold λ in Algorithm 1 is served for the classification of $\Delta_{ij}^{\mathcal{D}}$ to be regarded as following χ_q^2 or $\chi_q'^2(\cdot)$. In DiffS, the assumption is made to the data that,

Assumption 3.1 (Subgroup differentiation)**.** Denote $\mathcal{S}$ as the true learnability structure and $K = |\mathcal{S}|$. $\exists \lambda_-, \lambda_+$ $(0 < \lambda_- < \lambda_+)$ that satisfy $\forall k \in [K]$, $\forall i, j \in \mathcal{S}_k$, $\max_{i,j \in \mathcal{S}_k} \Delta_{ij}^{\mathcal{D}} < \lambda_-$, and $\forall k \in [K]$, $\exists i \in \mathcal{S}_k$, $\min_{j \notin \mathcal{S}_k} \Delta_{ij}^{\mathcal{D}} > \lambda_+$.

Since DiffS chooses the fixed threshold as $F_q^{-1}(0.99)$, where $F_q^{-1}(\cdot)$ is the inverse CDF of χ_q^2, Assumption 3.1 is actually with an implicit contraint that,

Condition 1 (DiffS constraint)**.** $\lambda_- \leqslant F_q^{-1}(0.99) \leqslant \lambda_+$.

However, due to the randomness of real-world data, $\boldsymbol{\mu}_{ij}$ varies dramatically that Condition 1 would possibly not hold. Under such situation, we propose to correct the threshold to a more proper one based on the estimated heterogeneous parameters $\boldsymbol{\theta}_i^r$ at round r, denoted as $\widetilde{\lambda}(\boldsymbol{\theta}^r)$, in order to guarantee the performance of learnability structure recovery.

Assume there is an oracle threshold decision method, to which $\boldsymbol{\mu}_{ij}$, p_{ij} is accessible a priori. The best choice λ^* of the threshold should satisfy that

$$\begin{aligned} p_\chi \chi_q^2(\lambda^*) &= \sum_{i<j} p_{ij} \chi_q'^2(\lambda^*; \|\boldsymbol{\mu}_{ij}\|^2) \\ \Rightarrow\ \mathcal{P}_\Delta(\lambda^*) &= 2p_\chi \chi_q^2(\lambda^*). \end{aligned} \tag{10}$$

Algorithm 3 AdaGrP at time step τ

Input: Number of clients M; Sample size of each domain n_i; Maximum communication round limit R; Number of local update steps T; Learning rate η;
if $\tau = 1$ **then**
  Initialize $\mathcal{S}^0 \leftarrow \{\{i\}, i \in [M]\}$;
  Initialize $\boldsymbol{\beta}^0$, $\boldsymbol{\alpha}_k^0, k \in [|\mathcal{S}^0|]$;
else
  $\mathcal{S}^0 \leftarrow \mathcal{S}^{(\tau-1)}$, $\boldsymbol{\beta}^0 \leftarrow \boldsymbol{\beta}^{(\tau-1)}$, $\boldsymbol{\alpha}_k^0 \leftarrow \boldsymbol{\alpha}_k^{(\tau-1)}$;
end if
for $r = 1$ to R **do**
  for $i \in [M]$ **client in parallel do**
    $k_i \leftarrow k$ such that $\mathcal{D}_i \in \mathcal{S}_k^{r-1}$;
    $\boldsymbol{\beta}_i^r, \boldsymbol{\theta}_i^r \leftarrow$ LOCALUPDATE$(\boldsymbol{\beta}^{r-1}, \boldsymbol{\alpha}_{k_i}^{r-1})$;
  end for
  $\Delta_{ij}^r \leftarrow \Delta(\boldsymbol{\theta}_i^r, \boldsymbol{\theta}_j^r), \forall i, j$;
  $\widetilde{\lambda}(\boldsymbol{\theta}^r) \leftarrow \Lambda(\boldsymbol{\Delta}(\boldsymbol{\theta}^r))$;
  $\mathcal{S}^r \leftarrow \Psi(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r))$;
  $\boldsymbol{\beta}^r, \boldsymbol{\alpha}^r \leftarrow$ AGGREGATION$(\boldsymbol{\beta}_i^r, \boldsymbol{\theta}_i^r, \mathcal{S}^r)$
end for
return $\mathcal{S}^R$, $\boldsymbol{\beta}^R$, $\boldsymbol{\alpha}_k^R, k \in [|\mathcal{S}^R|]$.

LOCALUPDATE$(\boldsymbol{\beta}, \boldsymbol{\theta}_i)$:
for $t = 1$ to T **do**
  $(\boldsymbol{\beta}, \boldsymbol{\theta}_i) = (\boldsymbol{\beta}, \boldsymbol{\theta}_i) - \eta \nabla \ell_i(\boldsymbol{\beta}, \boldsymbol{\theta}_i)$;
end for
return $\boldsymbol{\beta}$, $\boldsymbol{\theta}_i$.

AGGREGATION$(\boldsymbol{\beta}_i, \boldsymbol{\theta}_i, \mathcal{S})$:
$N = \sum_{i \in [M]} n_i$, $N_k = \sum_{\mathcal{D}_i \in \mathcal{S}_k} n_i$, $\forall k$;
$\boldsymbol{\beta} \leftarrow \sum_{i \in [M]} \frac{n_i}{N} \boldsymbol{\beta}_i$, $\boldsymbol{\alpha}_k \leftarrow \sum_{\mathcal{D}_i \in \mathcal{S}_k} \frac{n_i}{N_k} \boldsymbol{\theta}_i$, $\forall k$;
return $\boldsymbol{\beta}$, $\boldsymbol{\alpha}_k$, $\forall k$.

As an additional but reasonable assumption, we suppose $\lambda_- < \lambda^* < \lambda^+$. Thus, by setting λ^* as the threshold in Algorithm 1, Theorem 4.3 in (Xu et al., 2022) tells that it is able to recover the true learnability structure. Inspired by (10), we propose to estimate λ^* by nonparametric density estimation as described in Algorithm 2. Therefore, $\lambda_{\inf}$, $\lambda_{\sup}$ serves for a search range for λ^*. It can be detected that $\widetilde{\lambda}$ approaches λ^* by (10). Since we start with a overestimated $\widetilde{\lambda}$ and assume the PDFs are monotonic around λ^*, the criterion reduces to (8). Thus the returned $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ would be the best estimation of λ^*. We further theoretically guarantee the perfect recovery with the corrected threshold $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ in Section 4.2. We also show the detailed condition of AdaGrP is much relaxed compared with Condition 1. The whole algorithm of AdaGrP is described in the next section.

3.3. Group Personalization based Solution

AdaGrP solves (6) by **Group Personalization** and **Adap-**

tive Threshold Correction as presented in Algorithm 3. In the beginning, learnability structure is initialized with one client in each subgroup and parameters are initialized randomly. AdaGrP communicates with clients for R rounds at time step τ. At the r-th communication round, with previous learnability structure $\mathcal{S}^{r-1}$, clients start local update with the shared global parameter $\boldsymbol{\beta}^{r-1}$ and the heterogeneous parameter $\alpha_{k_i}^{r-1}$ of their corresponding subgroups. After local updates, standardized distances between domains Δ_{ij}^r are calculated with the updated heterogeneous parameters $\boldsymbol{\theta}^r$ and are further used in the **Adaptive Threshold Correction** algorithm $\Lambda(\boldsymbol{\Delta}(\boldsymbol{\theta}^r))$. New learnability structure $\mathcal{S}^r$ is then recovered by Algorithm 1 with the corrected threshold $\widetilde{\lambda}(\boldsymbol{\theta}^r)$. The aggregation of parameters is in a mixed **Group Personalization** manner. Global parameter $\boldsymbol{\beta}^r$ is aggregated just as FedAvg, while heterogeneous parameters $\boldsymbol{\alpha}_k^r$ are aggregated by subgroups. With true learnability structure, the aggregation of Algorithm 3 is identical to FedAvg framework. As we prove in Section 4.2 that the $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ corrected by Algorithm 2 is able to recover the true structure during communication, the convergence of AdaGrP is then guaranteed by the convergence analyses of FedAvg framework (Li et al., 2019; Wang & Joshi, 2021), which has been widely studied. When the last communication round ends, AdaGrP provides the final estimation of $\boldsymbol{\beta}^R$, $\boldsymbol{\alpha}^R$, $\mathcal{S}^R$ as the estimation of the current time step τ as $\boldsymbol{\beta}^{(\tau)}$, $\boldsymbol{\alpha}^{(\tau)}$, $\mathcal{S}^{(\tau)}$.

