Bioinformatics, 2025, 1–10
doi: XXXXXXX.XXXXXXX
Advance Access Publication Date: Day Month Year
Paper

Privacy-Preserving Federated Unsupervised Domain Adaptation with Application to Age Prediction from DNA Methylation Data

Cem Ata Baykara [1,*] Ali Burak Ünal [1,2] Nico Pfeifer [2] and Mete Akgün [1,2]

[1]Medical Data Privacy and Privacy-Preserving Machine Learning, University of Tübingen, Tübingen, Germany and [2]Institute for Bioinformatics and Medical Informatics, University of Tübingen, Tübingen, Germany

*Corresponding author: cem.baykara@uni-tuebingen.de

Abstract

Motivation: Generalizing machine learning models across small, high-dimensional, and heterogeneous biological datasets remains a critical challenge due to domain shifts caused by variations in data collection, population differences, and privacy constraints that restrict data sharing. Existing federated domain adaptation (FDA) approaches primarily rely on deep learning and focus on classification tasks, making them unsuitable for privacy-sensitive, small-scale regression problems in biomedical research. We introduce a privacy-preserving federated method for unsupervised domain adaptation in regression, enabling robust learning across distributed, high-dimensional datasets while maintaining full data privacy.

Results: Our method is the first to enable distributed training of Gaussian Processes for domain adaptation, ensuring complete privacy through randomized encoding and secure aggregation. Unlike deep learning-based FDA approaches, our method is specifically designed for small-scale, high-dimensional biological data, overcoming prior limitations in scalability and generalization. We evaluate our approach on age prediction from DNA methylation data, demonstrating that it achieves performance comparable to non-private state-of-the-art methods while fully preserving data privacy. This work enables secure and effective cross-institutional collaboration in biomedical research without requiring raw data sharing.

Availability: The source code for our method is available at https://github.com/mdppml/FREDA.

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Introduction

Machine learning (ML) has rapidly become a powerful tool with applications across numerous fields, including computational biology and healthcare, where it has shown great potential in solving complex problems (Angermueller et al., 2016; Greener et al., 2022; Thumuluri et al., 2022; Valentini et al., 2009). However, collecting and labeling biological datasets is often challenging, costly, and time-consuming. As a result, many datasets in these fields are small-scale, unlabeled, and heterogeneous, often collected from different sources under varying environmental and experimental conditions—such as different laboratories, hospitals, or institutions (Orouji et al., 2024). These challenges introduce two critical issues: (1) data from different sources often exhibit distinct statistical distributions while lacking labeled samples, complicating direct model transfer; and (2) privacy regulations and the sensitive nature of biomedical data often restrict data sharing across institutions, necessitating collaborative learning approaches.

Unsupervised Federated Domain Adaptation (FDA) addresses these challenges by collaboratively aligning distributions between training and test data, referred to as source and target domains, respectively, without requiring direct data sharing (Farahani et al., 2021). The primary motivation for unsupervised FDA is the scarcity of labeled data in the target domain, making it impractical to train models from scratch. Most existing FDA methods aim to mitigate distributional differences between domains (Sun et al., 2016; Peng et al., 2019; Liu et al., 2024; Liang et al., 2021). While deep learning-based FDA methods have achieved success in computer vision (Ganin and Lempitsky, 2015; Long et al., 2016; Feng et al., 2021; Sener et al., 2016), their application to biological data remains limited due to high dimensionality and small sample sizes. Moreover, FDA research has predominantly

focused on classification tasks, while regression-based approaches remain significantly underexplored despite their importance in biomedical applications (Poplin et al., 2018; Lundberg et al., 2018; Li et al., 2022).

In this context, we introduce *freda* (**f**eder**a**t**e**d **d**omain **a**daptation), a novel method for privacy-preserving, federated unsupervised domain adaptation in regression tasks. Unlike conventional deep learning-based approaches that struggle with data scarcity and high dimensionality, *freda* is the first method to leverage distributed training of Gaussian Processes regressors (GPRs), enabling collaborating entities to model complex features without pooling their private data.

Gaussian Processes are particularly well-suited for feature modeling due to their probabilistic nature, providing not only point predictions but also uncertainty estimates in the form of Gaussian-distributed confidence intervals. This property is especially valuable in domain adaptation, where assessing prediction reliability is crucial when transferring knowledge across domains. However, like other kernel-based algorithms, GPRs require pairwise computation of data matrices, making it extremely challenging to train them when data is distributed across entities that cannot share raw samples.

To overcome this, *freda* introduces a novel combination of randomized encoding and secure aggregation, enabling distributed training of Gaussian Processes while preserving complete data privacy. By facilitating robust feature modeling without direct access to raw data, *freda* is particularly well-suited for biological datasets, where privacy constraints, limited sample sizes, and data heterogeneity pose significant challenges.

We evaluate *freda* on a challenging benchmark task of age prediction from DNA methylation data. Our results demonstrate that *freda* achieves performance comparable to non-private methods while preserving complete data privacy. By addressing the challenges of small-scale, heterogeneous, and privacy-sensitive regression problems, our approach significantly expands the applicability of domain adaptation to real-world biomedical applications. Our contributions are as follows:

- We propose *freda*, the first method to enable privacy-preserving, federated unsupervised domain adaptation for regression tasks, specifically designed for small-scale, high-dimensional biological datasets.
- Through a novel combination of randomized encoding and secure aggregation techniques, *freda* is the first method to enable the distributed training of GPRs for effective feature modeling while ensuring complete data privacy.
- We evaluate *freda* on the challenging task of age prediction from DNA methylation data, demonstrating that it effectively models complex feature relationships in small-scale, heterogeneous, and distributed biological datasets, achieving performance comparable to non-private approaches while preserving privacy.

Related Work

Unsupervised domain adaptation has been widely studied, primarily in image classification, where abundant labeled data and low-dimensional features facilitate domain transfer (Yue et al., 2023; Wang and Deng, 2018; Weng et al., 2023). Many approaches focus on aligning feature representations across domains using adversarial training, such as Domain-Adversarial Neural Networks (Ganin et al., 2016) and Maximum Mean Discrepancy-based methods like Deep Adaptation Networks (Long et al., 2015). Self-supervised learning techniques, including CDAN (Long et al., 2018), further integrate task-specific predictions to improve adaptation. While these methods perform well on image benchmarks, they are less suited to high-dimensional, small-sample biological datasets.

