Classification Under Local Differential Privacy with Model Reversal and Model Averaging

Caihong Qin   CAIHQIN@IU.EDU
Department of Epidemiology and Biostatistics
School of Public Health, Indiana University
Indiana, USA

Yang Bai   STATBYANG@MAIL.SHUFE.EDU.CN
School of Statistics and Data Science &
Institute of Data Science and Statistics
Shanghai University of Finance and Economics
Shanghai, China

Editor: Rahul Mazumder

Abstract

Local differential privacy (LDP) has become a central topic in data privacy research, offering strong privacy guarantees by perturbing user data at the source and removing the need for a trusted curator. However, the noise introduced by LDP often significantly reduces data utility. To address this issue, we reinterpret private learning under LDP as a transfer learning problem, where the noisy data serve as the source domain and the unobserved clean data as the target. We propose novel techniques specifically designed for LDP to improve classification performance without compromising privacy: (1) a noised binary feedback-based evaluation mechanism for estimating dataset utility; (2) model reversal, which salvages underperforming classifiers by inverting their decision boundaries; and (3) model averaging, which assigns weights to multiple reversed classifiers based on their estimated utility. We provide theoretical excess risk bounds under LDP and demonstrate how our methods reduce this risk. Empirical results on both simulated and real-world datasets show substantial improvements in classification accuracy.

Keywords: dataset utility, excess risk, functional data, private learning, transfer learning

1. Introduction

Data privacy has become a paramount concern as technological innovations continue to permeate daily life. Differential Privacy (DP; Dwork et al., 2006) has emerged as a leading framework for privacy-preserving statistical analyses, providing a rigorous definition that limits the information an adversary can learn from released summaries. However, classical DP often assumes a trusted data curator who receives the raw data and applies privacy mechanisms. This assumption often fails in scenarios involving untrusted data collectors or highly sensitive information (Liu et al., 2022). Alternative privacy-preserving approaches such as cryptographic methods (Salami et al., 2023) and Homomorphic Encryption (Marcolla et al., 2022) have been considered. While these methods can protect raw data through encryption and secure computation, they typically introduce high computational overhead and do not inherently address all privacy risks from published statistics (Gong et al., 2024;

Peng, 2019). By contrast, Local Differential Privacy (LDP; Kasiviswanathan et al., 2011) addresses the problem at the data source. Under LDP, each individual perturbs their own data before sending any information to the collector, thereby removing the need for a trusted curator. This decentralized approach can significantly mitigate privacy leakage and is relatively scalable in real-world systems. Currently, multiple variants of LDP exist (Geumlek and Chaudhuri, 2019; Erlingsson et al., 2020; Meehan et al., 2022; Huang and Canonne, 2022; Qin et al., 2023), with pure ε-LDP being the most widely used and well-established. Its formal definition is given as follows:

Definition 1 (ε-LDP, Kasiviswanathan et al. 2011) *A randomized mechanism $\mathcal{M}$ is said to satisfy ε-local differential privacy (LDP) if, for any pair of input values u_1, u_2 in the domain of $\mathcal{M}$ and any possible output v, it holds that*

$$\mathbb{P}\big(\mathcal{M}(u_1) = v\big) \ \leq \ e^{\varepsilon} \, \mathbb{P}\big(\mathcal{M}(u_2) = v\big).$$

LDP has already been deployed by major technology companies such as Apple (Differential Privacy Team, 2017), Google (Erlingsson et al., 2014), and Microsoft (Ding et al., 2017). While it offers stronger privacy protects against untrusted data collectors, LDP often adds more noise per user than standard DP, potentially harming data utility. Balancing the privacy-utility trade-off is a key challenge, particularly in private learning scenarios, which involve training machine learning models under LDP constraints (Yang et al., 2020). This contrasts with much of the existing LDP literature, which focuses on simpler statistical queries (e.g., frequency counts or sums) (Wang et al., 2017).

Private learning under LDP faces two major difficulties (Ye and Hu, 2020). First, preserving correlations among features (Wang et al., 2019b) and between features and labels (Yilmaz et al., 2020) is critical for accurate model training. The noise required by LDP can disrupt these relationships, reducing model accuracy. Second, when data are high-dimensional, one may split the privacy budget ε across dimensions or randomly select a single dimension to report (Nguyên et al., 2016; Arcolezi et al., 2021, 2022). As the dimensionality d grows, the effective privacy budget per dimension drops drastically, deteriorating utility even further. Meanwhile, omitting certain dimensions altogether may cause incomplete information for learning.

These challenges motivate our work. We focus on the classification problem, aiming to fully exploit data utility and enhance classifier performance while maintaining the same level of LDP protection. By better leveraging noisy data, we aim to boost predictive accuracy on the true (unperturbed) distribution without sacrificing privacy. Equivalently, for a desired performance target, stronger privacy guarantees can be offered to data owners. We draw inspiration from transfer learning (Weiss et al., 2016; Hosna et al., 2022), which typically uses source data to aid learning on a target domain. Here, the noised (LDP-perturbed) data play the role of source data, and the unobserved true data serve as the target. However, two key distinctions arise:

- Unlike standard transfer learning, we lack direct observations from the target distribution, as only noised data are available under LDP.

- Negative transfer scenarios (Li et al., 2022) (where some source datasets could degrade performance) may be more frequent under LDP, because the noise injection can substantially distort the correlation information.

Although existing transfer learning techniques cannot be applied directly, we adapt several ideas to measure and exploit the transferability of noised datasets. Doing so enables us to assess dataset utility and design robust private-learning techniques. Below, we outline how our methods address the classic transfer learning questions (Pan and Yang, 2009)—what to transfer, when to transfer, and how to transfer—under LDP settings:

1. What to transfer. Our goal is to learn the relationship between the response and covariates. One standard approach under LDP collects noised feature-label pairs from clients for model training. In addition to this, we introduce a novel LDP mechanism in which a subset of clients provides privatized binary evaluations of model performance. This mechanism preserves correlation structure and achieves higher utility by more effectively leveraging the underlying relationship between features and responses.

2. When to transfer. We quantify the transferability, i.e., the utility of an LDP-perturbed dataset, to assess whether a source is beneficial or detrimental. Our proposed LDP-tailored evaluation process provides an unbiased estimate of this utility, enabling more effective identification of negative datasets or classifiers.

3. How to transfer. We introduce two new techniques, guided by our utility evaluation, to improve performance of classifiers obtained under LDP:

   (a) Model Reversal (MR): When a noised classifier performs worse than random guessing, we reverse its decision boundary. This effectively corrects negative datasets without discarding them entirely.

   (b) Model Averaging (MA): We adapt ensemble-learning ideas to the LDP context, assigning weight to each reversed classifier based on its estimated utility. The resulting averaged classifier tends to outperform any single classifier, especially in the presence of high noise.

We address key challenges in private learning under LDP by introducing a novel framework inspired by transfer learning. Our main contributions are summarized below:

1. **Linking LDP and Transfer Learning.** We recast private learning under LDP as a transfer learning problem, interpreting the noisy dataset as a source and the true data as a target. Building on Qin et al. (2024), we use their proposed definition of transferability to measure dataset utility under a given privacy budget.

2. **Techniques for Private Learning.** We develop three LDP-tailored tools:

   (a) An evaluation scheme that requests privatized binary feedback on a classifier's performance, based on which we provide unbiased accuracy estimates, effectively addressing the lack of target data under LDP.

   (b) Model reversal, which salvages weak classifiers with below-50% accuracy by inverting their decision boundaries, effectively leveraging negative datasets.

   (c) Model averaging, which averages multiple reversed weak classifiers, weighting them by their estimated utility under LDP.