4. Theoretical Analysis

We theoretically analyze the proposed AdaGrP, especially the effectiveness of adaptive threshold correction. Firstly we show in Section 4.1 that adaptive threshold correction is able to find the best threshold λ^* defined by oracle decision (10). We further combine with the FL framework to derive the condition for AdaGrP to perfectly recover the learnability structure during communications in Section 4.2.

4.1. Threshold Correction in Centralized Setting

Assume we apply adaptive threshold correction (Algorithm 2) to decide the threshold in Algorithm 1 in a centralized learning framework, i.e., using $\boldsymbol{\theta}^{\mathcal{D}}$ instead of $\boldsymbol{\theta}^r$ to compute standardized domain distances. Denote the threshold corrected by Algorithm 2 as $\widetilde{\lambda}$, and the best threshold defined by (10) as λ^*. We analyze the learnability structure recovered by $\widetilde{\lambda}$ in the following condition and theorem.

Condition 2 (Centralized AdaGrP constriant)**.** $\lambda_- \leqslant \lambda_{\sup}$, $\lambda_+ \geqslant \lambda_{\inf}$, where $\lambda_{\inf}$, $\lambda_{\sup}$ are defined in Algorithm 2.

Theorem 4.1. *Assume Assumption 3.1 holds and $\lambda_- < \lambda^* < \lambda_+$. Denote the true learnability structure as $\mathcal{S}^*$. Under Condition 2, it satisfies that $\Psi(\boldsymbol{\theta}^{\mathcal{D}}; \widetilde{\lambda}) = \mathcal{S}^*$, where Ψ refers to Algorithm 1.*

Remark 4.2. See proof in Appendix A.1. The main idea is that λ^* can be consistently estimated by $\widetilde{\lambda}$ via the condition (10) used in Algorithm 2. By Theorem 4.3 in (Xu et al., 2022), it ensures $\widetilde{\mathcal{S}}$ is identical to the true $\mathcal{S}$. Note that the constraint here, i.e., Condition 2, is well relaxed compared with the implicit constraint in DiffS, i.e., Condition 1, so that AdaGrP is able to cover more cases of data in the estimation. Thus, the recovery ability of AdaGrP is guaranteed under centralized setting.

4.2. Recovery Ability in Federated Setting

In this section, we analyze the ability of learnability structure recovery of AdaGrP and answer the question that: could AdaGrP keep capturing the dynamic learnability structure in federated environment with concept drifts?

In the federated setting, local updates introduce uncertainty to the estimated parameters, leading to the uncertainty of λ^* estimation. We first derive the error bound of the standardized domain distance estimated during communication.

Lemma 4.3 (Error bound of standardized domain distance)**.** *Denote ω as the smallest eigenvalue among all the $\boldsymbol{G}_i^\top \boldsymbol{W}_i \boldsymbol{G}_i$, $\forall i$. The error between the true standardized domain distance $\Delta_{ij}^{\mathcal{D}}$ and the standardized domain distance Δ_{ij}^r calculated at r-th round after t local update steps in Algorithm 3 with learning rate $\eta < \frac{1}{2\omega}$ satisfies*

$$\mathbb{E}\left[\left|\Delta_{ij}^r - \Delta_{ij}^{\mathcal{D}}\right|\right] \leqslant C_{ij}(1-2\eta\omega)^{t/2}, \tag{11}$$

where $C_{ij} = 2C_0\|\boldsymbol{\Sigma}_{ij}^{-1/2}\|_2^2 + 4\sqrt{C_0}\|\boldsymbol{\Sigma}_{ij}^{-1}(\boldsymbol{\theta}_i^{\mathcal{D}} - \boldsymbol{\theta}_j^{\mathcal{D}})\|$ and $C_0 = \max_i \|\boldsymbol{\theta}_i^0 - \boldsymbol{\theta}_i^{\mathcal{D}}\|^2$.

See detailed proof in Appendix A.2. Denoting $E_t = \max_{i,j} \mathbb{E}\left[\left|\Delta_{ij}^r - \Delta_{ij}^{\mathcal{D}}\right|\right]$ and $C_M = \max_{i,j} C_{ij}$, we state the constraint of AdaGrP under FL setting as

Condition 3 (AdaGrP constraint)**.** The following condition holds in each rount r: (1) $\lambda_- + E_t \leqslant \lambda_{\sup}$; (2) $\lambda_+ - E_t \geqslant \lambda_{\inf}$; (3) $\eta < \frac{1}{2\omega}$; (4) $t > \frac{2(\ln(\lambda_+ - \lambda_-) - \ln 2C_M)}{\ln(1-2\eta\omega)}$.

Remark 4.4. According to Lemma 4.3, we know that Δ_{ij}^r converges linearly with gradient descent. Due to the error, the conditions in Condition 2 are tightened with E_t. We emphasize here that such tightening would vanish quickly with larger local steps t. The condition (4) is derived from the condition that $E_t < 1/2(\lambda_+ - \lambda_-)$ so that the interval (λ_-, λ_+) would not vanish and λ^* could still exist. For SGD, the learning rate, i.e., condition (3), should be changed to $1/\omega t$ and the convergence rate reduces to $O(1/t)$. Correspondingly, the condition (4) would be like $t > C'_M/(\lambda_+ - \lambda_-)^2$.

Considering recovering the learnability structure $\mathcal{S}^{(\tau)}$ at time step τ, we provide the following conclusion of the recovery ability of AdaGrP under federated setting.

Theorem 4.5. *Assume Assumption 3.1 holds and* $\lambda_- < \lambda^* < \lambda_+$. *Under Condition 3, AdaGrP satisfies that*

$$\Psi(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r)) = \Psi(\boldsymbol{\theta}^r; \lambda^*), \forall r \in [R]. \tag{12}$$

With sufficient local updates that $t > \frac{2 \ln C_\lambda / C_M}{\ln (1-2\eta\omega)}$,

$$\Psi(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r)) = \mathcal{S}^{(\tau)}, \forall r \in [R], \tag{13}$$

where $C_\lambda = \min(\lambda_+ - \lambda^*, \lambda^* - \lambda_-)$.

Remark 4.6. See proof in Appendix A.3. The theorem indicates that with sufficient local updates and acceptable tightening of Condition 2, the learnability structure could be perfectly recovered via the updated parameters $\boldsymbol{\theta}^r$ and the corrected threshold $\widetilde{\lambda}(\boldsymbol{\theta}^r)$. Since such perfect recovery is achievable in each communication round at any time step, an affirmative answer is thus obtained for the question in the beginning: AdaGrP is able to capture the dynamic learnability structure during the whole procedure under federated concept drift environment.