Domain Adaptation for Regression

Domain adaptation for regression is less explored due to the complexity of aligning continuous output spaces. DINO (Wu et al., 2022) leverages distribution-aware neural networks to mitigate distribution shifts in regression tasks, achieving strong performance in image-based settings with large datasets. However, it remains untested on high-dimensional, low-sample-size biological data. A key method for unsupervised adaptation in this setting is *wenda* (Handl et al., 2019), which estimates feature dependencies and applies adaptive regularization to handle domain shifts. Handl et al. (2019) demonstrated its effectiveness in DNA methylation-based age prediction, making it a relevant baseline for evaluating *freda*.

Federated Domain Adaptation

Federated domain adaptation extends domain adaptation to distributed settings, addressing both domain shifts and privacy constraints. Methods such as PartialFed (Sun et al., 2021) dynamically mix global and local model parameters, improving performance on cross-domain classification tasks. Similarly, FedGP (Jiang et al., 2024) enhances adaptation in low-data scenarios by filtering noisy gradients and optimally combining source and target information. However, these approaches focus on image datasets with abundant samples and low-dimensional features, making their applicability to high-dimensional biological data uncertain.

The Freda Method

We consider the following distributed setting: there are N source domain clients, each with a local labeled dataset $X^{s_i} = \{(x_j^{s_i}, y_j^{s_i})\}_{j=1}^{n_i}$ and a sample size of n_i. The entire source domain data, distributed across the N clients, is denoted as $X^S = \bigcup_{i=1}^{N} X^{s_i}$. Similarly, there is a target client with a dataset $X^T = \{x_m^t\}_{m=1}^{n_t}$ containing n_t samples for the same prediction task, but without any available labels. In both the source and target domain datasets, the samples $\{x_j^{s_i}\}$ and $\{x_m^t\}$ are $\mathcal{P}$-dimensional vectors, where $\mathcal{P} \in \mathbb{N}$, and the labels $\{y_j^{s_i}\}$ are scalars. The goal is to leverage source domain data to train a model that generalizes well to the target domain without explicit data sharing.

Freda follows four key steps: (1) learning feature dependencies via federated feature models, (2) computing confidence scores to derive feature weights, (3) predicting optimal regularization parameters through federated training, and (4) training the final adaptive model. While we describe the method for a single target domain, *freda* can be extended to multiple target domains as demonstrated in Section 4. A high-level summary of the algorithm as well as a detailed security analysis of the protocol, including communication assumptions and privacy guarantees under a semi-honest model, is provided in the supplementary material.

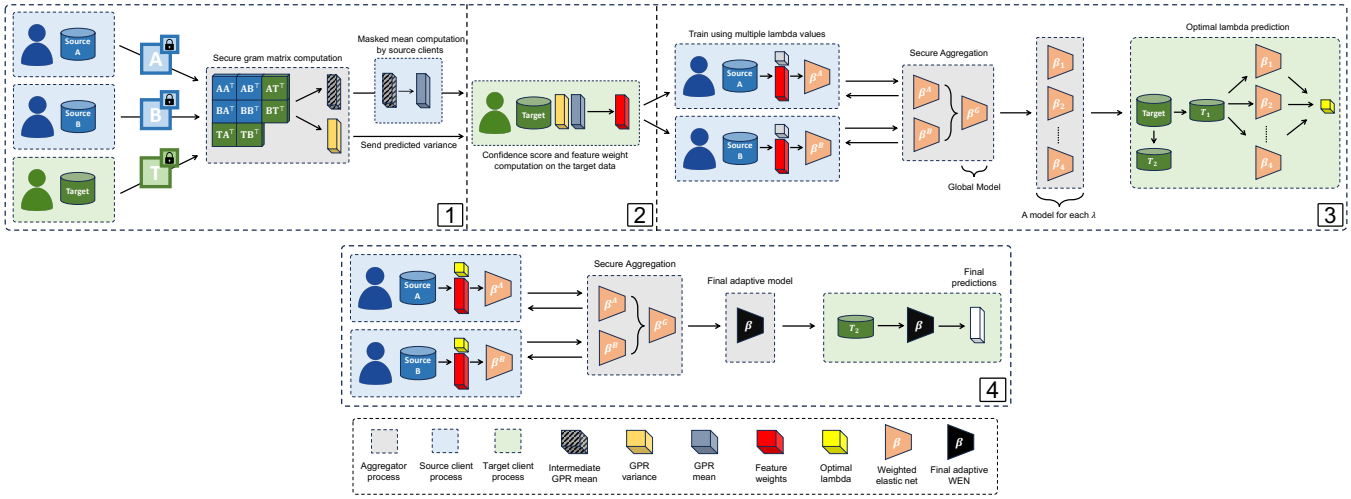

Fig. 1: Overview of the *freda* framework for unsupervised domain adaptation in a federated setting. The framework consists of four phases: (1) **Feature Model Training**, where each feature is modeled using federated hyper-parameter optimization and GPRs with secure aggregation; (2) **Feature Weight Computation**, where the target client computes confidence scores from the predicted feature distributions and converts them into weights, which are shared with source clients via an aggregator; (3) **Optimal Lambda Prediction**, where multiple weighted elastic nets with varying regularization parameters (λ) are trained, and the optimal λ values are selected by the target client and shared with all participants; and (4) **Final Adaptive Model Training**, where the final adaptive WEN models are trained federatively and sent to the target client for inference.

Feature Model Training

Following the motivation of Handl et al. (2019), we begin by modeling dependencies between features. In the absence of target labels, this approach allows us to estimate how well each feature can be explained by the others, based on patterns learned from the source domain. The intuition is that features with stable dependency structures across domains are more likely to generalize and should be weighted more heavily in the final regression model.

In our distributed setting, we must compute these feature dependencies without sharing raw data between parties. To achieve this, we leverage Bayesian models, specifically GPRs, to model the conditional distribution of each feature f given all other features. For each feature, we train a separate GPR model g_f using the source domain inputs distributed across the participating clients.

We emphasize that these GPR models are not used for final prediction. Instead, they serve as intermediate models to compute confidence-based feature weights, which are later used in the federated training of the final adaptive weighted elastic net model.

The training data for a given feature f includes $X^S_{\neg f}$, the entire source domain data with the column corresponding to feature f removed, as inputs, and X^S_f, the corresponding feature vector, as labels. For new data points $X^T_{\neg f} = [x^t_{1,\neg f}, \ldots, x^t_{n_t,\neg f}]$, the target domain data with the column for feature f removed, the goal is to predict the corresponding feature vector $X^t_f = [x^t_{1,f}, \ldots, x^t_{n_t,f}]$.