3. **Theoretical Guarantees.** We provide excess risk bounds for our proposed methods, illustrating how noise affects classification performance under LDP and demonstrating how MR and MA help reduce this bound.

Beyond these contributions, our proposed techniques are broadly applicable across various data structures and classification algorithms. The framework can be easily adapted to different privacy protection mechanisms by modifying the noise injection strategy. As a demonstration, we present a functional data classification example that, to our knowledge, represents the first LDP-based approach for modeling functional covariates.

In the following sections, we formalize these ideas and demonstrate their effectiveness both theoretically and empirically. Section 2 reviews related work. Section 3 introduces our utility evaluation process and the proposed techniques tailored for the LDP setting. Section 4 establishes excess risk bounds for the proposed classifiers. Section 5 presents the complete process of building a functional classifier under LDP using our framework. Section 6 reports experimental results that demonstrate the effectiveness of the proposed techniques. Section 7 presents real-data applications, and Section 8 concludes the paper.

2. Related work

In this section, we review the literature on supervised learning under LDP, with a particular focus on classification methods. We then discuss key concepts from transfer learning that motivate our approach, followed by a brief overview of general classification techniques. Finally, we highlight relevant developments in functional data analysis that inform our application setting.

Supervised Learning Under LDP. Given challenges arising from data correlations and high dimensionality, existing research on supervised learning under LDP is limited. Although a few studies have explored learning under LDP, preserving both privacy and utility remains a significant challenge. Some research focuses on empirical risk minimization (ERM), treating the learning process as an optimization problem solved through defining a series of objective functions. Wang et al. (2019a) constructed a class of machine learning models under LDP that can be expressed as ERM, and solved by stochastic gradient descent (SGD). To address the high dimensionality issue, Liu et al. (2020) privately selected the top-k dimensions according to their contributions in each iteration of federated SGD. Deep learning models under LDP have also been studied; for example, Sun et al. (2021) employed adaptive data perturbation and parameter shuffling to mitigate privacy loss. Nevertheless, balancing privacy guarantees with competitive performance is still an open problem, particularly in large-scale, high-dimensional settings.

Classification Under LDP. Currently, only a few classification algorithms have been developed under LDP. Among the most relevant are the works of Yilmaz et al. (2020); Berrett and Butucea (2019); Ma and Yang (2024). Yilmaz et al. (2020) proposed a naive Bayes classifier using LDP-based frequency and mean estimators. Their method assumes independence among features, which may limit its applicability in real-world settings. While they suggest dimensionality reduction techniques such as principal component analysis and discriminant component analysis for high-dimensional data, they do not detail how these can be implemented in the LDP context, where raw data are not directly accessible. Berrett and Butucea (2019) introduced a histogram-based classifier and established minimax rates

for excess risk. Their approach applies Laplace noise with a scale parameter that grows exponentially with the number of features, potentially introducing substantial distortion, even in moderately low-dimensional settings, which may severely degrade classification performance. Ma and Yang (2024) considered the setting where additional public data are available and proposed an LDP decision tree classifier that achieves minimax-optimal rates under this setup. As discussed in Section 1, leveraging public data effectively transforms the problem into a transfer learning task, aligning with the framework in Cai and Wei (2021). Their algorithm also remains implementable in the absence of public data, in which case the procedure reduces to a standard locally private tree classifier.

Transfer Learning. Transfer learning (Torrey and Shavlik, 2010) enhances a target task by leveraging knowledge acquired from related source tasks. Its success depends on accurately quantifying the relationship between source and target domains to assess how informative the source is and to detect negative sources. Negative transfer occurs when a poorly matched source degrades target performance (Weiss et al., 2016). The identification of informative or negative sources critically depends on the chosen metric for transferability. In a series of parameter-based transfer learning works, it is common to measure the transferability of a source dataset by quantifying the distance between the parameters of interest in the source and target datasets; see Li et al. (2022); Lin and Reimherr (2022); Tian and Feng (2022). Yet, parameter-based measures alone can be misleading in classification settings, as smaller parameter distances do not necessarily imply higher transferability (Qin et al., 2024). Alternative approaches assess the distributional similarity through a relative signal exponent (Cai and Wei, 2021) or label-matching probabilities (Qin et al., 2024). In the context of LDP, available noised datasets, which serve as source data, are often prone to negative transfer because severe perturbations can distort key information and hinder reliable transferability assessment.

Standard Classification Methods. Classification tasks appear across numerous real-world scenarios, ranging from medical diagnoses to spam detection. Classic methods include discriminant analysis (Hall et al., 2001), Naïve Bayes classification (Dai et al., 2017), and logistic regression (Leng and Müller, 2006), while contemporary research often employs more advanced approaches such as support vector machines (SVM, Blanchard et al., 2008) or neural-network-based classifier (Saravanan and Sasithra, 2014). Comprehensive overviews of classification methods can be found in Kotsiantis et al. (2007); Kumari and Srivastava (2017). From a theoretical perspective, many works investigate excess risk bounds to establish statistical guarantees. For instance, Blanchard et al. (2008) derived non-asymptotic bounds for SVM-based classifiers, Sang et al. (2022) developed similar results for distance-weighted discrimination (DWD), and Ko et al. (2023) studied the excess risk convergence rates of neural network-based classifiers. In transfer learning contexts, Cai and Wei (2021) and Qin et al. (2024) analyzed the excess risk of transfer-learning-based classifiers. However, results on classification under LDP remain limited, particularly in understanding how noise injection impacts both model training and theoretical guarantees.

Functional Data Analysis. Functional data, such as curves or surfaces defined over continuous domains, are inherently infinite-dimensional (Ramsay and Silverman, 2005; Horváth and Kokoszka, 2012). Common examples include time-continuous measurements collected by wearable devices, which are often associated with sensitive health information. To facilitate statistical analysis, recent research has focused on functional projection tech-

niques that map functional observations into finite-dimensional representations (Horváth and Rice, 2015; Pomann et al., 2016; Kraus and Stefanucci, 2019; Bai et al., 2023). Several differential privacy methods have been developed for functional data (Hall et al., 2013; Mirshani et al., 2019; Jiang et al., 2023; Lin and Reimherr, 2023), but most are based on the global DP setting, where a trusted data curator has access to the entire dataset. In contrast, LDP requires that data be perturbed at the individual level prior to collection, introducing unique challenges for functional data. As a result, training predictive models on functional data under LDP remains an open and underexplored problem, leaving a gap in privacy-preserving techniques for infinite-dimensional data.

3. Privacy-Preserving Classification under LDP

Under local differential privacy, observed data are perturbed to protect sensitive information, which often leads to a loss in utility. In this section, we introduce a utility measure to evaluate the quality of perturbed data and propose novel techniques to improve classifier performance under LDP without compromising privacy.

3.1 Data and Notation

In practice, original data collected from clients can be of diverse forms, including high-dimensional data (Araveeporn, 2021), functional data (Kraus and Stefanucci, 2019), images (Rawat and Wang, 2017), and text (Blitzer et al., 2006). To derive usable features from these data under privacy constraints, one often applies suitable representation methods and then adds random noise to the resulting features or labels. In this subsection, we detail the representation process and introduce the notations used throughout.