5. Related Work

Centralized Mutual Transfer Learning Centralized approaches for mutual transfer learning mainly try to learn the parameters with regularization or assumption in the loss functions. For example, some assume parameters are linear combinations of some latent or low-dimensional cluster centers (Han & Zhang, 2015). Different regularizers are also used in the previous work, such as Lasso (Tibshirani, 1996), Frobenious norm (Evgeniou & Pontil, 2004), and other common norms (Gong et al., 2012; Jalali et al., 2010). Chen et al. propose the state-of-the-art method called CD Fusion (Cheng et al., 2020) that leverage confidence distribution to accelerate the computation. Inspired by the confidence distribution, Xu et al. propose DiffS (Xu et al., 2022) in order to significantly reduce the computational complexity with standardized domain difference.

In centralized approaches, data are collected and sent to the central server for estimation. In such a scenario, the algorithms can easily achieve optimal solution. However, data may not available nowadays for those privacy-sensitive applications (Mothukuri et al., 2021; Liu et al., 2022a). Additionally, the upload of raw data costs much in communication between clients and especially the server. As the scale of data grows, communication bandwidth would become the bottleneck to transfer all the data.

Clustered Federated Learning Federated approaches with similar assumptions with mutual transfer learning are mainly clustered federated learning (CFL) methods. Clustered federated learning was first proposed by (Sattler et al., 2020a;b) in order to address the statistical heterogeneity problem (Sattler et al., 2019) in federated learning. CFL tries to split the clients based on their gradient directions to form multiple subgroups of clients, which is consistent with the definition of learnability structure in mutual transfer learning. The idea of clustering clients was further developed towards different directions. Some methods focus on improving the algorithm by applying conventional but simple clustering methods such as hierarchical clustering (Briggs et al., 2020), k-means (Long et al., 2023), or decentralized clustering strategy (Ghosh et al., 2020). Other authors put efforts on utilizing CFL in the scenario of personalized federated learning (PFL) (Tan et al., 2022). Group personalization (Liu et al., 2022b; Duan et al., 2021) was therefore proposed to study how the clustering strategy improves personlization performance.

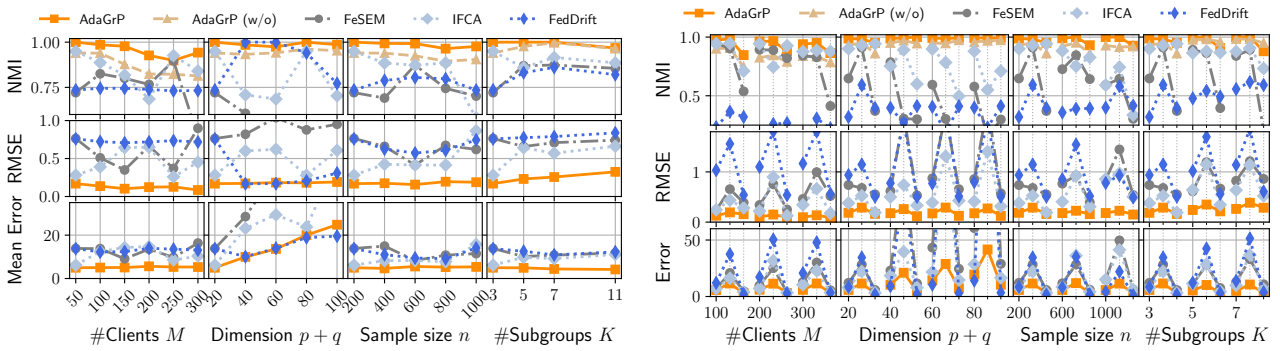
Although FL approaches address the communication concerns, drawbacks also exist. Vanilla CFL (Sattler et al., 2020a) bi-partition the clients at a time, which incurs inefficient computation costs. The followers (Ghosh et al., 2020; Long et al., 2023) handled the drawback via classical clustering methods but they need to specify the number of clusters ahead of learning. Group personalized methods even require detailed cluster structure before estimation. This hinders their easy use in federated mutual transfer learning where the true learnability structure are unknown.

6. Experiments

The following baseline methods are included in the comparisons: two SOTA CFL methods, IFCA (Ghosh et al., 2020), FeSEM (Long et al., 2023), the SOTA FL method for concept drift, FedDrift (Jothimurugesan et al., 2023). Also, we include a version of AdaGrP with fixed threshold $F_q^{-1}(0.99)$ as AdaGrP (w/o), as an ablation study. All the experiments were conducted on a Linux server with two Intel(R) Xeon(R) Gold 5117 CPUs and 256 GiB memory.

6.1. Experiments on Synthetic Data

Experiment Settings We generate synthetic data following (Xu et al., 2022; Cheng et al., 2020). Detailed generation method is described in Appendix B.1. We divide M clients into K subgroups as learnablity structure. Each client generates n samples as the design matrix $(\boldsymbol{X}_i, \boldsymbol{Z}_i)$. Data parameters $\{M, n, K, p, q\}$ are varied for comparisons in different data scales one at a time. The samples are divided with the proportion of $7:1:2$ for training, validation, and testing. We generate 5 replications for each setting of data parameters and report the average performance. For each method, we set the maximum number of communication rounds $R = 30$, the maximum number of local steps $T = 10000$. Other detailed settings are described in Appendix B.2. Clients early stops when validation error does not drop by 50 updates. Note that IFCA and FeSEM require preset number of clusters. We assume there is oracle for



(a) Changing Learnability Structure

(b) Varying Noise Level

Figure 2. Overall performance of the methods under concept drift environment. NMI, RMSE and Mean Error represent the performance of learnability structure recovery, parameter estimation and prediction. The data parameters are varied one at a time based on $M = 50$, $p = q = 10$, $n = 200$, $K = 3$. The three connected points of each case in (b) represents the values of three time steps respectively.

these methods and set as the true K.

Metrics We compare these methods in the following aspects: **(1) Learnabiltiy structure recovery ability:** We use normalized mutual information (NMI) (Ana & Jain, 2003; Vinh et al., 2009), a permutation-independent measurement of the agreement of two cluster assignments, to evaluate the learnability recovery ability for each method. Its value varies within $[0, 1]$ and the larger value indicates more agreement. **(2) Heterogenous parameter estimation error:** We use root mean square error (RMSE) to measure the error between the estimated heterogeneous parameters $\widehat{\boldsymbol{\theta}}_i$ and the true ones $\boldsymbol{\alpha}_{k_i}$. **(3) Prediction error:** Mean squared error is used to evaluate the prediction error of each method.

Results: (a) Changing learnability structure In order to evaluate the methods under concept drift environment, we set time step $\tau = 1, 2$ and change the learnability structure at $\tau = 2$. Specifically, the data of clients are shifted by one client along $[M]$, i.e., $k_i^{(2)} = k_{i+1}^{(1)}$. The results is shown in Figure 2a. AdaGrP (w/o) is omitted in RMSE and Error comparisons because it is very close to AdaGrP compared with others. The full figure is shown in Appendix B.3. Generally, AdaGrP (w/o) is slightly worse than AdaGrP due to its fixed threshold. The performances of the 3 baselines roughly arranged in the order of IFCA, FeSEM, FedDrift. Note that although both IFCA and FeSEM benefit from the oracle number of clusters, they still behave poorly in many cases due to lack of theoretical guarantee in recovering learnability structure. They may behave even worse with suboptimal number of clusters. Baselines show unstable performance when data setting varies, resulting in undulate metrics in Figure 2a. We notice that the results is similar with normal experiments without concept drift. As a result, we think the unstable performance is caused by their suboptimal recovery of learnability structure. AdaGrP is only affected heavier with varying dimensionality. This is reasonable since task complexity grows with dimensionality. Additionally, we notice that the performances on learnability structure recovery (NMI), parameter estimation (RMSE), and prediction (Error) are positively correlated. It demonstrates our claim that better recovery of learnability structure helps better parameter learning. In addition, we find it difficult for FedDrift to decide its hyper-parameter δ properly. We search from $\delta = \{0.01, 0.1, 1, 10, 100\}$ and only $\delta = 100$ provides reasonable learnability structure instead of one client in each subgroup. Notice that FedDrift has less error at $p = q = 50$ is because it creates 11.8 clusters on average instead of the true number 3.