For a specific source domain i, the vector containing feature f is denoted $X^{s_i}_f = [x^{s_i}_{1,f}, \ldots, x^{s_i}_{n_i,f}]$, while the $(n_i \times (\mathcal{P}-1))$-matrix of remaining features is $X^{s_i}_{\neg f} = [x^{s_i}_{1,\neg f}, \ldots, x^{s_i}_{n_i,\neg f}]$. Thus, for a given feature f, the GPR model g_f provides a closed-form predictive distribution:

$$g_f(X^T_{\neg f}) \sim \mathcal{N}(K_* K^{-1} X^S_f, K_{**} - K_* K^{-1} K_*^\top) \tag{1}$$

where:

$$\begin{aligned} K &= k(X^S_{\neg f}, X^S_{\neg f}) + \sigma^2_\epsilon \mathbb{1}_{n_S} \\ K_* &= k(X^S_{\neg f}, X^T_{\neg f}) \\ K_{**} &= k(X^T_{\neg f}, X^T_{\neg f}) \end{aligned} \tag{2}$$

.Here, $k(.,.)$ computes the linear kernel with the variance of the prior on the coefficients σ^2_k between the given input matrices (Williams and Rasmussen, 2006). Additionally, n_S denotes the total number of samples in the entire source domain dataset X^S. Unlike traditional supervised regression models that predict a single value for a given input, GPRs provide a full predictive distribution as output (Seeger, 2004), which we later use to compute feature weights. This GPR model involves two hyper-parameters that must be optimized for the best performance: the variance of the kernel, σ^2_k, and the variance of the additive Gaussian noise, σ^2_ϵ, from the closed-form solution in Eq. 1. The optimal values of these hyper-parameters are determined by maximizing the marginal likelihood for each feature. For a specific source client i, and the covariance matrix $K = k(X^{s_i}_{\neg f}, X^{s_i}_{\neg f}) + \sigma^2_\epsilon \mathbb{1}_{n_i}$ from Eq. 2, source client i maximizes:

$$\log \mathcal{L}(X^{s_i}_f | X^{s_i}_{\neg f}) = -\frac{1}{2}(X^{s_i}_f)^\top K^{-1} X^{s_i}_f - \frac{1}{2}\log|K| - \frac{n_i}{2}\log(2\pi) \tag{3}$$

Training feature models is straightforward when both target and source domains are accessible simultaneously. However, significant challenges arise when these datasets are distributed.

The first challenge is that, if the source domain is distributed across multiple entities, the optimization of hyper-parameters shown in Eq. 3 cannot be performed across the entire source domain. The second, and more complex, challenge is that due to the distribution of the source and target domains, the closed-form solution of the GPR model (as shown in Eq. 1) cannot

be computed directly. Since GPRs are non-parametric machine learning algorithms, obtaining predictions requires computing the three matrices in Eq. 2, namely K, K_*, and K_{**}.

In our setting, computing K_{**} is straightforward and can be performed locally by the owner of the target domain, as it requires only the target domain data. However, computing K and K_* as well as the predictive mean of the feature model $K_* K^{-1} X_f^S$ presents significant challenges. Computing K is challenging because, although it only requires source domain data, the data is distributed across multiple entities, and the entire matrix product of K cannot be computed trivially. Instead, it must be computed collaboratively among all source domain owners while preserving privacy. Similarly, computing K_* is challenging because it requires access to both source domain data and target domain data. Lastly, computing $K_* K^{-1} X_f^S$ is not straightforward because the feature column for the entire source domain data, X_f^S, is distributed between N source clients. Moreover, explicitly sharing X_f^S would compromise privacy since the aggregator could reconstruct the data of all source clients after the feature model training phase is completed, as each feature is modeled independently.

Freda addresses all of these challenges by employing secure aggregation and a customized masking scheme for matrix product computation (Hannemann et al., 2023).

Federated Hyper-Parameter Optimization

To optimize the GPR model hyper-parameters σ_k^2 (prior variance) and σ_ϵ^2 (noise variance) in a federated setting, *Freda* employs secure aggregation via zero-sum masking (Bonawitz et al., 2016). This method ensures privacy by having each client mask its data with random values that cancel out when aggregated, revealing only the global sum without exposing individual contributions. Each source client locally optimizes σ_k^2 and σ_ϵ^2 by maximizing the marginal likelihood over its dataset. These values are then securely aggregated to compute global averages, effectively approximating joint optimization over the entire source domain.

Federated GPR Computation

In the GPR model prediction process, the most challenging components to compute are K, K_*, and their product with the global feature vector X_f^S, as these require access to both the source domain data and the target domain data to calculate the matrix product. A naive plaintext approach would compromise privacy. To overcome this, we utilize a framework for secure and private matrix product computation (Hannemann et al., 2023), which allows us to calculate the product of matrices from the source and target domains without disclosing their plaintext values.

Privacy-Preserving Masking Process. We use special masking matrices to hide the input matrices of the matrix product and reveal only the result of this multiplication to the aggregator. This protects the privacy of the input matrices as well as their dimensionality. Specifically, both the source clients and the target client share a common seed to generate a shared mask matrix $M \in \mathbb{R}^{d \times \mathcal{P}}$, where d is a higher-dimensional space than the original feature space. To ensure that a left inverse exists, M is required to be full column rank. Each client p then locally computes a left inverse $L^p \in \mathbb{R}^{\mathcal{P} \times d}$ such that $L^p M = I_{\mathcal{P}}$, and applies the mask to its data:

$$\tilde{X}^p = X^p L^p (M M^\top)^{1/2}, \tag{4}$$

where X^p represents the local dataset of client p ($X^p = X^{s_i}$ for a source client or $X^p = X^T$ for the target client). The masked data $\tilde{X}^p$ is then sent to the aggregator.

Secure Gram Matrix Computation. The aggregator computes the Gram matrix for all clients using the masked data. For a pair of clients p and q, the Gram matrix is computed as:

$$\begin{aligned} \tilde{X}^p (\tilde{X}^q)^\top &= (X^p L^p (M M^\top)^{1/2})(X^q L^q (M M^\top)^{1/2})^\top \\ &= X^p L^p (M M^\top)^{1/2} ((M M^\top)^{1/2})^\top (L^q)^\top (X^q)^\top \\ &= X^p (L^p M)(M^\top (L^q)^\top)(X^q)^\top \\ &= X^p X^{q\top} \end{aligned} \tag{5}$$

where the masking terms cancel out, revealing only the product $\tilde{X}^p (\tilde{X}^q)^\top$ without exposing the individual data of the clients.