Let $\mathcal{X}$ be an instance set, i.e., the space of original inputs (e.g., the diverse data types mentioned above). Let $\mathcal{Z}$ be a feature space, with $\mathcal{Z} = \mathbb{R}^d$ as a common example, where elements in $\mathcal{Z}$ correspond to finite-dimensional representations of the original input. Let $\mathcal{Y} = \{-1, 1\}$ denote the label set for binary classification. A mapping function is characterized by a distribution over $\mathcal{X}$ and a (stochastic) mapping function $\zeta : \mathcal{X} \to \mathcal{Y}$. For an instance $X \in \mathcal{X}$, the value $\zeta(X)$ corresponds to the probability that its label Y is 1. Let Q denote the joint distribution of $\{X, Y\}$, and let Q_X be the marginal distribution of X. We write $Q_{Y|X}(x) = \zeta(x)$ to represent the conditional distribution $\mathbb{P}(Y = 1 \mid X = x)$.

A representation function $\mathcal{R} : \mathcal{X} \to \mathcal{Z}$ maps instances to features, thereby inducing a distribution over $\mathcal{Z}$ and a (stochastic) mapping function $\eta : \mathcal{Z} \to \mathcal{Y}$. Specifically, for any Borel set $B \subseteq \mathcal{Z}$ such that $\mathcal{R}^{-1}(B)$ is Q_X-measurable,

$$\mathbb{P}(Z \in B) \;=\; Q_X\big(\mathcal{R}^{-1}(B)\big).$$

Let P denote the joint distribution of $\{Z, Y\}$, and let P_Z be the marginal distribution of Z. We denote by $\eta(z) \triangleq \mathbb{P}\big(Y = 1 \mid Z = z\big)$ the conditional distribution of Y given Z, where

$$\eta(Z) \;=\; \mathbb{E}_{Q_X}\big[\zeta(X) \mid \mathcal{R}(X) = Z\big].$$

Note that $\eta(Z)$ may be a stochastic function even if $\zeta(X)$ is not. This is because the function $\mathcal{R}$ can map two instances with different labels to the same feature representation (Ben-David et al., 2006).

Next, consider an ε-LDP mechanism that transforms (Z, Y) into $(Z^{(\varepsilon_z)}, Y^{(\varepsilon_y)})$, where $\boldsymbol{\varepsilon} = (\varepsilon_z, \varepsilon_y)$ denotes the privacy budget allocation with $\varepsilon_z, \varepsilon_y > 0$. When both Z and Y require protection, the budget is split such that $\varepsilon_z + \varepsilon_y = \varepsilon$. Notable cases include protecting only covariates ($\varepsilon_z = \varepsilon$, $\varepsilon_y = \infty$, meaning no privacy constraint on Y), or only labels ($\varepsilon_y = \varepsilon$, $\varepsilon_z = \infty$), where the summation constraint does not apply as the entire budget is assigned to one component. One way to achieve ε-LDP is by setting:

$$Z^{(\varepsilon_z)} = Z + \delta^{(\varepsilon_z)},$$
$$\mathbb{P}\big(Y^{(\varepsilon_y)} = Y\big) = q^{(\varepsilon_y)}, \tag{1}$$

where the noise $\delta^{(\varepsilon_z)}$ follows a distribution $D^{(\varepsilon_z)}$ whose scale is calibrated to the sensitivity of Z under the chosen ε_z-LDP mechanism, and smaller ε_z corresponds to a larger noise scale. For example, if $Z \in [a, b]^d$, then under the Laplace mechanism (Dwork et al., 2006) one can add i.i.d. Laplace$(0, \lambda_z)$ noise to each coordinate with $\lambda_z = d(b-a)/\varepsilon_z$. A common choice for $q^{(\varepsilon_y)}$ is $e^{\varepsilon_y}/(1 + e^{\varepsilon_y}) \in [1/2, 1)$, corresponding to the classical randomized response mechanism (Warner, 1965). This added noise weakens the dependency between Z and Y, thereby degrading data utility. In Section 3.2, we show how utility is affected by the allocation of the privacy budget.

Parallel to the original distribution P, let $P^{(\boldsymbol{\varepsilon})}$ be the joint distribution of $(Z^{(\varepsilon_z)}, Y^{(\varepsilon_y)})$. We denote its marginal distribution by $P_Z^{(\boldsymbol{\varepsilon})}$ and the conditional distribution by $P_{Y|Z}^{(\boldsymbol{\varepsilon})}(z) = \mathbb{P}(Y^{(\varepsilon_y)} = 1 \mid Z^{(\varepsilon_z)} = z) := \eta^{(\boldsymbol{\varepsilon})}(z)$. Under LDP, we only observe the noised data $(Z^{(\varepsilon_z)}, Y^{(\varepsilon_y)})$, which defines the source distribution $P^{(\boldsymbol{\varepsilon})}$. Our goal, however, is to learn a model that performs well on the unobserved target distribution P (and ultimately Q), in order to gain insights into the real world. This discrepancy between the source and target distributions highlights a fundamental challenge in the LDP setting.

In this work, we focus on developing techniques that improve classification performance under a given LDP mechanism. The methods proposed in the following sections can be adapted to different variants of LDP (Qin et al., 2023) by appropriately modifying the noise injection procedure in Equation (1). For concreteness, we use the standard ε-LDP framework to illustrate our approach throughout this paper.

3.2 Measure of Utility

In this subsection, we introduce a utility measure for the perturbed dataset $(Z^{(\varepsilon_z)}, Y^{(\varepsilon_y)})$ by adopting the concept of dataset transferability proposed by Qin et al. (2024). For any feature point $z_0 \in \mathcal{Z}$, we define the pointwise utility function using two independent draws: $(Y, Z) \sim P$ from the original distribution and $(Y^{(\varepsilon_y)}, Z^{(\varepsilon_z)}) \sim P^{(\boldsymbol{\varepsilon})}$ from the privatized distribution. Specifically,

$$g^{(\boldsymbol{\varepsilon})}(z_0) = \mathbb{P}\big(Y^{(\varepsilon_y)} = Y \mid Z = z_0, \ Z^{(\varepsilon_z)} = z_0\big) \in [0, 1], \tag{2}$$

which represents the probability that the label from the original distribution with $Z = z_0$ agrees with the label from the privatized distribution with $Z^{(\varepsilon_z)} = z_0$. Proposition 2 below clarifies how $g^{(\boldsymbol{\varepsilon})}(z_0)$ relates to the mapping functions η and $\eta^{(\boldsymbol{\varepsilon})}$, and how it is affected by the privacy budget allocation $(\varepsilon_z, \varepsilon_y)$.

Proposition 2 *With the mapping functions η and $\eta^{(\varepsilon)}$ defined above, we have*

$$g^{(\varepsilon)}(z_0) \; = \; 2\Big(\eta(z_0) - \tfrac{1}{2}\Big)\Big(\eta^{(\varepsilon)}(z_0) - \tfrac{1}{2}\Big) \; + \; \tfrac{1}{2}, \tag{3}$$

where

$$\eta^{(\varepsilon)}(z_0) - \tfrac{1}{2} \; = \; 2\Big(q^{(\varepsilon_y)} - \tfrac{1}{2}\Big) \, \mathbb{E}_{P_Z}\Big[\omega\big(z \mid z_0, \varepsilon_z\big)\Big(\eta(z) - \tfrac{1}{2}\Big)\Big] \tag{4}$$

and

$$\omega\big(z \mid z_0, \varepsilon_z\big) \; = \; \frac{D^{(\varepsilon_z)}\big(\delta_z^{(\varepsilon_z)} = z_0 - z\big)}{\mathbb{E}_{P_Z}\Big\{D^{(\varepsilon_z)}\big(\delta_z^{(\varepsilon_z)} = z_0 - z\big)\Big\}}. \tag{5}$$