Results: (b) Varying Noise Level As mentioned before, the population of standardized domain distance Δ_{ij} would be heavily affected by the dynamic environment. We simulate such variation by varying the noise level of the random effect $\boldsymbol{u}_i$ and $\boldsymbol{\varepsilon}$ to change $\boldsymbol{\mu}_{ij}$ in (5). In this experiment, we set time step $\tau = 1, 2, 3$ and change the noise level at each step. The noise levels σ_u, σ_ε are doubled in $\tau = 2$ and then quartered in $\tau = 3$. The results are shown in Figure 2b and the three points starting from each tick represent the value of three step respectively. We omit AdaGrP (w/o) as in Results (a). Generally, the concept drift problem becomes more serious and baselines all behave worse than changing learnability structure experiments. AdaGrP is hardly affected due to the tuning-free adaptive threshold correction strategy. FedDrift is affected the most and we notice that it could hardly recover the true learnability structure in the experiments. The estimated number of clusters by FedDrift varies from single digit to almost M (the number of clients), resulting in the poor performance. IFCA and FeSEM can provide reasonable estimates in the first time step ($\tau = 1$), while they are easy to fall in suboptimal cluster structure in the following time steps ($\tau = 2, 3$, concept drift occurs).

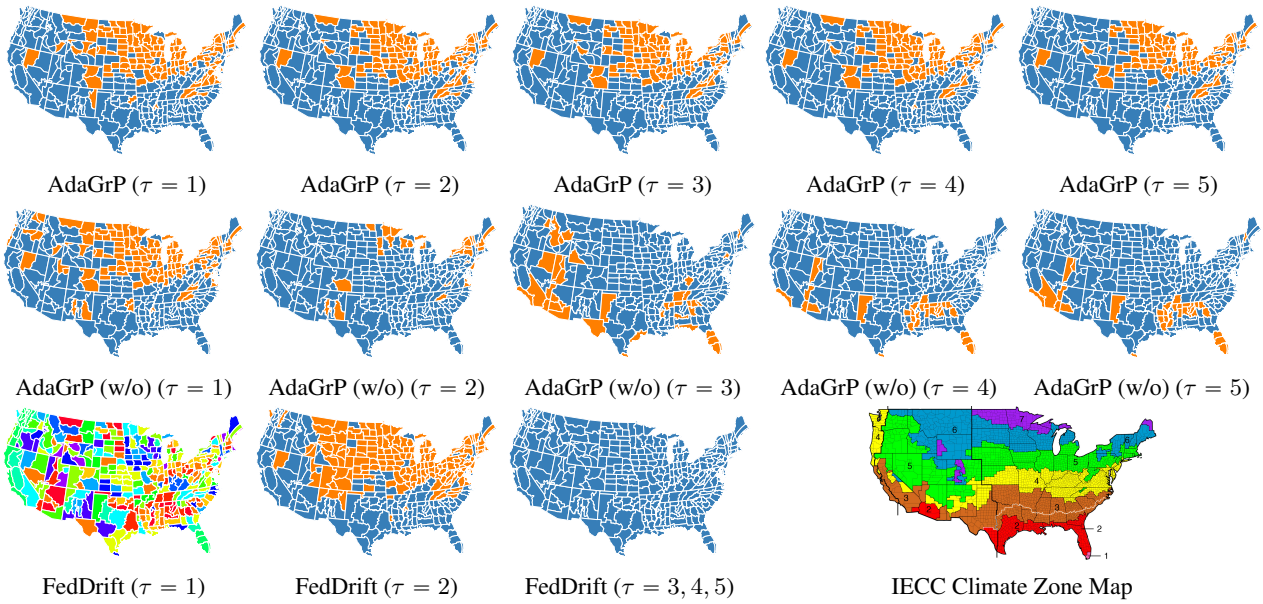


Figure 3. Learnability structure recovered during 5 time steps for the average temperature prediction task in NOAA nClimDiv dataset.

AdaGrP (w/o) drops more performance than AdaGrP does in most of the cases, indicating that adaptive threshold is able to handle the varying distribution of standardized domain distance.

6.2. Experiments on NOAA nClimDiv Database

We apply AdaGrP and the above baseline methods in the NOAA nClimDiv database (Vose et al., 2014) for the average temperature prediction task. The main task is to estimate the monthly average temperature based on 8 meteorological features. Palmer drought severity index (PDSI), Palmer hydrological drought index (PHDI), precipitation (PCPN) and Palmer Z index (ZNDX) are features collected in each data domain. Following (Cheng et al., 2020; Xu et al., 2022), three dummy variables, i.e., Summer (June, July, August), Fall (September, October, November) and Winter (December, January, February) are added. Another dummy variable Spring (March, April, May) feature is treated as the intercept term. The 3 heterogeneous features, i.e., intercept, PCPN and ZNDX, are selected by inspecting the kernel densities following (Cheng et al., 2020). To analyze the concept drift in the climate data, we split the data of 125 years into 5 periods, 25 years in one, to form 5 time steps, denoted as $\tau = 1, 2, 3, 4, 5$. We show the recovered learnability structures at the end of each time step in Figure 3. We only compare the three methods, AdaGrP, AdaGrP (w/o), and FedDrift, is because these methods can automatically figure out the number of clusters while the number of clusters in IFCA and FeSEM has to be tuned manually. We conduct hyper-parameter tuning for FedDrift from $\delta = \{0.01, 0.1, 1, 10, 100\}$. Only the last one $\delta = 100$ provides reasonable estimation and the other δ result in too many clusters (> 30). In the initial step ($\tau = 1$), both AdaGrP with or without adaptive threshold correction provide reasonable recovery of the learnability structure, while FedDrift estimates 18 clusters that are randomly distributed. In the time step 2, FedDrift manages to converge to a good estimation because local updates become stable. AdaGrP (w/o) almost discards its initial estimation and only leaves few divisions in one of the two clusters. In the following time steps ($\tau = 3, 4, 5$), the estimation of AdaGrP remains the outline of the initial one and only has several changing in the cluster identities of divisions. It is reasonable since climate zones would not change quickly. AdaGrP provides similar learnability structure compared with the climate zones defined by International Energy Conservation Code (IECC) (Council, 2012). Orange group roughly corresponds to the zone 6, 7 and the eastern part of zone 5. However, with fixed threshold, AdaGrP (w/o) has difficulty in distinguishing the subgroups. The clusters estimated by AdaGrP (w/o) is hard to explain and completely unbalanced. FedDrift is unable to figure out learnability structure in the beginning and eventually merges all the subgroups together. It may be explained by its sensibility to the threshold used in its merging strategy. With a large threshold, the cluster would tend to be merged into a single one, while with a small threshold, the new clusters would appear more frequently. This make FedDrift to require careful hyper-parameter tuning in practice. To compare with, AdaGrP gets rid of hyper-parameters and still provides valuable estimations. These results indicate the threshold correction method used by AdaGrP is not only able to capture real learnability structure hidden in the data but also easy to apply.

Acknowledgements

This work was supported by National Natural Science Foundation of China [Grant Numbers 61906040, 61972085, 62276063, 6509009710]; the Natural Science Foundation of Jiangsu Province [Grant Numbers BK20221457, BK20230083]; the Fundamental Research Funds for the Central Universities [Grant No. 2242021R41177]; and the National Key Research and Development Program of China [Grant No. 2022YFF0712400].