Using the computed Gram matrix G^{pq}, the aggregator calculates K and K_* as follows:

$$\begin{aligned} K &= \sigma_k^2 G^{pq} + \sigma_\epsilon^2 \mathbb{1}_{n_S} \\ K_* &= \sigma_k^2 G^{pq} \end{aligned} \tag{6}$$

Computing the Predicted Mean. Computing the predicted mean $K_* K^{-1} X_f^S$ is challenging as the aggregator does not have access to the label vector X_f^S (feature column for the modeled feature g_f), which remains distributed across source clients. Explicitly sharing X_f^S would also compromise privacy since the aggregator could reconstruct the data of all source clients after the feature model training phase is completed, as each feature is modeled independently. To address this, *Freda* performs the following:

1. **Aggregator Step:** The aggregator computes the intermediate matrix product $K_* K^{-1}$, then applies a random mask matrix $C \in \mathbb{R}^{n_t \times n_t}$, resulting in the masked matrix:

$$\tilde{B} = C K_* K^{-1} \tag{7}$$

   The masked matrix $\tilde{B}$ is split row-wise into sub-matrices corresponding to each source client's data, denoted $\tilde{B}^{s_i}$. The aggregator sends $\tilde{B}^{s_i}$ to source client i, and separately sends C^{-1} to the target client.
2. **Source Client Computation:** Each source client i receives their masked sub-matrix $\tilde{M}^{s_i}$ and locally computes:

$$v^{s_i} = \tilde{B}^{s_i} X_f^{s_i}, \tag{8}$$

   where $X_f^{s_i}$ is the vector of values for feature f held by client i. The result $v^{s_i} \in \mathbb{R}^{n_t \times 1}$ is then sent to the target client.
3. **Target Client Aggregation:** The target client aggregates the received vectors:

$$v = \sum_{i=1}^{N} v^{s_i}, \tag{9}$$

   and removes the mask using the inverse matrix C^{-1}:

$$K_* K^{-1} X_f^S = C^{-1} v = C^{-1} \sum_{i=1}^{N} \tilde{B}^{s_i} X_f^{s_i} \tag{10}$$

This protocol enables the target client to compute the predicted mean without accessing any raw data from the source

clients, thereby preserving data privacy. Additionally, the random masking step ensures that source clients cannot infer information about other clients or the full matrix $K_* K^{-1}$ from their sub-matrices.

Computing the Predicted Variance. To complete the predictive distribution, the predicted variance term $K_{**} - K_* K^{-1} K_*^\top$ is computed by the aggregator using the Gram matrix G^{pq} and sent to the target client. With both the predicted mean and variance available, the target client can reconstruct the full closed-form predictive distribution for feature f as given in Eq. 1.

Feature Weight Computation

For a sample from the target domain, denoted as x_m^t, and a given feature f, let $x_{m,f}^t$ represent the value of feature f in x_m^t, and $x_{m,\neg f}^t$ represent the values of all other features in x_m^t. Given $x_{m,\neg f}^t$, the feature model g_f outputs a posterior distribution that describes the expected value of $x_{m,f}^t$ according to the dependency structure learned from the source domain. For a GPR model, this posterior is a normal distribution, which is directly obtained from the closed-form solution shown in Eq. 1 and collaboratively computed in the previous phase.

To evaluate how well the observed value $x_{m,f}^t$ fits the predicted distribution, we apply the confidence measure proposed by Jalali and Pfeifer (2016):

$$c_f(x_m^t) = 2\Phi\left(\frac{|x_{m,f}^t - \mu_{g_f}(x_{m,\neg f}^t)|}{\sigma_{g_f}(x_{m,\neg f}^t)}\right) \tag{11}$$

where Φ denotes the cumulative distribution function of a standard normal distribution. Here, $\mu_{g_f}(x_{m,\neg f}^t)$ and $\sigma_{g_f}(x_{m,\neg f}^t)$ denote the mean and standard deviation of the predictive distribution computed in Eq. 1 for the input $x_{m,\neg f}^t$. Specifically, they correspond to the closed-form expressions $K_* K^{-1} X_f^S$ and $K_{**} - K_* K^{-1} K_*^\top$, respectively, obtained from the feature model g_f. This confidence score represents the probability that a value as extreme as $x_{m,f}^t$, or more, relative to the predicted mean $\mu_{g_f}(x_{m,\neg f}^t)$, occurs within the posterior distribution predicted by g_f.

The overall confidence for feature f in the target domain is then defined as the average of $c_f(x_m^t)$ across all target domain samples:

$$c_f = \frac{1}{n_t}\sum_{i=1}^{n_t} c_f(x_m^t) \tag{12}$$

Where n_t is the total number of samples in the target domain. For each feature, c_f quantifies how well the source-domain dependencies for feature f align with those in the target domain. Once the confidence scores for all features have been computed, the target client then computes the weight of feature f as follows:

$$w_f = (1 - c_f)^k \tag{13}$$

Here, k is a hyper-parameter specified by the target client, with $k > 0$. This hyper-parameter determines how the confidence scores are transformed into feature weights. As k increases, progressively higher penalties are applied to features with low confidence, while features with higher confidence are penalized less severely. Both the confidence score formulation and the transformation of confidence scores into feature weights are identical to those used in *wenda* (Handl et al., 2019), which also relies on feature-wise predictive distributions to guide adaptation.

In our experiments, we empirically evaluate the performance of our framework with respect to the weighting parameter k and adjust its value accordingly (Section 4.4.3).

Federated Weighted Elastic Net Training

The remaining phases of our framework involves collaboratively training weighted elastic nets in a federated manner, to preserve the privacy of individual source clients. By using a weighted elastic net, source clients scale the contribution of each feature in their local data to the regularization term based on the feature weights computed by the target client in the previous step. The weighted elastic net solves the following optimization problem:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left(\|y - X\beta\|^2 + \lambda J(\beta)\right) \tag{14}$$

where $\|y - X\beta\|^2$ represents the residual sum of squares on the source domain data, λ is the regularization parameter, and $J(\beta)$ is the regularization term defined as:

$$J(\beta) = \alpha \sum_{f=1}^{F} w_f |\beta_f| + \frac{1}{2}(1-\alpha)\sum_{f=1}^{F} w_f \beta_f^2 \tag{15}$$

Training a weighted elastic net is conceptually similar to training a neural network with a single linear layer in a federated setting, where the coefficient for each feature is scaled by its corresponding confidence-based weight. Since all source clients share the same fixed feature weights, the federated training focuses on updating the model coefficients collaboratively.