In Proposition 2, $\omega\big(z \mid z_0, \varepsilon_z\big)$ is analogous to a continuous weighting scheme, satisfying $\mathbb{E}_{P_Z}\big(\omega(z \mid z_0, \varepsilon_z)\big) = 1$. As discussed in Qin et al. (2024), Equation (3) indicates that $g^{(\varepsilon)}(z_0)$ depends on how η and $\eta^{(\varepsilon)}$ deviate from $1/2$. When they deviate in the same direction, $g^{(\varepsilon)}(z_0) \geq 1/2$, making the noised dataset informative; otherwise, it can be negative. Equation (4) illustrates how the privacy budget parameters $(\varepsilon_z, \varepsilon_y)$ jointly influence $g^{(\varepsilon)}(z_0)$. To gain clearer insight into their roles, we consider the following two notable cases:

1. Protecting only Y: Let $\varepsilon_y = \varepsilon$ and $\varepsilon_z = \infty$. We then have $P_Z^{(\varepsilon)} = P_Z$ but $P_{Y|Z}^{(\varepsilon)} \neq P_{Y|Z}$, which matches the classic posterior drift scenario (Qin et al., 2024). In this setting,

$$\eta^{(\varepsilon)}(z_0) - \tfrac{1}{2} \; = \; 2\Big(q^{(\varepsilon_y)} - \tfrac{1}{2}\Big)\Big(\eta(z_0) - \tfrac{1}{2}\Big),$$

   and the dataset remains informative when $q^{(\varepsilon_y)} \geq 1/2$, meaning the perturbed data and the original data share the same Bayes classifier; otherwise, it becomes negative.

2. Protecting only Z: Let $\varepsilon_z = \varepsilon$ and $q^{(\varepsilon_y)} = 1$. Then $\eta^{(\varepsilon)}(z_0) - 1/2$ primarily depends on the weight $\omega\big(z \mid z_0, \varepsilon_z\big)$ defined in Equation (5). To examine the impact of ε_z on this weight, we consider a special case where $z \sim U(-1, 1)$, the uniform distribution on the interval $(-1, 1)$. The noise $\delta_z^{(\varepsilon_z)}$ is drawn from a Laplace distribution with mean zero and scale parameter $2/\varepsilon_z$, which satisfies ε-LDP. We then compute the weight for $z_0 \in [-2, 2]$ with $\varepsilon_z \in \{10, 1, 0.1, 0.01\}$. Figure 1 illustrates how the weight function $\omega\big(z \mid z_0, \varepsilon_z\big)$ varies with z for different z_0 and ε_z, where the range of weight becomes increasingly concentrated around 1 as ε_z approaches 0. Given that $\int_z \omega\big(z \mid z_0, \varepsilon_z\big)dz = 1$, these heatmaps indicate that as ε_z decreases (i.e., stronger privacy constraints and more noise), the weight function becomes highly diffuse, spreading nearly uniformly over $z \in [-1, 1]$ rather than concentrated around the diagonal $z = z_0$ for $z_0 \in [-1, 1]$ or at the boundaries $z = -1$ and $z = 1$ when $z_0 < -1$ and $z_0 > 1$, respectively. Consequently, as privacy constraints tighten and the weight function grows more diffuse, the deviation between $\eta^{(\varepsilon)}(z) - \tfrac{1}{2}$ and $\eta(z) - \tfrac{1}{2}$ becomes larger.

3.3 Evaluation of Utility

Utility assessment is crucial when integrating information from multiple data sources. Unlike in transfer learning, where one can directly evaluate source-trained models on target
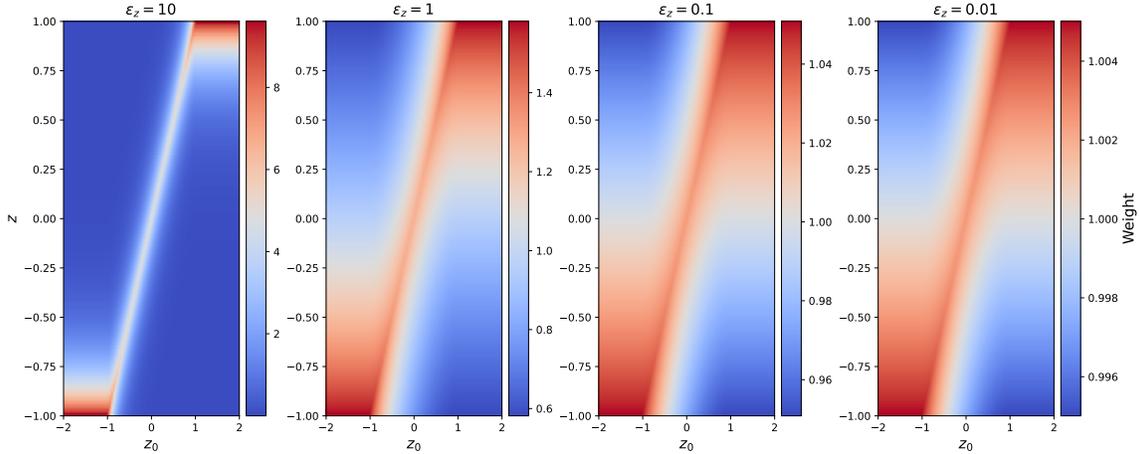
Figure 1: Heatmaps of the weight function $\omega(z \mid z_0, \varepsilon_z)$ changing with z for various values of z_0 and ε_z, where we assume $z \sim U(-1, 1)$ and $z_0 = z + \delta_z^{(\varepsilon_z)}$ with $\delta_z^{(\varepsilon_z)}$ drawn from a Laplace distribution having mean zero and scale $2/\varepsilon_z$. As ε_z decreases, the value intervals of the color bars become increasingly concentrated around 1.

data, the LDP setting lacks access to unperturbed target data. To address this, we propose a novel alternative approach that requires only a noised binary evaluation from each client. Based on these privatized responses, we provide an unbiased estimate of the utility of a model trained on perturbed data.

Given a perturbed dataset $\mathcal{D}_0 = \{(z_i^{(\varepsilon_z)}, y_i^{(\varepsilon_y)})\}_{i=1}^{n_0}$, our goal is to estimate its utility $g^{(\varepsilon)}(z_0)$, as defined in Equation (2), for any given z_0 drawn from the distribution P_Z. A key challenge is that both $y(z_0)$ and $y^{(\varepsilon_y)}(z_0)$ are unobservable. To overcome this, instead of requesting noisy feature-label pairs from each client, we apply the classifier $f^{(\varepsilon)}(z_0)$, trained on $\mathcal{D}_0$, to each client in evaluation set and request a privatized binary response indicating whether $y(z_0) = \hat{y}^{(\varepsilon_y)}(z_0)$, where $\hat{y}^{(\varepsilon_y)}(z_0) = \text{sign}(f^{(\varepsilon)}(z_0))$. This approach enables a more efficient form of querying, as it preserves the correlation between the response and the functional covariate in the target distribution while injecting significantly less noise, since only a single binary response is privatized per client. Furthermore, measuring the utility of $\mathcal{D}_0$ in this way is equivalent to estimating the classification accuracy of $f^{(\varepsilon)}(z_0)$ trained on it. This interpretation aligns with Proposition 2, which indicates that higher dataset utility corresponds to better Bayes classification performance achievable from that dataset.

In what follows, we present the detailed procedures for classifier training and dataset utility evaluation. These involve two distinct types of client-side information access: collecting noised observation pairs and obtaining privatized binary evaluations. We assume a total of N clients, partitioned into a training set $\mathcal{D}_{\text{train}}$ of size N_0 and an evaluation set $\mathcal{D}_{\text{eval}}$ of size N_1, where $N_0 + N_1 = N$.