Impact Statement

In the paper, we focus on the federated mutual transfer learning tasks and propose AdaGrP as an adaptive and tuning-free solution. We expect that overall our method could help applications like IPD-MA, longitudinal data analysis, and climate analysis to be practically applied in FL framework. These tasks are naturally distributed and privacy-concerned. They could benefit from the efficient and privacy-preserved federated learning paradigm by leveraging AdaGrP without loss of adaptability and estimation accuracy. However, we believe it is important to ask the users' consent before using their private data in the mutual transfer learning, in order to avoid abusing or privacy invasion.

References

Ana, L. F. and Jain, A. K. Robust data clustering. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pp. II–II. IEEE, 2003.

Bonomi, L. and Jiang, X. Linking temporal medical records using non-protected health information data. *Statistical methods in medical research*, 27(11):3304–3324, 2018.

Briggs, C., Fan, Z., and Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2020.

Cheng, C.-W., Qiao, X., and Cheng, G. Mutual transfer learning for massive data. In *International Conference on Machine Learning*, pp. 1800–1809. PMLR, 2020.

Council, I. C. *2012 International Energy Conservation Code*. International Code Council, Inc., 2012.

Duan, M., Liu, D., Ji, X., Liu, R., Liang, L., Chen, X., and Tan, Y. Fedgroup: Efficient federated learning via decomposed similarity-based clustering. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 228–237. IEEE, 2021.

Evgeniou, T. and Pontil, M. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117, 2004.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33: 19586–19597, 2020.

Gong, P., Ye, J., and Zhang, C. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 895–903, 2012.

Han, L. and Zhang, Y. Learning multi-level task groups in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Hsieh, T., Arzani, B., and Mallick, A. Data drift mitigation in machine learning for large-scale systems, November 17 2022. US Patent App. 17/322,184.

Hu, S., Wang, Y.-G., Drovandi, C., and Cao, T. Predictions of machine learning with mixed-effects in analyzing longitudinal data under model misspecification. *Statistical Methods & Applications*, pp. 1–31, 2022.

Jalali, A., Sanghavi, S., Ruan, C., and Ravikumar, P. A dirty model for multi-task learning. *Advances in neural information processing systems*, 23, 2010.

Janmey, V. and Elkin, P. L. Re-identification risk in hipaa de-identified datasets: The mva attack. In *AMIA Annual Symposium Proceedings*, volume 2018, pp. 1329. American Medical Informatics Association, 2018.

Jothimurugesan, E., Hsieh, K., Wang, J., Joshi, G., and Gibbons, P. B. Federated learning under distributed concept drift. In *International Conference on Artificial Intelligence and Statistics*, pp. 5834–5853. PMLR, 2023.

Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

Li, W., Tong, J., Anjum, M. M., Mohammed, N., Chen, Y., and Jiang, X. Federated learning algorithms for generalized mixed-effects model (glmm) on horizontally partitioned data from distributed sources. *BMC Medical Informatics and Decision Making*, 22(1):269, 2022.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Liu, Z., Guo, J., Yang, W., Fan, J., Lam, K.-Y., and Zhao, J. Privacy-preserving aggregation in federated learning: A survey. *IEEE Transactions on Big Data*, 2022a.

Liu, Z., Hui, Y., and Peng, F. Group personalized federated learning. *arXiv preprint arXiv:2210.01863*, 2022b.

Long, G., Xie, M., Shen, T., Zhou, T., Wang, X., and Jiang, J. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023.

Luo, C., Islam, M. N., Sheils, N. E., Buresh, J., Schuemie, M. J., Doshi, J. A., Werner, R. M., Asch, D. A., and Chen, Y. dpql: a lossless distributed algorithm for generalized linear mixed model with application to privacy-preserving hospital profiling. *Journal of the American Medical Informatics Association*, 29(8):1366–1371, 2022.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

Richardson, A. and Welsh, A. H. Asymptotic properties of restricted maximum likelihood (reml) estimates for hierarchical mixed linear models. *Australian Journal of statistics*, 36(1):31–43, 1994.

Salam, A. and Salam, A. Internet of things for environmental sustainability and climate change. *Internet of Things for sustainable community development: Wireless communications, sensing, and systems*, pp. 33–69, 2020.

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020a.

Sattler, F., Müller, K.-R., Wiegand, T., and Samek, W. On the byzantine robustness of clustered federated learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8861–8865. IEEE, 2020b.

Tan, A. Z., Yu, H., Cui, L., and Yang, Q. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

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

Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pp. 1073–1080, 2009.

Vose, R. S., Applequist, S., Squires, M., Durre, I., Menne, M. J., Williams, C., and Arndt, D. Noaa's gridded climate divisional dataset (climdiv). *NOAA National Climatic Data Center*, 2014. doi: 10.7289/V5M32STR.

Wang, J. and Joshi, G. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *The Journal of Machine Learning Research*, 22 (1):9709–9758, 2021.

Xu, H., Wang, M., and Wang, B. A difference standardization method for mutual transfer learning. In *International Conference on Machine Learning*, pp. 24683–24697. PMLR, 2022.

Zhang, D. Y., Kou, Z., and Wang, D. Fedsens: A federated learning approach for smart health sensing with class imbalance in resource constrained edge computing. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2021.

Zhao, P. and Lai, L. Analysis of knn density estimation. *IEEE Transactions on Information Theory*, 68(12):7971–7995, 2022.

A. Proof of Theorems

A.1. Proof of Theorem 4.1

Proof. Consider the case that $q > 2$. Denote the mode of $\chi_q^2(x)$ as $m_\chi = \max(q-2, 0)$. Also denote the smallest mode of $\chi_q'^2(x; \|\boldsymbol{\mu}_{ij}\|^2)$ among i, j as $m_{\chi'}$. It can be inferred under Condition 2 that $m_\chi < \lambda^* < m_{\chi'}$ and with high possibility that $m_\chi < \lambda_-, m_{\chi'} > \lambda_+$. Due to the monotonicity of the χ^2 distributions, $\chi_q^2(x)$ decreases and $\sum_{i<j} \chi_q'^2(x; \|\boldsymbol{\mu}_{ij}\|^2)$ increases within $(m_\chi, m_{\chi'})$. As λ^* satisfies (10), thus $\forall \lambda \in (\lambda_-, \lambda^*)$,

$$\begin{aligned} & p_\chi \chi_q'^2(\lambda) < \sum_{i<j} p_{ij} \chi_q'^2(\lambda; \|\boldsymbol{\mu}_{ij}\|^2)) \\ \Rightarrow\ & \mathcal{P}_\Delta(\lambda) < 2 p_\chi \chi_q'^2(\lambda). \end{aligned} \tag{14}$$

Also $\forall \lambda \in (\lambda^*, \lambda_+)$,

$$\mathcal{P}_\Delta(\lambda) > 2 p_\chi \chi_q'^2(\lambda). \tag{15}$$

With the analysis of k nearest neighbor density estimation method (Zhao & Lai, 2022), it satisfies that

$$\mathbb{E}[\widehat{\mathcal{P}}_\Delta(\lambda)] = \mathcal{P}_\Delta(\lambda). \tag{16}$$