We employ the FedAvg algorithm (McMahan et al., 2017) to train the weighted elastic net in a privacy-preserving manner. Specifically, each client performs local gradient-based updates using their own data, and the resulting model updates are securely aggregated at the central server (Tajabadi et al., 2024) using the secure aggregation protocol (Bonawitz et al., 2016). The aggregated global model is then broadcast back to all clients for the next training round. To reflect the differing amounts of data across clients, each client's contribution to the global model is weighted proportionally to its local sample size. This setup ensures that clients with larger and more representative data pools have a stronger influence on the global model, which can be particularly beneficial in imbalanced scenarios.

As shown in Equations 14, 15, *freda* has two external parameters: the proportion of L_1 and L_2 penalties in the weighted elastic net α, and the regularization parameter λ. Following Handl et al. (2019), we fix $\alpha = 0.8$.

Optimal Lambda Prediction

The regularization parameter λ plays a critical role in domain adaptation by balancing feature penalty strength and model learning (Handl et al., 2019). If λ is too small, the model may not sufficiently penalize feature differences, while an overly large λ could dominate the objective function, limiting meaningful adaptation. Since cross-validation is impractical in an unsupervised setting where target labels are unavailable, *freda* adopts the prior knowledge approach from Handl et al. (2019), leveraging domain similarities known by the target client.

The target client partitions its data into subsets X^{t_1} and X^{t_2}. Source clients federatively train weighted elastic nets across a range of λ values, and the target client selects the best-performing model for X^{t_1} based on MAE, assuming labels for this subset are available. A simple linear model is then fitted to capture the

relationship between domain similarity and the optimal λ values obtained from source-trained models. This model is used to predict λ values for X^{t_2}, which are then sent back to source clients for the final training phase.

Final Adaptive Model Training

The source clients federatively train the final weighted elastic net models by selecting the regularization parameter λ based on the predicted values received from the target client in the previous step.

The model obtained at the end of this step is sent to the target client by the aggregator for the final prediction task on the target domain data.

Results

To evaluate the performance of our proposed method, we provide a benchmark on the problem of age prediction from DNA methylation data across multiple tissues. This section presents a detailed comparison of *freda* with existing baselines, highlighting its effectiveness in preserving data privacy while achieving competitive predictive accuracy.

Implementation

We implemented our framework in Python 3.8.18, the source code to reproduce the experiments is available on GitHub (https://github.com/mdppml/FREDA). For the feature models, we implemented our own GPR models and used the Python package GPy[1] (gpy, 2012) to compute the optimal values for the hyper-parameter optimization explained in Section 3.1.1. As for the weighted elastic nets, we used TensorFlow 2.13.1 with custom kernel regularization. All federated processes, including source clients, target client, and aggregator, are simulated locally to enable reproducible evaluations. To encourage adoption in broader settings, we also provide a task-agnostic version of our framework, available at https://github.com/mdppml/FREDA-CV. This implementation removes the need for prior domain similarity knowledge by using cross-validation to select the regularization parameter λ.

Dataset and Pre-Processing

We used DNA methylation and donor age data from TCGA (Weinstein et al., 2013) and GEO (Edgar et al., 2002), following the exact preprocessing steps of Handl et al. (2019). This included imputing missing values ($<$ 0.5% of samples) and reducing dimensionality from 466,094 to 12,980 features. Ages were transformed using Horvath's method (Horvath, 2013) to account for nonlinear methylation changes, then standardized. The dataset was split into a source set (1,866 samples from 19 tissues, ages 0–103) and a target set (1,001 samples from 13 tissues, including unseen ones like cerebellum). As in Handl et al. (2019), similar tissue types were aggregated to ensure sufficient sample sizes. Detailed pre-processing steps is available in the supplementary material.

[1] https://github.com/SheffieldML/GPy

Baselines

We compared *freda* with two baselines: the state-of-the-art unsupervised domain adaptation method *wenda* (Handl et al., 2019), and the non-adaptive model (Horvath, 2013).

***Wenda* Baseline**

Wenda (weighted elastic net for domain adaptation) is a state-of-the-art method for unsupervised domain adaptation on small-scale, high-dimensional biological datasets (Handl et al., 2019). Like our approach, it leverages the dependency structure between features across source and target domains, penalizing discrepant features while emphasizing robust ones. Despite its strong performance over non-adaptive models, *wenda* assumes simultaneous access to both domains, hence it is only suitable for non-private settings.

Wenda has three key parameters: the weighting parameter k, the elastic net mixing parameter α, and the regularization parameter λ. Following Handl et al. (2019), we fix $\alpha = 0.8$, while λ is computed using prior knowledge on tissue similarity (*wenda-pn*), as cross-validation on the target domain is infeasible in an unsupervised setting. For k, we select $k = 3$ based on both our experiments and prior work (Handl et al., 2019).

Non-Adaptive Baseline

For our non-adaptive baseline, we adopt the method proposed by Horvath (2013), which combines the elastic net with a least-squares fit. The idea is to first fit a standard elastic net and then apply a linear least-squares fit based only on features that obtained non-zero coefficients in the elastic net. This baseline was first proposed by Horvath (2013) for age prediction from DNA methylation data, where he demonstrated that using an elastic net followed by a least-squares fit resulted in improved performance on his dataset. We refer to this non-adaptive method as *en-ls*.

Setup for Freda

We consider a distributed setting with multiple source domain clients, a target client, and an aggregator, which has no data. The labeled source domain data is distributed across 2, 4, or 8 source clients, and we evaluate *freda* in each of these settings.

Data Distribution

Source domain data is assigned uniformly at random among the source clients. Given that the DNA methylation dataset contains 1,866 training samples, each client receives approximately 933, 466, or 233 samples in the 2, 4, and 8-client settings, respectively.

To assess the robustness of *freda*, we do not consider tissue types when distributing data. Due to the inherent imbalance of DNA methylation data across tissues, this results in some clients having only a few or no samples from certain tissues.

Setup for Weighted Elastic Net Models

The weighted elastic net model is trained for 100 global iterations, with source clients updating their local models for 20 epochs per iteration before the global model is securely updated. To improve convergence, we apply an exponential learning rate decay across iterations, starting at 1×10^{-4} and decreasing to 1×10^{-5} by the final iteration (Yan et al., 2022).

Parameter Selection in *Freda*

Freda has three key parameters: the weighting parameter k, the elastic net mixing parameter α, and the regularization parameter λ. We fix $\alpha = 0.8$ as a design choice and determine λ using the prior knowledge approach (Section 3.3.1) proposed by Handl et al. (2019).