1. **Classifier Training.** The server collects perturbed data $\{(z_i^{(\varepsilon_z)}, y_i^{(\varepsilon_y)})\}_{i \in \mathcal{D}_{\text{train}}}$ from the training clients to fit or construct multiple weak classifiers. Specifically, for each $b = 1, \ldots, B$, we randomly sample $n_0(< N_0)$ pairs from $\mathcal{D}_{\text{train}}$, denoted as $\mathcal{D}_{\text{train}}^{(b)}$. Using these samples, we construct B weak classifiers $\{f^{(b)}\}_{b=1}^B$. (Here and after, we

ignore the ε in $f^{(b)}$ for simple representation.) Each client only uploads once perturbed observation, thus preserving their privacy guarantee.

2. **Utility Evaluation.** For the N_1 clients in the evaluation set, we do not collect their perturbed observations. Instead, each evaluation client provides a noised binary indicator of misclassification, which is used to evaluate a classifier's performance. Specifically, we split the evaluation set into B subsets: $\mathcal{D}_{\text{eval}}^{(b)}$, $b = 1, \ldots, B$. For a client $i \in \mathcal{D}_{\text{eval}}^{(b)}$ with true observation (x_i, y_i), we compute

$$\hat{y}_i := I\big[f^{(b)}\{\mathcal{R}(x_i)\} > 0\big] \quad \text{and} \quad r_i := I(\hat{y}_i = y_i). \tag{6}$$

Each client then reports a privatized value r_i' using randomized response, where

$$\mathbb{P}(r_i' = r_i) = q = \frac{e^{\varepsilon_v}}{1 + e^{\varepsilon_v}} \quad \text{and} \quad \mathbb{P}(r_i' \neq r_i) = 1 - q. \tag{7}$$

Theorem 3 shows that this evaluation mechanism satisfies ε-LDP when $\varepsilon_v = \varepsilon$, and provides an unbiased estimate of the accuracy of the classifier $f^{(b)}$. This estimate can be taken as an evaluation of the utility of the dataset $\mathcal{D}_{\text{train}}^{(b)}$, for $b = 1, \ldots, B$.

Theorem 3 *Let $\mathcal{M}$ be the privacy mechanism mapping (x, y) to r' as described above. Then $\mathcal{M}$ satisfies ε_v-LDP. Furthermore, let $r^{(b)}$ be the true classification accuracy of $f^{(b)}$ under the target distribution P, and let $n_1^{(b)} = |\mathcal{D}_{\text{eval}}^{(b)}|$. Define*

$$\tilde{r}^{(b)} = \frac{\hat{r}^{(b)} + q - 1}{2q - 1}, \quad \text{where} \quad \hat{r}^{(b)} = \frac{\sum_{i \in \mathcal{D}_{\text{eval}}^{(b)}} r_i'}{n_1^{(b)}}. \tag{8}$$

Then $\mathbb{E}\big(\tilde{r}^{(b)}\big) = r^{(b)}$, and $\text{Var}\big(\tilde{r}^{(b)}\big) \leq \big(\frac{e^{\varepsilon_v}+1}{e^{\varepsilon_v}-1}\big)^2 (4\, n_1^{(b)})^{-1}$.

Theorem 3 indicates that as $n_1^{(b)}$ increases, $\tilde{r}^{(b)}$ converges more closely to the true classification accuracy of $f^{(b)}$ under LDP. Consequently, enlarging the evaluation set substantially enhances the precision of accuracy estimates and assists in selecting the better-performing weak classifiers. Since heavily perturbed data may limit the gains from training on more samples, these findings suggest allocating a greater proportion of clients to evaluation rather than training. Simulation results in Appendix A.3 provide evidence supporting this strategy. In addition, during the evaluation step, each client in $\mathcal{D}_{\text{eval}}^{(b)}$ evaluates only a single classifier, ensuring that their privacy budget remains ε by setting $\varepsilon_v = \varepsilon$. If a client were to evaluate multiple classifiers, the budget would need to be divided among them, potentially reducing the precision of each evaluation. Specifically, when evaluating k classifiers, the privacy budget must be allocated as $\varepsilon_v = \varepsilon/k$ for each evaluation, leading to an increase in $\text{Var}(\tilde{r}^{(b)})$ as k grows.

3.4 Model Reversal and Model Averaging

Under LDP, the added noise can be substantial, causing the weak classifiers $\{f^{(b)}\}_{b=1}^B$ to perform poorly. As shown in Proposition 2, it is possible for $g^{(\varepsilon)}(z)$ to fall below $1/2$,

indicating a negative dataset that performs worse than random guessing. We propose model reversal and model averaging to extract useful information from such seemingly uninformative data and improve classification performance. The implementation of these techniques builds on the unbiased utility evaluation established in Theorem 3.

Model Reversal. We first propose a novel procedure to enhance the performance of each weak classifier. Formally, for a weak classifier $f^{(b)}$, we define its reversed classifier as

$$f^{*(b)} = \begin{cases} -f^{(b)}, & \text{if its estimated accuracy } \tilde{r}^{(b)} < 0.5, \\ f^{(b)}, & \text{otherwise.} \end{cases} \tag{9}$$

The model reversal process defined in Equation (9) means that for a given classifier $f^{(b)}$, its sign is reversed if its estimated accuracy $\tilde{r}^{(b)} < 0.5$; otherwise, it remains unchanged. Consequently, the estimated accuracy of the reversed classifier $f^{*(b)}$ becomes $\tilde{r}^{*(b)} = \max\{\tilde{r}^{(b)}, 1 - \tilde{r}^{(b)}\}$, ensuring it is larger than 0.5. This idea exploits the fact that classification primarily depends on the direction of the coefficient vector relative to the decision boundary, rather than its magnitude. By identifying whether a classifier is less than random guessing, we can flip its decision boundary to ensure it surpasses 50% accuracy. Here we focus on binary classification, where model reversal reduces to a simple "flip or keep" decision. Extending the MR idea to multi-class problems is an interesting direction for future work. For example, one could reduce a K-class problem to K one-vs-rest binary subproblems, or apply permutation-based corrections to the predicted class labels of each weak classifier, guided by confusion matrices estimated from suitably designed LDP-protected queries in the evaluation step.

Model Averaging. Next, given B reversed weak classifiers $\{f^{*(b)}\}_{b=1}^{B}$ with estimated accuracies $\{\tilde{r}^{*(b)}\}_{b=1}^{B}$, we combine them by weighted averaging:

$$w_b = \frac{\max\{\tilde{r}^{*(b)} - r_0, 0\}}{\sum_{j=1}^{B} \max\{\tilde{r}^{*(j)} - r_0, 0\}}, \tag{10}$$

where $r_0 \in (0.5, 1)$ is a server-specified cutoff. Classifiers with $\tilde{r}^{*(b)} \le r_0$ receive weight 0, and the final averaged classifier is

$$f^{\dagger}(\cdot) = \sum_{b=1}^{B} w_b f^{*(b)}(\cdot). \tag{11}$$

In Equation (11), we aggregate reversed weak classifiers by a weighted average. For parametric classifiers where $f^{*(b)}(\cdot)$ is linear in the model parameters, Equation (11) is equivalent to averaging the corresponding parameter vectors. More generally, when the weak classifiers are nonlinear in their parameters or do not share a common parameterization, the same utility-based weights $\{w_b\}$ from Equation (10) can be combined with alternative aggregation rules, such as weighted voting or averaging predicted probabilities or decision scores.

The effectiveness of MR and MA relies on accurate utility estimation. Theorems 5 and 6 show that these techniques improve excess risk bounds. Our experiments further confirm that MR and MA significantly enhance classification accuracy.