Here we assume $\lambda_{\sup}$ is not too large and $\lambda_{\inf}$ is not too small that points $\Delta_{ij}^{\mathcal{D}}$ within the search interval $(\lambda_{\inf}, \lambda_{\sup})$ would not account for much proportion. Then the estimation of $\mathcal{P}_\Delta(\lambda)$ and p_χ can be regarded consistent during the algorithm. In practice, we let $\lambda_{\sup} = F_q^{-1}(1 - 10^{-3})$ and $\lambda_{\inf} = F_q^{-1}(0.9)$. Given $\widetilde{\lambda}_-$ that triggers Algorithm 2 to break for satisfying

$$\widetilde{\mathcal{P}}_\Delta(\widetilde{\lambda}_-) < 2 p_\chi \chi_q'^2(\widetilde{\lambda}_-), \tag{17}$$

it can be inferred that $\widetilde{\lambda}_- \in (\lambda_-, \lambda^*)$ compared with (14) and based on the condition $\lambda_- \leqslant \lambda_{\sup}$. We denote the λ in the last iteration as $\widetilde{\lambda}_+$. Similarly,

$$\widetilde{\mathcal{P}}_\Delta(\widetilde{\lambda}_+) < 2 p_\chi \chi_q'^2(\widetilde{\lambda}_+), \tag{18}$$

which indicates that $\widetilde{\lambda}_+ \in (\lambda^*, \lambda_+)$ compared with (15) and based on the condition $\lambda_+ \geqslant \lambda_{\inf}$. Thus, according to the algorithm, $\widetilde{\lambda}_-$ and $\widetilde{\lambda}_+$ are two neighbor points selected from the sorted $\Delta_{ij}^{\mathcal{D}}$, choosing any threshold $\widetilde{\lambda} \in [\widetilde{\lambda}_-, \widetilde{\lambda}_+)$ has the same effect as threshold λ^*. According to Theorem 4.3 in (Xu et al., 2022), which we put here for reference,

Theorem A.1 (Learnability structure recovery guarantee)**.** *Denoting $\mathcal{S}^*$ as the true learnability structure, supposing that the Assumption 3.1 is satisfied and learnability structure $\widehat{\mathcal{S}}$ is recovered via Algorithm 1 with some threshold $\lambda \in (\lambda_-, \lambda_+)$, thus $\widehat{\mathcal{S}} = \mathcal{S}^*$.*

any threshold $\widetilde{\lambda} \in [\widetilde{\lambda}_-, \widetilde{\lambda}_+)$ would satisty the condition and would therefore guarantee the learnability structure recovery to be perfect. Since the $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ decided by Algorithm 2 is $\widetilde{\lambda}_-$ which is within the interval, Theorem 4.1 is thus proved. □

A.2. Proof of Lemma 4.3

In the following, we denote Ω as the largest eigenvalue among all the $\boldsymbol{G}_i^\top \boldsymbol{W}_i \boldsymbol{G}_i, i \in [M]$, which is

$$\Omega = \max_{i \in [M]} \rho\left(\boldsymbol{G}_i^\top \boldsymbol{W}_i \boldsymbol{G}_i\right), \tag{19}$$

and ω as the least, which is

$$\omega^{-1} = \max_{i \in [M]} \rho\left((\boldsymbol{G}_i^\top \boldsymbol{W}_i \boldsymbol{G}_i)^{-1}\right), \tag{20}$$

where ρ is the spectral radius.

Lemma A.2 (Ω-smoothness and ω-strongly convexity of local loss)**.** *Local losses $\ell_i(\boldsymbol{\beta}, \boldsymbol{\theta}), i \in [M]$ are all Ω-smooth and ω-strongly convex, which means $\forall \boldsymbol{w}_1 = (\boldsymbol{\beta}_1, \boldsymbol{\theta}_1), \boldsymbol{w}_2 = (\boldsymbol{\beta}_2, \boldsymbol{\theta}_2)$,*

$$\ell_i(\boldsymbol{w}_1) \leqslant \ell_i(\boldsymbol{w}_2) + (\boldsymbol{w}_1 - \boldsymbol{w}_2)^\top \nabla \ell_i(\boldsymbol{w}_2) + \frac{\Omega}{2} \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|^2, \tag{21}$$

$$\ell_i(\boldsymbol{w}_1) \geqslant \ell_i(\boldsymbol{w}_2) + (\boldsymbol{w}_1 - \boldsymbol{w}_2)^\top \nabla \ell_i(\boldsymbol{w}_2) + \frac{\omega}{2} \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|^2. \tag{22}$$

The proofs of the above lemma are trivial. Now we can prove Lemma 4.3.

Proof. Denote $\Delta\boldsymbol{\theta}^r = \boldsymbol{\theta}_i^r - \boldsymbol{\theta}_j^r$ and $\Delta\boldsymbol{\theta}^{\mathcal{D}} = \boldsymbol{\theta}_i^{\mathcal{D}} - \boldsymbol{\theta}_j^{\mathcal{D}}$. The expected error between the estimated standardized domain distance at round r and the true one is

$$\begin{aligned}
\mathbb{E}\left[|\Delta_{ij}^r - \Delta_{ij}^{\mathcal{D}}|\right] &= \mathbb{E}\left[|\Delta\boldsymbol{\theta}^{r\top}\boldsymbol{\Sigma}_{ij}^{-1}\Delta\boldsymbol{\theta}^r - \Delta\boldsymbol{\theta}^{\mathcal{D}\top}\boldsymbol{\Sigma}_{ij}^{-1}\Delta\boldsymbol{\theta}^{\mathcal{D}}|\right] \\
&= \mathbb{E}\left[|(\Delta\boldsymbol{\theta}^r - \Delta\boldsymbol{\theta}^{\mathcal{D}})^\top\boldsymbol{\Sigma}_{ij}^{-1}(\Delta\boldsymbol{\theta}^r + \Delta\boldsymbol{\theta}^{\mathcal{D}})|\right] \\
&= \mathbb{E}\left[|(\Delta\boldsymbol{\theta}^r - \Delta\boldsymbol{\theta}^{\mathcal{D}})^\top\boldsymbol{\Sigma}_{ij}^{-1}(\Delta\boldsymbol{\theta}^r - \Delta\boldsymbol{\theta}^{\mathcal{D}}) + 2\Delta\boldsymbol{\theta}^{\mathcal{D}\top}\boldsymbol{\Sigma}_{ij}^{-1}(\Delta\boldsymbol{\theta}^r - \Delta\boldsymbol{\theta}^{\mathcal{D}})|\right] \\
&\leqslant \mathbb{E}\left[\|\boldsymbol{\Sigma}_{ij}^{-1/2}\|_2^2\|(\Delta\boldsymbol{\theta}^r - \Delta\boldsymbol{\theta}^{\mathcal{D}})\|^2 + 2\|\boldsymbol{\Sigma}_{ij}^{-1}\Delta\boldsymbol{\theta}^{\mathcal{D}}\| \cdot \|(\Delta\boldsymbol{\theta}^r - \Delta\boldsymbol{\theta}^{\mathcal{D}})\|\right] \\
&\leqslant 2\|\boldsymbol{\Sigma}_{ij}^{-1/2}\|_2^2 \cdot \mathbb{E}\left[\|\boldsymbol{\theta}_i^r - \boldsymbol{\theta}_i^{\mathcal{D}}\|^2\right] + 4\|\boldsymbol{\Sigma}_{ij}^{-1}\Delta\boldsymbol{\theta}^{\mathcal{D}}\| \cdot \mathbb{E}\left[\|\boldsymbol{\theta}_i^r - \boldsymbol{\theta}_i^{\mathcal{D}}\|\right].
\end{aligned} \tag{23}$$

The next step is to bound the local update error $\mathbb{E}[\|\boldsymbol{\theta}_i^{\mathcal{D}} - \boldsymbol{\theta}_i^r\|]$. Note that $\boldsymbol{\theta}_i^{\mathcal{D}}$ is the minimizer of local loss, it is commonly known that with learning rate $\eta < 1/(2\omega)$, gradient descent guarantees linear convergence rate for the strongly convex loss ℓ_i of

$$\mathbb{E}\left[\|\boldsymbol{\theta}_i^t - \boldsymbol{\theta}_i^{\mathcal{D}}\|^2\right] \leqslant \|\boldsymbol{\theta}_i^0 - \boldsymbol{\theta}_i^{\mathcal{D}}\|^2(1 - 2\eta\omega)^t, \forall i \in [M]. \tag{24}$$