For tissue similarity calculations, we use data from the GTEx consortium, which provides genotype and gene expression data across 42 human tissues (analysts: Aguet François 1 Brown Andrew A. 2 3 4 Castel Stephane E. 5 6 Davis Joe R. 7 8 He Yuan 9 Jo Brian 10 Mohammadi Pejman 5 6 Park YoSon 11 Parsana Princy 12 Segrè Ayellet V. 1 Strober Benjamin J. 9 Zappala Zachary 7 8 et al., 2017), following the methodology in Handl et al. (2019). In the federated setting, only the target domain owner requires access to tissue similarity information.

To evaluate performance, we follow the evaluation strategy of Handl et al. (2019) for *wenda-pn*, iteratively splitting test tissues into subsets: one for fitting domain similarity relationships and another for evaluation (Section 3.3.1). We iterate over all three-tissue combinations with at least 20 training samples, assessing performance on the remaining tissues. Based on our experiments, the optimal weighting parameter is $k = 3$.

Experiments

We compare the performance of *freda* against *wenda-pn* and the non-private, non-adaptive baseline model *en-ls*, as described in Section 4.3. The main performance metric is the Mean Absolute Error (MAE) of the predicted chronological ages of the tissues. For *wenda-pn*, we calculate the MAE only on samples not used for fitting the tissue similarity-λ relationship, reporting the mean and standard deviation across all splits. Similarly, for *freda*, we report the MAE exclusively for the target client's tissues that were not part of the similarity-λ fit, along with the mean and standard deviation over all splits.

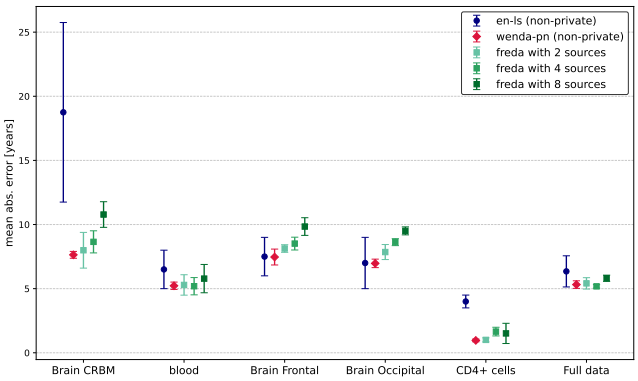

Fig. 2: MAE per target tissue and on the full target dataset for *en-ls*, *wenda-pn* ($k = 3$), and *freda* ($k = 3$) with 2, 4, and 8 source parties.

For the non-adaptive baseline *en-ls*, Handl et al. emphasize that the heterogeneous nature of the data and the random splitting of the training data used for 10-fold cross-validation significantly influence its performance. Therefore, we follow their approach and report the mean ± standard deviation over 10 runs for *en-ls*. For *wenda-pn*, the mean ± standard deviation is calculated

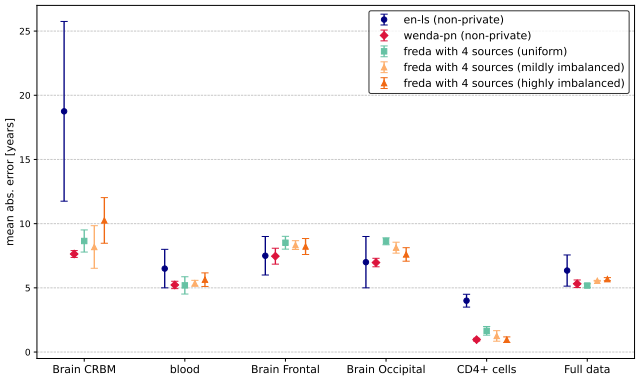

Fig. 3: MAE per target tissue, as well as on full target data, for *en-ls*, *wenda-pn* ($k = 3$), and *freda* ($k = 3$) under uniform and two imbalanced 4-client data distributions.

over all splits of the test tissues where the tissue of interest was included in the evaluation set. For *freda*, we report the mean ± standard deviation for each setting (2, 4, and 8 sources) over 5 different uniform random distributions of source data across the source parties, considering all splits where the tissue of interest was included in the evaluation set.

For *wenda-pn*, Handl et al. (2019) treat each tissue in the test dataset as a separate target domain, training the final weighted elastic net models independently for each tissue. Specifically, Handl et al. (2019) compute the average confidences, as defined in Equation 12, only over the samples of the same tissue and train a separate model for each tissue, always using the entirety of the training (source) data but applying tissue-specific feature weights. We follow the same approach for *freda* in all our experiments, where the clients inside the federated learning system train a separate weighted elastic net model for each tissue in the target domain (for further information see Section 3).

The performance of *en-ls*, *wenda-pn*, and *freda* for 2, 4, and 8 source parties on the relevant tissues of the target domain, as well as on all samples of the target domain data, is shown in Figure 2. For the full target dataset, the non-private baseline methods *en-ls* and *wenda-pn* yield an MAE of 6.34 ± 1.21 and 5.31 ± 0.29, respectively. These results indicate that when the entire target domain data is considered, *wenda-pn* provides only a slight improvement in performance compared to the non-adaptive *en-ls*. The effect of distribution shift is most visible when we observe the performance of our baselines on cerebellum samples. As shown in Figure 2, the non-adaptive *en-ls* yields a significantly higher MAE on cerebellum samples compared to other tissues.

Figure 4 show the predicted versus true ages for the samples of the target domain data, colored by tissue, for *freda* with $k = 3$ for 2, 4, and 8 source parties, *en-ls* and *wenda-pn*, respectively. From Figures 4d and 4e, we can clearly see that both non-private methods perform well on most tissues, except for *en-ls* on cerebellum samples. As shown in Figure 4d, the ages predicted by *en-ls* for cerebellum samples are consistently lower than the true chronological ages. In contrast, Figure 4e demonstrates that *wenda-pn* achieves much closer alignment between the predicted and true ages for cerebellum samples.