3.5 Single-Server Classification with MRMA

This subsection integrates the previously introduced components, privacy-preserving data collection, utility evaluation, model reversal, and model averaging, into a unified procedure for classifier construction on a single server. The full process is presented in Algorithm 1, which takes as input the randomized mechanism, privacy budget, and learning parameters, and returns a final classifier trained under LDP.

Algorithm 1 Single-Server Classification with MRMA

1: **procedure** SERVER($\mathcal{R}, N_0, N_1, n_0, B, \varepsilon, \varepsilon_z, \varepsilon_y, r_0$)
2:     Divide clients into a training set $\mathcal{D}_{\text{train}}$ and an evaluation set $\mathcal{D}_{\text{eval}}$ with $|\mathcal{D}_{\text{train}}| = N_0, |\mathcal{D}_{\text{eval}}| = N_1$, and spilt the evaluation set into B subsets, $\{\mathcal{D}_{\text{eval}}^{(b)}\}_{b=1}^B$.
3:     **for** each client in the training set **do**
4:         Obtain $(z_i^{(\varepsilon_z)}, y_i^{(\varepsilon_y)})$ by Equation (1).   ▷ privacy-preserving data collection
5:     **end for**
6:     **for** $b = 1, \ldots, B$ **do**
7:         Randomly draw n_0 samples from $\mathcal{D}_{\text{train}}$ without replacement and denote as $\mathcal{D}_{\text{train}}^{(b)}$.
8:         Build a classifier $f^{(b)}$ based on $\{(z_i^{(\varepsilon_z)}, y_i^{(\varepsilon_y)})\}_{i \in \mathcal{D}_{\text{train}}^{(b)}}$. ▷ server-specified classifier
9:         **for** each client in the evaluation set $\mathcal{D}_{\text{eval}}^{(b)}$ **do**
10:             Compute r_i' by Equations (6) and (7).
11:         **end for**
12:         Estimate $\tilde{r}^{(b)}$ by Equation (8).   ▷ utility evaluation
13:         Compute $f^{*(b)}$ by Equation (9).   ▷ model reversal
14:     **end for**
15:     Estimate $\boldsymbol{w} = (w_1, \ldots, w_B)^\top$ by Equation (10) with cutoff value r_0.
16:     **return** The final estimated classifier $f^\dagger$ by Equation (11).   ▷ model averaging
17: **end procedure**

Sample Size Balancing. In practice, when the total sample size N is large, increasing the training size N_0, the evaluation size N_1, and the number of classifiers B can improve performance. However, for a fixed N, it is important to carefully choose N_0, N_1 and B. Larger values of N_0 and B help train more effective weak classifiers, while smaller values ensure sufficient data in each evaluation subset. Section 4 provides theoretical guidance on setting these parameters. We also empirically study their impact in Appendix A.2 and A.3, offering practical recommendations.

Broad Applicability. While this work focuses on a specific algorithmic structure, the proposed evaluation process and MRMA techniques are broadly applicable. Regarding data types, our framework accommodates vectors, functional data, images, or text, depending on the representation function and classifier specified at line 8 in Algorithm 1. Regarding privacy mechanisms, MRMA is compatible with various LDP variants by adjusting how noise is added in the training and evaluation phases, as implemented in lines 4 and 10 of Algorithm 1, respectively. This flexibility enables the method to adapt to a wide range of practical applications with different data modalities and privacy requirements.

Online data. Although our primary focus is on protecting static information such as individual profiles, many real-world systems produce continuous data streams from sensors or digital interactions. Our proposed procedure naturally extends to such online settings. On one hand, accuracy estimates for existing weak classifiers can be updated incrementally, allowing dynamic adjustment of their weights in the aggregated model. On the other hand, if performance deteriorates on new data, additional weak classifiers can be trained on newly collected, privatized samples. As a result, the final classifier can adapt over time by incorporating fresh models and reweighting existing ones, effectively tracking changes in the underlying data distribution.

3.6 Multi-Server Classification with MRMA under Heterogeneity

In many real-world scenarios, data are distributed across multiple servers. However, these servers may exhibit varying degrees of heterogeneity in their local data, which existing methods do not always handle effectively. Our proposed techniques extend naturally to the multi-server setting under LDP. Assume there are K servers, each with a locally trained classifier $f_k(x)$. These servers share their models with one another. Each server can then re-evaluate the shared classifiers and apply MRMA, as described in Sections 3.3 and 3.4, treating classifiers from different servers as a collection of weak classifiers. Details of model evaluation and MRMA in this setting are provided in the following two paragraphs.

Perturbation and Evaluation. A server with sufficient clients can split its evaluation set into $B + K$ parts, using B parts for evaluating its internal weak classifiers and K parts for evaluating the externally received classifiers $\{f_k\}_{k=1}^{K}$. Each evaluation is performed via randomized response at privacy level ε, as in Section 3.3. When data are limited, one can sequentially evaluate internal and external classifiers, dividing the privacy budget accordingly (by splitting ε appropriately).

MRMA. Upon receiving performance assessments for all K classifiers from multiple servers, each server can independently apply MR and MA with a cutoff r_0^* (see Equation (10) in Section 3.4) and obtain the final classifier $f_k^{\dagger}$ for the k-th server. Because the servers and their data distributions may be highly heterogeneous, employing our MRMA approach can mitigate negative transfer effects by assigning small or zero weights to less relevant models. Overall, by integrating LDP with MR and MA, each server can capitalize on the shared knowledge from others, achieving improved classification performance while still honoring the local privacy constraints.

Our discussion of the multi-server setting focuses on how locally trained ε-LDP classifiers can be re-evaluated and then combined via MRMA to improve each server's final decision rule. In practice, distributed learning across multiple servers can involve additional considerations that are widely studied in the federated learning literature (Truex et al., 2020; Sun et al., 2021). Our evaluation and MRMA procedures can be used in such settings, as they only require a collection of candidate ε-LDP classifiers and their utility estimates.

4. Theoretical Guarantees for Classification under LDP

In this section, we analyze how the LDP mechanism affects classification performance and demonstrate the effectiveness of our proposed techniques through their improvements in

excess risk bounds. We begin with necessary definitions for Bayes classifiers and proceed to establish our theoretical guarantees.

Recall that $\eta(z)$ denotes the conditional probability function, and the corresponding Bayes classifier under the original (unperturbed) distribution P is given by $h^*(z) = \text{sign}\{\eta(z) \geq 1/2\}$. The expected classification accuracy of h^* is $\frac{1}{2} + \mathbb{E}_{P_Z}(|\eta(z) - \frac{1}{2}|)$. We measure the performance of a classifier $h(z)$ under P relative to $h^*(z)$ using the excess risk, defined as:

$$L_P(h, h^*) = \mathbb{E}_P\big[\ell(h) - \ell(h^*)\big],$$

where $\ell(h) = I\{y\, h(z) \leq 0\}$ is the 0–1 loss and $I\{\cdot\}$ denotes the indicator function. Similarly, for the perturbed distribution $P^{(\varepsilon)}$ with conditional probability function $\eta^{(\varepsilon)}(z)$, the corresponding Bayes classifier is $h^{*(\varepsilon)}(z) = \text{sign}\{\eta^{(\varepsilon)}(z) \geq 1/2\}$. The excess risk of a classifier $h(z)$ under $P^{(\varepsilon)}$ is defined as:

$$L_{P^{(\varepsilon)}}\big(h, h^{*(\varepsilon)}\big) = \mathbb{E}_{P^{(\varepsilon)}}\big[\ell(h) - \ell(h^{*(\varepsilon)})\big].$$

Let $f^{(b)}(z)$ be a weak classifier trained from the noised data set $\{(z_i^{(\varepsilon_z)}, y_i^{(\varepsilon_y)})\}_{i=1}^{n_0}$. The next theorem bounds the excess risk of $f^{(b)}(z)$ under the unperturbed distribution P.