Denoting $C_0 = \max_i \|\boldsymbol{\theta}_i^0 - \boldsymbol{\theta}_i^{\mathcal{D}}\|^2$, the error (23) becomes

$$\begin{aligned}
\mathbb{E}\left[|\Delta_{ij}^r - \Delta_{ij}^{\mathcal{D}}|\right] &\leqslant 2C_0\|\boldsymbol{\Sigma}_{ij}^{-1/2}\|_2^2(1 - 2\eta\omega)^t + 4\sqrt{C_0}\|\boldsymbol{\Sigma}_{ij}^{-1}\Delta\boldsymbol{\theta}^{\mathcal{D}}\|(1 - 2\eta\omega)^{t/2} \\
&\leqslant \left(2C_0\|\boldsymbol{\Sigma}_{ij}^{-1/2}\|_2^2 + 4\sqrt{C_0}\|\boldsymbol{\Sigma}_{ij}^{-1}\Delta\boldsymbol{\theta}^{\mathcal{D}}\|\right)(1 - 2\eta\omega)^{t/2}.
\end{aligned} \tag{25}$$

For stochastic gradient descent, the convergence rate is changed to $O(1/t)$ with $\eta = 1/\omega t$ and the error bound becomes $O(1/\sqrt{t})$. The derivation is similar. □

A.3. Proof of Theorem 4.5

Proof. With $t > \frac{2(\ln(\lambda_+ - \lambda_-) - \ln 2C_M)}{\ln(1-2\eta\omega)}$ in Condition 3, according to Lemma 4.3,

$$\lambda_- + E_t < \lambda_+ - E_t. \tag{26}$$

Firstly, we consider the simple situation that $\lambda_- + E_t < \lambda^* < \lambda_+ - E_t$. Under Condition 3, we can denote that $\lambda'_- = \lambda_- + E_t$, $\lambda'_+ = \lambda_+ - E_t$. Therefore, $\lambda'_- < \lambda^* < \lambda'_+$ and Condition 2 is satisfied. According to the proof of Theorem 4.1, the recovered learnability structure is perfect as

$$\Psi(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r)) = \Psi(\boldsymbol{\theta}^r; \lambda^*) = \mathcal{S}^{(\tau)}, \text{ if } \lambda^* \in (\lambda_- + E_t, \lambda_+ - E_t). \tag{27}$$

Consider the situation that $\lambda^* \notin (\lambda'_-, \lambda'_+)$. Condition 2 is no longer satisfied and thus $\Psi(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r))$ is not guaranteed to be identical to $\mathcal{S}^{(\tau)}$. However, the corrected threshold $\widetilde{\lambda}(\boldsymbol{\theta}^r)$ is still the best estimation of λ^*. It can be proved similarly that

$$\Psi(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r)) = \Psi(\boldsymbol{\theta}^r; \lambda^*), \text{ if } \lambda^* \notin (\lambda_- + E_t, \lambda_+ - E_t). \tag{28}$$

To conclude, $\Psi(\boldsymbol{\theta}^r; \widetilde{\lambda}(\boldsymbol{\theta}^r)) = \Psi(\boldsymbol{\theta}^r; \lambda^*)$ under Condition 3 only.

Further consider the situation that $t > \frac{2\ln C_\lambda/C_M}{\ln(1-2\eta\omega)}$, where $C_\lambda = \min(\lambda_+ - \lambda^*, \lambda^* - \lambda_-)$. In this situation, $E_t < C_\lambda$ according to Lemma 4.3. Thus, $\lambda^* \in (\lambda_- + E_t, \lambda_+ - E_t)$ now and the proof reduces to the first situation. For SGD updater, the condition of local steps t would be changed to $O(1/C_\lambda^2)$, but the main conclusion is the same. □

B. Additional Details of Experiments

B.1. Data Preparation

we divide M clients into K subgroups as learnablity structure. K clients are first selected for each subgroup. The subgroup labels of the rest clients are randomly chosen. Each client generates n samples as the design matrix $(\boldsymbol{X}_i, \boldsymbol{Z}_i) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathcal{D}})$

where $\boldsymbol{\Sigma}_{\mathcal{D}} = 0.3 \cdot \mathbf{1}_{(p+q)\times(p+q)} + 0.7\boldsymbol{I}_{p+q}$. The response vector $\boldsymbol{y}_i$ is calculated via (1). $\boldsymbol{\beta}$ is sampled from $\mathcal{N}(\mathbf{0}, 16\boldsymbol{I}_p)$. In order to meet Assumption 3.1 and keep a proper signal-noise ratio, $\boldsymbol{\alpha}_k$ is obtained by shift the elements of $\boldsymbol{\alpha}_{k-1}$, e.g., $\boldsymbol{\alpha}_k = (\boldsymbol{\alpha}_{k-1}[2:], \boldsymbol{\alpha}_{k-1}[0])$. α_0 is $4\sigma_u^2\sqrt{q}$ times of the evenly spaced q points from $[-1, 1]$. The matrix of $\boldsymbol{\alpha}_k$ is then rotated by a random orthogonal matrix. The noise parameters are set as $\sigma_u^2 = 0.5$, $\sigma_\varepsilon^2 = 1$. The parameters M, K, n, p, and q are varied for comparisons in different data scales one at a time with the base case of $M = 50$, $K = 3$, $n = 200$, $p = q = 10$.

B.2. Supplement to Experiment Settings

- **Learning rate:** We conduct multiple tentative experiments to decide the optimal learning rate within $\{0.1, 0.01, 0.001, 0.0001\}$. For the synthetic data, we found that $\eta = 0.001$ has the best convergence rate and would not lead to NaNs. For the NOAA dataset, we choose $\eta = 0.005$ in order to slightly accelerate the learning.
- **Communication round:** In the experiment, we found that most of the method converges after about 5–10 rounds. We set the maximum of communication rounds as 30 in the synthetic experiments so that there would be 15 or 10 rounds in each time step in either experiment. For the NOAA dataset, we set 10 rounds per time step with a total of 50 communication rounds.