Additionally, for the remaining target domain tissues, the predictions of *wenda-pn* are comparable to those of *en-ls*, as

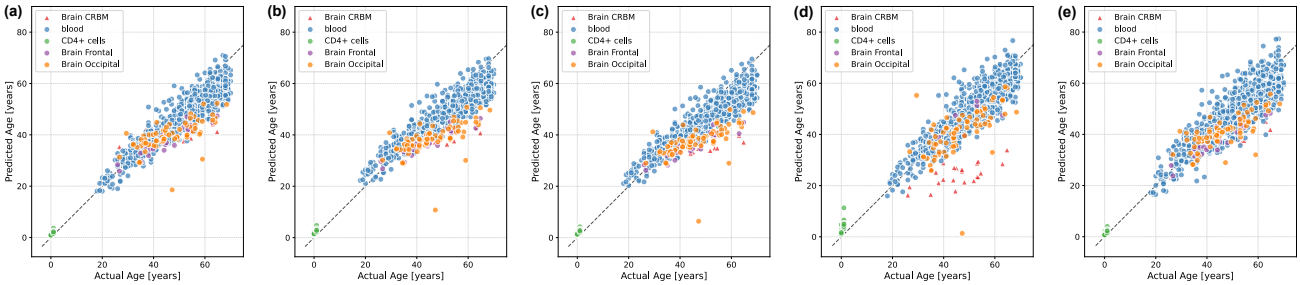

Fig. 4: Predicted versus true chronological age under various settings. Figures **(a)**, **(b)**, and **(c)** correspond to *freda* with $k = 3$ for 2, 4, and 8 source parties, respectively. Predictions are averaged over all splits where the tissue of interest was included in the evaluation set, as well as over 5 different distributions for each setting. Panels **(d)** and **(e)** correspond to *en-ls* and *wenda-pn*, respectively. For *en-ls*, predictions are averaged over 10 runs of 10-fold cross-validation, while for *wenda-pn*, predictions are averaged over all splits where the tissue of interest was included in the evaluation set.

confirmed by the quantitative results in Figure 2. *Wenda-pn* not only yields significantly lower errors than *en-ls* on cerebellum samples but also maintains similar or better performance on other test tissues. Specifically, on cerebellum samples, *en-ls* produces a mean absolute error (MAE) of 7.63 ± 0.26. These results highlight the significance of improving prediction performance on cerebellum samples without a drop in performance on other tissues.

Our experimental results, presented in Figures 2 and 4, are consistent with the findings of Handl et al. (2019), who highlight the difficulty of predicting the age of cerebellum samples. These samples are not represented in the training data and are known to be biologically distinct, even from other brain tissues, in terms of function and gene expression patterns (Fraser et al., 2005; analysts: Aguet François 1 Brown Andrew A. 2 3 4 Castel Stephane E. 5 6 Davis Joe R. 7 8 He Yuan 9 Jo Brian 10 Mohammadi Pejman 5 6 Park YoSon 11 Parsana Princy 12 Segrè Ayellet V. 1 Strober Benjamin J. 9 Zappala Zachary 7 8 et al., 2017). Hence, our evaluation focuses on whether federated privacy-preserving domain adaptation, as implemented by *freda*, can achieve comparable performance on these samples to the non-private method *wenda-pn*.

For the full target dataset, *freda* achieves a MAE of 5.41 ± 0.44, 5.41 ± 0.44, and 5.81 ± 0.24 for the 2, 4, and 8 source domain settings, respectively. These results indicate that, when considering the full target domain data, *freda* provides a performance level almost identical to that of *wenda-pn* and consistently better than *en-ls* across all configurations, despite operating in a distributed environment.

Effect of Data Distribution on Performance

To evaluate the robustness of *freda* under more realistic deployment scenarios, we extended our benchmark by introducing non-uniform data distributions across source clients. While our primary experiments used a uniform random distribution of samples among clients, real-world federated settings often exhibit substantial data imbalance. In our context, where the task is regression and the data is already highly tissue-imbalanced, we focus on *sample-wise imbalance*.

We simulate two increasingly imbalanced scenarios in the 4 source-client setting. In the first, mildly imbalanced setting, clients receive data according to a skewed distribution of $[0.5, 0.2, 0.2, 0.1]$, and in the second, highly imbalanced setting, sample proportions follow $[0.533, 0.266, 0.133, 0.068]$, where each client has roughly double the number of samples of the next. These distributions mimic real-world scenarios where certain institutions contribute significantly more data than others.

As shown in Figure 3, *freda* maintains strong predictive performance even under considerable sample imbalance. Although minor degradations in MAE can be observed specifically for cerebellum samples, the overall performance remains competitive with non-private baselines.

Discussion

Cerebellum samples continue to represent the most challenging case for age prediction under domain shift, consistent with prior findings (Handl et al., 2019; Fraser et al., 2005; analysts: Aguet François 1 Brown Andrew A. 2 3 4 Castel Stephane E. 5 6 Davis Joe R. 7 8 He Yuan 9 Jo Brian 10 Mohammadi Pejman 5 6 Park YoSon 11 Parsana Princy 12 Segrè Ayellet V. 1 Strober Benjamin J. 9 Zappala Zachary 7 8 et al., 2017). These samples differ biologically from other brain and non-brain tissues, and are not well represented in training data. Despite this, *freda* achieves comparable performance to the non-private method *wenda-pn* in the 2- and 4-source scenarios, with MAEs of 7.99 ± 1.39 and 8.64 ± 0.86, respectively. In contrast, the non-adaptive baseline *en-ls* consistently underestimates the ages of cerebellum samples, leading to poor performance on this difficult target domain.

Across all test tissues, *freda* closely matches the performance of *wenda-pn* while significantly outperforming *en-ls*. Notably, despite being trained in a privacy-preserving federated setting, *freda* maintains high predictive performance and effectively captures domain-specific distribution shifts. This confirms that the federated adaptation strategy does not sacrifice performance, even though clients operate under strict data privacy constraints.

We also investigated the impact of scaling to more source clients. As the number of source domains increases from 2 to 8, a slight degradation in performance is observed on cerebellum samples, with MAE increasing to 10.77 ± 0.99. This suggests that partitioning the source data too finely can hinder adaptation performance, possibly due to reduced statistical power per client. Nonetheless, even in the 8-party case, *freda* still outperforms the non-private and non-adaptive baseline.

To further assess robustness in more realistic settings, we extended our evaluation with experiments using imbalanced

data distributions across source clients. These scenarios simulate common federated learning situations where data contributions vary widely across institutions. Interestingly, we observe that predictive performance on some target tissues actually improves as the degree of imbalance increases. We attribute this to two main factors: first, our randomized encoding scheme for feature modeling allows the aggregator to compute global feature statistics as if all data were pooled, despite privacy constraints. Second, during the federated training of the weighted elastic nets, weighted aggregation implicitly favors clients with larger datasets, which can lead to more stable model updates when the dominant client has a representative sample distribution. Performance on cerebellum, however, remains more sensitive to imbalance possibly due to its distinct biological characteristics. Despite this, *freda* maintains competitive performance, demonstrating robustness to imbalanced real-world settings.

Together, these findings demonstrate that *freda* successfully balances privacy, performance, and adaptability, even in challenging domain adaptation tasks and realistic federated learning settings.