Theorem 4 *Let $\mathcal{R}$ be a fixed representation function from $\mathcal{X}$ to $\mathcal{Z}$. Then for any weak classifier $f^{(b)}$ trained with n_0 samples from $P^{(\varepsilon)}$,*

$$
\begin{aligned}
L_P\big(f^{(b)}, h^*\big) \leq{}& L_{P^{(\varepsilon)}}\big(f^{(b)}, h^{*(\varepsilon)}\big) + \frac{3}{4} d_{\text{TV}}(P_Z, P_Z^{(\varepsilon)}) + \mathbb{E}_{P_Z}\Big[\big|\eta(z) - \eta^{(\varepsilon)}(z)\big|\Big] \\
& + \mathbb{E}_{P_Z}\big[\big|\eta(z) - \tfrac{1}{2}\big| - \big|\eta^{(\varepsilon)}(z) - \tfrac{1}{2}\big|\big],
\end{aligned}
$$

where

$$d_{\text{TV}}(P_Z, P_Z^{(\varepsilon)}) = 2 \sup_{B \subseteq \mathcal{Z}} \Big| P_Z(B) - P_Z^{(\varepsilon)}(B) \Big|$$

is the total variation distance between P_Z and $P_Z^{(\varepsilon)}$.

In Theorem 4, the term $L_{P^{(\varepsilon)}}\big(f^{(b)}, h^{*(\varepsilon)}\big)$ captures the excess risk of $f^{(b)}$ under the privatized distribution $P^{(\varepsilon)}$. Since $f^{(b)}$ is trained on data from $P^{(\varepsilon)}$, this term reflects the inherent difficulty of the learning task under $P^{(\varepsilon)}$ and depends on the representation function $\mathcal{R}$, the choice of classification algorithm, and the training sample size n_0; see related theoretical studies in Blanchard et al. (2008); Sang et al. (2022); Ko et al. (2023). The following three terms quantify discrepancies between the original and privatized distributions and do not depend on the training sample size n_0. The total variation distance term $d_{\text{TV}}(P_Z, P_Z^{(\varepsilon)})$ measures the discrepancy between the marginal distributions of Z and $Z^{(\varepsilon_z)}$. Similar excess risk bounds involving a total variation distance term can be found in Ben-David et al. (2006). To further illustrate the magnitude of this term, Appendix A.4 provides a numerical illustration showing how $d_{\text{TV}}(P_Z, P_Z^{(\varepsilon)})$ varies with the dimension d and the privacy level ε_z. The term $\mathbb{E}_{P_Z}\big[|\eta(z) - \eta^{(\varepsilon)}(z)|\big]$ quantifies how much the conditional distributions drift from η to $\eta^{(\varepsilon)}$, which can be seen as the discrepancy between the conditional distributions of $Y|Z$ and $Y^{(\varepsilon_y)}|Z$. The last term highlights the effect of deviation around the decision boundary $1/2$, quantifying the increased uncertainty from the original distribution to perturbed

distribution, where a narrower distance from $1/2$ means more uncertainty. A similar term appears in the excess risk bound derived by Qin et al. (2024).

Recall from Section 3.4 that model reversal flips a classifier's sign if it has estimated accuracy less than 0.5 under the original distribution P. Let n_1 be the sample size used for performance evaluation and $r^{(b)}$ be the true classification accuracy of $f^{(b)}$ under the distribution P. The following theorem shows how model reversal reduces the excess risk with high probability.

Theorem 5 *Let $\mathcal{R}$ be a fixed representation function from $\mathcal{X}$ to $\mathcal{Z}$. Suppose $f^{*(b)}$ is obtained by selecting between $\{f^{(b)}, -f^{(b)}\}$ via model reversal using n_1 evaluation samples. Then, with probability at least $\Phi\left(\sqrt{n_1}\,\frac{|r^{(b)}-1/2|}{\sqrt{r^{(b)}(1-r^{(b)})}}\right)$ (where Φ is the cumulative distribution function of normal distribution),*

$$L_P\big(f^{*(b)}, h^*\big) \le L_{P^{(\varepsilon)}}\big(f^{(b)}, h^{*(\varepsilon)}\big) + \frac{3}{4}\,d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)})$$
$$+\ 2\,\mathbb{E}_{P_Z}\Big[\big|\eta(z) - \tfrac{1}{2}\big| - \big|\eta^{(\varepsilon)}(z) - \tfrac{1}{2}\big|\Big].$$

Compared with the bound in Theorem 4, the term $\mathbb{E}_{P_Z}\big[|\eta(z) - \eta^{(\varepsilon)}(z)|\big] \in [0,1]$ is replaced in Theorem 5 by $\mathbb{E}_{P_Z}\Big[\big|\eta(z) - \tfrac{1}{2}\big| - \big|\eta^{(\varepsilon)}(z) - \tfrac{1}{2}\big|\Big] \in [0, 1/2]$. When all perturbed datasets are non-negative (i.e., those with $g^{(\varepsilon)}(z) < 1/2$), they share the same Bayes classifier as the original distribution, and $|\eta(z) - \eta^{(\varepsilon)}(z)| = |\eta(z) - \tfrac{1}{2}| - |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}|$. In this case, the excess risk bounds in Theorems 4 and 5 coincide. In contrast, when some datasets are negative, model reversal leads to a tighter bound, demonstrating its added benefit in such scenarios. Moreover, since dataset utility is typically unknown in practice, Theorem 5 guarantees that with a sufficiently large evaluation sample size n_1, model reversal can effectively mitigate the influence of less-informative data and improve overall model performance.

Next, we analyze the final averaged classifier $f^\dagger$ produced by combining B reversed weak classifiers $\{f^{*(b)}\}_{b=1}^B$ with weights determined by Equation (10). The following theorem provides a bound for its excess risk under P as $B \to \infty$.

Theorem 6 *Let $\mathcal{R}$ be a fixed representation function from $\mathcal{X}$ to $\mathcal{Z}$. For any $(z, y) \sim P$, let $F_z^{(\varepsilon)}$ denote the distribution of classification accuracy $\tilde{r}_z^{*(b)} = \mathbb{P}\{sign[f^{*(b)}(z)] = y\}$ under the random sampling of training and evaluation sets. Assume this distribution has support $[r_{z,0}, r_{z,1}]$ with non-degenerate mass near $r_{z,1}$. Let $f^\dagger$ be the model-averaged classifier defined in Equation (11), and let B_0 be the number of classifiers exceeding the cutoff r_0. If r_0 is chosen such that $B_0/B \to 0$ as $B \to \infty$, then*

$$\limsup_{B\to\infty} L_P\big(f^\dagger, h^*\big) \le L_{P^{(\varepsilon)}}\big(f^\dagger, h^{*(\varepsilon)}\big) + \frac{3}{4}\,d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)})$$
$$+\ 2\,\mathbb{E}_{P_Z}\Big[\big|\eta(z) - \tfrac{1}{2}\big| + \tfrac{1}{2} - r_{z,1}\Big], \quad \textit{in probability.}$$

This result shows that the excess risk of the model-averaged classifier is asymptotically bounded in probability bounded by a quantity that depends on the best-performing classifiers, i.e., those near the upper end $r_{z,1}$ of the accuracy distribution. As the number

of classifiers B increases, the weight assigned to lower-performing models vanishes under the thresholding scheme in Equation (10), leading the ensemble to concentrate on top-performing classifiers. This effectively implements a form of regularized extreme value selection. The convergence rate of the bound depends on the amount of probability mass that $F_z^{(\varepsilon)}$ places near $r_{z,1}$; see Leadbetter et al. (2012) for related results from extreme value theory. The third term in the bound reflects the remaining gap between the best achievable accuracy $r_{z,1}$ and the Bayes-optimal level $|\eta(z) - \frac{1}{2}| + \frac{1}{2}$. Compared to Theorem 5, this bound is tighter when high-quality classifiers exist and can be identified reliably through evaluation. These insights are consistent with our empirical findings in Section 6, where model averaging improves performance, especially in high-noise regimes.