B.3. Detailed Results on Synthetic Dataset

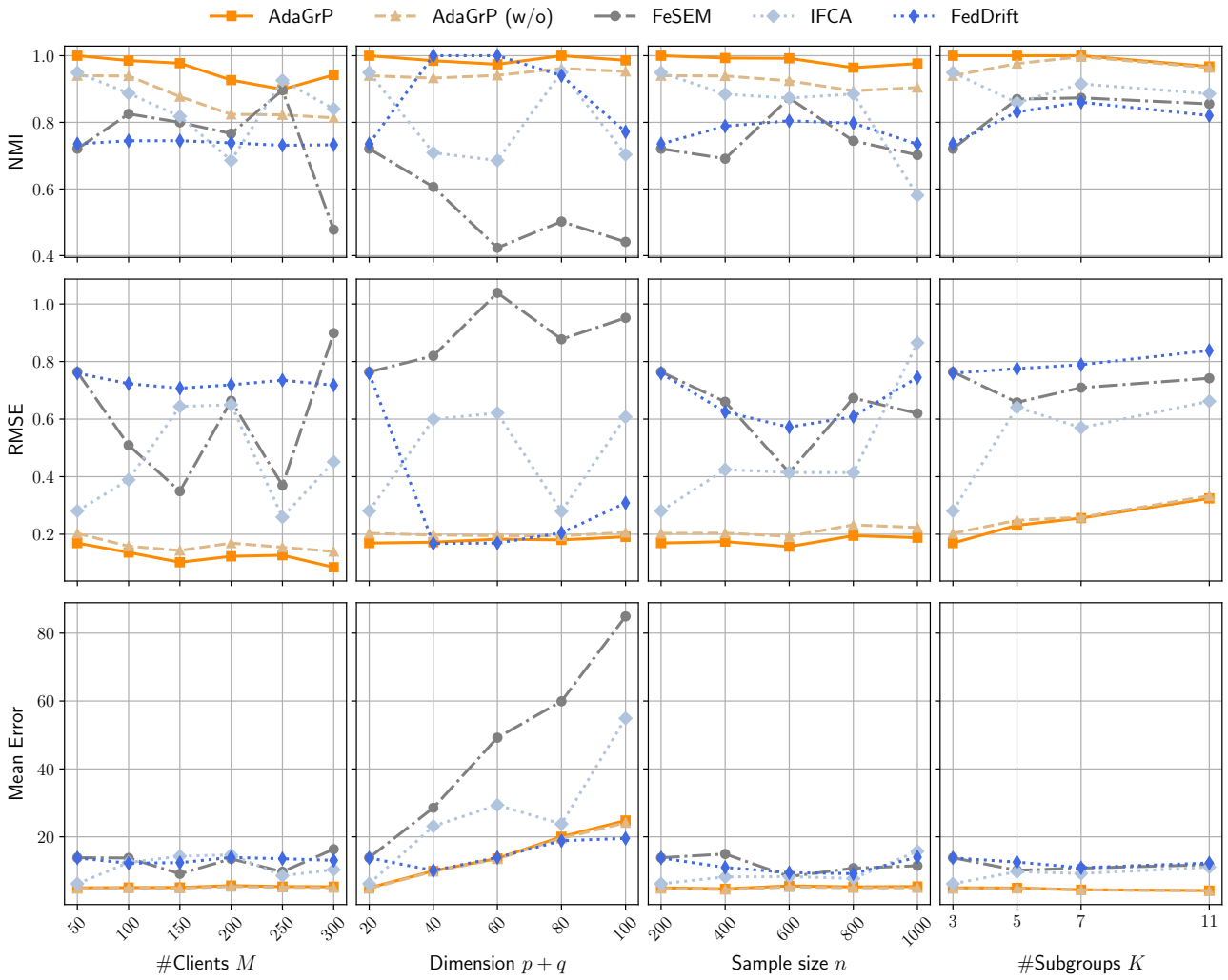


Figure 4. Full results for Changing Learnability Structure experiments.

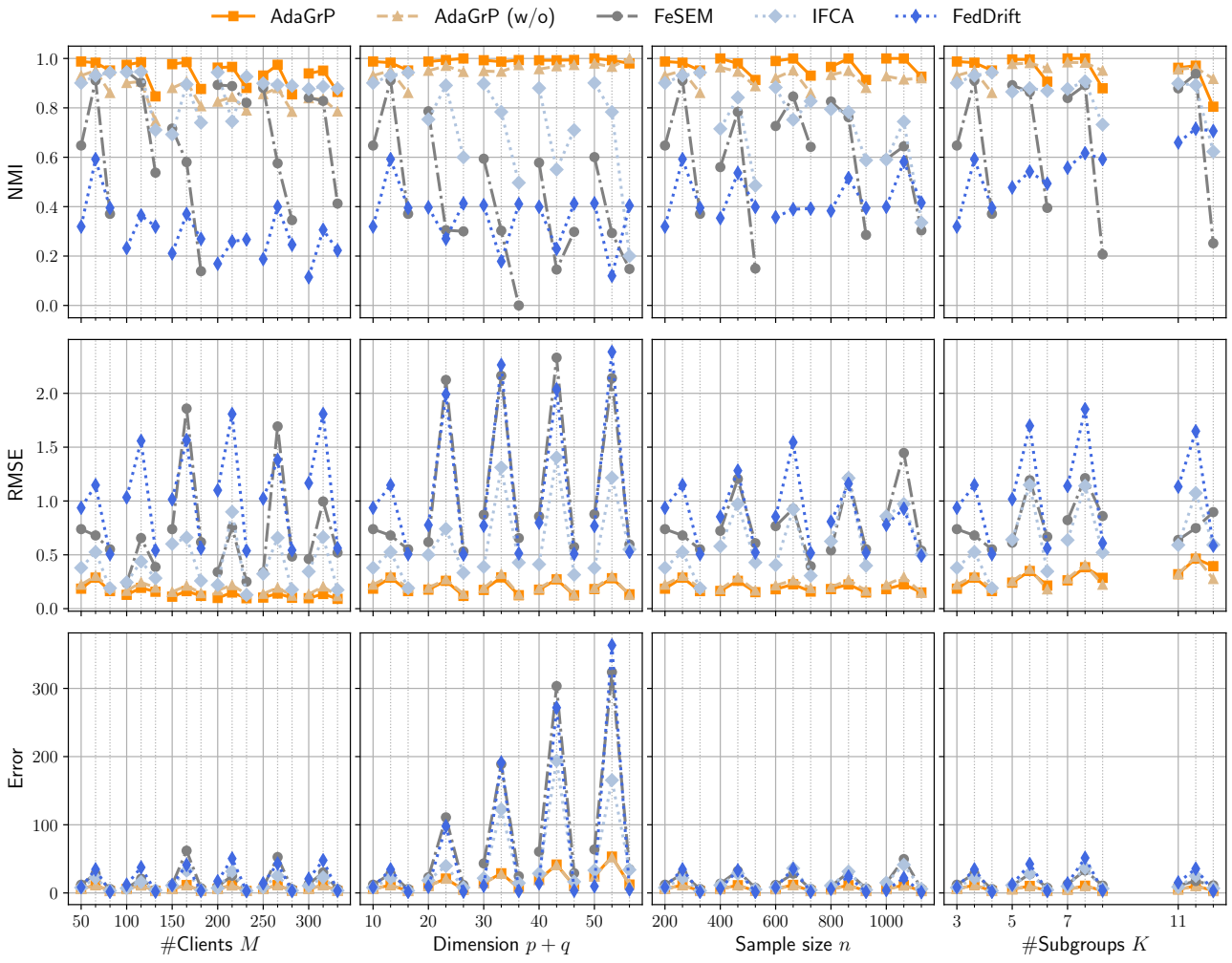


Figure 5. Full results for Varying Noise Level experiments.

C. Complexity Comparison

The computational complexity of Algorithm 2 is dominated by the sort of distances, which is $O(M^2 \log M)$. However, the total complexity of AdaGrP is dominated by the calculation of standardized domain difference with $O(M^2 q)$ instead of Algorithm 2. To compare with, previous centralized mutual transfer learning methods cost $O(M^2 q)$ for clustering. The common CFL methods (IFCA, FedDrift) cost $O(MKn^2(p+q))$ for clustering under mutual transfer learning task since they need to compute loss. We emphasize here that Algorithm 2 mainly runs on the server which has more adequate computation resources, while CFL methods require clients to compute their own cluster identities, leading to the scaling bottleneck.

D. Limitation Discussion

We consider the limitation of AdaGrP as follows:

- **Data Distribution:** Our method may fail with data that distribute completely differently from the linear mixed-effects model, e.g., natural language processing, recommendation systems tasks. These tasks are beyond mutual transfer learning and we suggest to address them with domain-specific solutions. **Data Distribution:** Our method may fail with data that distribute completely differently from the linear mixed-effects model, e.g., natural language processing, recommendation systems tasks. These tasks are beyond mutual transfer learning and we suggest to address them with domain-specific solutions.

- **Anomaly Detection:** In the paper, we assume all the concept drifts should be detected and the methods should react to them. However, it is possible that attackers or system failure (like anomalies) create fake concept drifts during the learning. A possible solution is to apply anomaly detection in AdaGrP and only react to true concept drifts to have more robustness against anomalies.
- **Model Averaging:** We use a simple model averaging strategy in AdaGrP as used in Group Personalized Federated Learning. In the aggregation, the clients' parameters are weighed by the local sample sizes. However, it is possible that clients with smaller data sizes should be weighed more due to their importance in the researches. Different model averaging ways could be further studied.