Conclusion

In this article, we introduced *freda*, the first privacy-preserving framework for federated unsupervised domain adaptation in regression tasks on high-dimensional, small-scale biological datasets. *Freda* enables multiple entities to collaboratively model complex feature relationships while maintaining complete data privacy. By combining randomized encoding and secure aggregation, it addresses the challenge of training Gaussian Processes in distributed settings, eliminating the need for pooled pairwise computations on non-shareable data.

Our evaluation on an age prediction task from DNA methylation data demonstrates that *freda* achieves performance comparable to non-private methods, including on challenging tissues such as cerebellum, while preserving data privacy. In addition, we observe that *freda* remains robust under increasingly imbalanced data distributions.

While *freda* demonstrates competitive performance to the non-private state-of-the-art even in distributed settings, we acknowledge that training a separate feature model for each feature in high-dimensional settings can be computationally intensive. However, there are several directions to improve scalability that we plan to explore in future work. First, since feature models are independent, they can be trained in parallel across multiple processors or compute nodes to reduce runtime. Second, recent work (Hannemann et al., 2025) proposes a more efficient masking strategy for the same randomized encoding framework used in *freda*, which reduces computation time significantly by speeding up the masking process.

Funding

This work was supported by the German Federal Ministry of Education and Research (BMBF) under project number 01ZZ2010 (MDPPML) and 01ZZ2316D (PrivateAIM).

References

Gpy: A gaussian process framework in python, 2012. Sheffield Machine Learning Group.

G. C. L. analysts: Aguet François 1 Brown Andrew A. 2 3 4 Castel Stephane E. 5 6 Davis Joe R. 7 8 He Yuan 9 Jo Brian 10 Mohammadi Pejman 5 6 Park YoSon 11 Parsana Princy 12 Segrè Ayellet V. 1 Strober Benjamin J. 9 Zappala Zachary 7 8, N. program management: Addington Anjene 15 Guan Ping 16 Koester Susan 15 Little A. Roger 17 Lockhart Nicole C. 18 Moore Helen M. 16 Rao Abhi 16 Struewing Jeffery P. 19 Volpi Simona 19, P. S. L. . B. M. E. . B. P. A. 16, N. C. F. N. C. R. 137, et al. Genetic effects on gene expression across human tissues. *Nature*, 550(7675):204–213, 2017.

C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7):878, 2016.

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

R. Edgar, M. Domrachev, and A. E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002.

A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pages 877–894, 2021.

H. Feng, Z. You, M. Chen, T. Zhang, M. Zhu, F. Wu, C. Wu, and W. Chen. Kd3a: Unsupervised multi-source decentralized domain adaptation via knowledge distillation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3274–3283. PMLR, 18–24 Jul 2021.

H. B. Fraser, P. Khaitovich, J. B. Plotkin, S. Pääbo, and M. B. Eisen. Aging and gene expression in the primate brain. *PLoS biology*, 3(9):e274, 2005.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

J. G. Greener, S. M. Kandathil, L. Moffat, and D. T. Jones. A guide to machine learning for biologists. *Nature reviews Molecular cell biology*, 23(1):40–55, 2022.

L. Handl, A. Jalali, M. Scherer, R. Eggeling, and N. Pfeifer. Weighted elastic net for unsupervised domain adaptation with application to age prediction from dna methylation data. *Bioinformatics*, 35(14):i154–i163, 2019.

A. Hannemann, A. B. Ünal, A. Swaminathan, E. Buchmann, and M. Akgün. A privacy-preserving framework for collaborative machine learning with kernel methods. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 82–90. IEEE, 2023.

A. Hannemann, A. Swaminathan, A. B. Ünal, and M. Akgün. Private, efficient and scalable kernel learning for medical image analysis. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 81–95. Springer, 2025.

S. Horvath. Dna methylation age of human tissues and cell types. *Genome biology*, 14:1–20, 2013.

A. Jalali and N. Pfeifer. Interpretable per case weighted ensemble method for cancer associations. *BMC genomics*, 17:1–10, 2016.

E. Jiang, Y. J. Zhang, and S. Koyejo. Principled federated domain adaptation: Gradient projection and auto-weighting. In *The Twelfth International Conference on Learning Representations*, 2024.

M. M. Li, K. Huang, and M. Zitnik. Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering*, 6(12):1353–1369, 2022.

J. Liang, D. Hu, Y. Wang, R. He, and J. Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021.

X. Liu, Z. Chen, L. Zhou, D. Xu, W. Xi, G. Bai, Y. Zhao, and J. Zhao. Ufda: Universal federated domain adaptation with practical assumptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14026–14034, 2024.

M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016.

M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10):749–760, 2018.

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

S. Orouji, M. C. Liu, T. Korem, and M. A. K. Peters. Domain adaptation in small-scale and heterogeneous biological datasets. *Science Advances*, 10(51):eadp6040, 2024. doi: 10.1126/sciadv.adp6040.

X. Peng, Z. Huang, Y. Zhu, and K. Saenko. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054*, 2019.

R. Poplin, A. V. Varadarajan, K. Blumer, Y. Liu, M. V. McConnell, G. S. Corrado, L. Peng, and D. R. Webster. Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. *Nature biomedical engineering*, 2(3):158–164, 2018.

M. Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. *Advances in neural information processing systems*, 29, 2016.

B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

B. Sun, H. Huo, Y. Yang, and B. Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. *Advances in Neural Information Processing Systems*, 34:23309–23320, 2021.

M. Tajabadi, R. Martin, and D. Heider. Privacy-preserving decentralized learning methods for biomedical applications. *Computational and Structural Biotechnology Journal*, 2024.

V. Thumuluri, J. J. Almagro Armenteros, A. Johansen, H. Nielsen, and O. Winther. DeepLoc 2.0: multi-label subcellular localization prediction using protein language models. *Nucleic Acids Research*, 50(W1):W228–W234, 04 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac278.

G. Valentini, R. Tagliaferri, and F. Masulli. Computational intelligence and machine learning in bioinformatics. *Artificial Intelligence in Medicine*, 45(2):91–96, 2009. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2008.08.014. Computational Intelligence and Machine Learning in Bioinformatics.

M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

Z. Weng, X. Yang, A. Li, Z. Wu, and Y.-G. Jiang. Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In *International Conference on Machine Learning*, pages 36978–36989. PMLR, 2023.

C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

J. Wu, J. He, S. Wang, K. Guan, and E. Ainsworth. Distribution-informed neural networks for domain adaptation regression. *Advances in Neural Information Processing Systems*, 35:10040–10054, 2022.

G. Yan, H. Wang, and J. Li. Seizing critical learning periods in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8788–8796, 2022.

Z. Yue, Q. Sun, and H. Zhang. Make the u in uda matter: Invariant consistency learning for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 36:26991–27004, 2023.