While the above results focus on a single-server setting, they naturally extend to multi-server scenarios introduced in Section 3.6. In a heterogeneous multi-server environment, each server can treat other servers' classifiers as additional weak classifiers and apply model reversal and model averaging locally. The same type of bounds hold, subject to analogous conditions for data partition and accuracy estimation.

Deciding how to split data between the training and evaluation sets, as well as how many weak classifiers to train, is crucial. The theoretical results above provide guidance: a sufficiently large training sample size n_0 is necessary to ensure that each weak classifier is adequately learned, achieving lower excess risk $L_{P^{(\varepsilon)}}(f^{(b)}, h^{*(\varepsilon)})$; a larger evaluation sample size n_1 improves the precision of utility estimates, thereby strengthening model reversal; and a larger number of weak classifiers B leads to a more stable aggregated classifier, enhancing the effect of model averaging. Meanwhile, our empirical findings (see Appendix A.2) suggest that in high-noise regimes, it may be more beneficial to allocate more samples to evaluation rather than training.

5. Functional Data Classification under LDP

Advances in sensing technologies have made it increasingly feasible to collect data densely sampled over temporal or spatial domains, giving rise to what is commonly referred to as functional data (Ramsay and Silverman, 2005). Examples include physiological signals such as heart rate and electrocardiogram readings from wearable devices, GPS traces from smartphones that capture mobility patterns, and sensor data from activity trackers monitoring daily routines or sleep cycles. These data streams, while informative, can also expose sensitive information related to an individual's health, behavior, and lifestyle. This makes privacy preservation particularly critical in biomedical and health-monitoring applications, where temporal or spatial patterns may reveal personal characteristics or preferences (Stisen et al., 2015). Despite its importance, privacy-preserving learning with functional data remains largely unexplored. As a concrete demonstration of our approach, this section presents the full pipeline for constructing a classifier for functional data under LDP. To the best of our knowledge, this is the first framework for functional classification under LDP.

Suppose there is a single server with access to N clients. Each client holds a square-integrable functional covariate $x(t)$, defined on a compact domain $\mathcal{I} = [0, 1]$, and a corresponding binary label $y \in \{-1, 1\}$. Our goal is to learn a functional linear classifier

$$f(x) = \alpha + \int_0^1 x(t)\, \beta(t)\, dt,$$

such that we predict $\hat{y}(x) = \text{sign}\{f(x)\}$. In accordance with Yang et al. (2020), an LDP algorithm typically proceeds in four stages: encoding, perturbation, aggregation, and estimation. Below, we describe how each stage applies to functional data.

5.1 Encoding and Perturbation

For each client with a functional covariate $x(t)$ and label y in the training set, the data reported to the server is transformed through the following steps.

Dimensionality Reduction. Functional data are often observed on a fine grid of time points (or locations), leading to high-dimensional vectors. To reduce dimensionality in a privacy-preserving way, we project $x(t)$ onto a finite set of basis functions $\{\phi_k(t)\}_{k=1}^d$, which is corresponding to the representation process in Section 3.1. Concretely,

$$x(t) = \sum_{k=1}^{d} z_k \, \phi_k(t) + \xi(t), \tag{12}$$

where $\boldsymbol{z} = (z_1, \ldots, z_d)^\top \in \mathbb{R}^d$ are the basis coefficients and $\xi(t)$ is a residual term excluded from the model. The choice of basis functions $\Phi = (\phi_1, \ldots, \phi_d)$ (e.g., B-splines, Fourier bases) and the dimension d are specified by the server, where d controls the representation fidelity and also affects the magnitude of the subsequent privatization for a fixed privacy budget. This projection serves two key purposes:

1. Functional data fitting. In practice, $x(t)$ is discretely observed at T time points, $\boldsymbol{x} = (x(t_1), \ldots, x(t_T))^\top$. For smooth families like B-splines of order m, fitting accuracy improves at the optimal rate of $T^{-2m/(2m+1)}$ as the number of knots grows like $T^{1/(2m+1)}$ (Eubank, 1999), which indicates a relatively slow rate of increase for d.

2. Privacy enhancement. Truncating at d dimensions mitigates overfitting to person-specific variations, thereby enhancing privacy. It also significantly reduces communication overhead compared to transmitting the full functional data.

When the chosen basis cannot adequately represent the underlying signal, or when the classification rule is too complex to be well approximated by a basis expansion, the separate perturbation mechanism in Equation (1) may lead to substantial information loss. This mechanism can be viewed as inducing a form of measurement error in both covariates and labels, and classification performance may deteriorate further in such settings (Li and Ma, 2024). More generally, Equation (1) can be replaced by more sophisticated LDP mechanisms, together with the corresponding algorithms for constructing weak classifiers (Berrett and Butucea, 2019; Ma and Yang, 2024), while our proposed evaluation, model reversal, and model averaging procedures remain applicable.

Rescaling. To bound sensitivity for LDP, we rescale each component of z into a fixed range, typically $[-1, 1]$. We consider two simple examples:

$$\begin{aligned}
\text{(Tanh Transformation)} \quad z_k^* &= \tanh(z_k), \\
\text{(Max-Abs Normalization)} \quad z_k^* &= \frac{z_k}{\max_{k'}|z_{k'}|}.
\end{aligned}$$

Here, $\boldsymbol{z}^* = (z_1^*, \ldots, z_d^*)^\top \in [-1,1]^d$. The tanh transformation is applied componentwise and introduces a nonlinear squashing effect, whereas the Max-Abs normalization is a linear rescaling that preserves coefficient signs and the relative magnitudes across coordinates. Other bounded transformations can also be used to map the feature vector into a fixed bounded range and thereby control sensitivity under the chosen LDP mechanism. In Appendix A.1, we compare these two rescaling choices and observe similar classification accuracy in our experimental setting. Therefore, for concreteness, the experiments in Section 6 report results based on the tanh transformation.

Perturbation. After rescaling, each client adds noise according to an LDP mechanism. Using a Laplace mechanism (Dwork et al., 2006), they report

$$\boldsymbol{z}^{(\varepsilon_z)} = \boldsymbol{z}^* + \boldsymbol{\delta}^{(\varepsilon_z)}, \tag{13}$$

where $\boldsymbol{\delta}^{(\varepsilon_z)} \in \mathbb{R}^d$ has independent and identically distributed (i.i.d.) Laplace entries with scale $\lambda = d\Delta/\varepsilon_z$ and $\Delta = 2$ (since each $z_k^* \in [-1,1]$). Hence each coordinate's privacy budget is effectively ε_z/d. To protect the binary label y, we apply the randomized response mechanism (Warner, 1965):

$$\mathbb{P}\big(y^{(\varepsilon_y)} = y\big) = \frac{e^{\varepsilon_y}}{1 + e^{\varepsilon_y}}, \tag{14}$$

where $\varepsilon_y = \varepsilon - \varepsilon_z$. In the experiment, we choose $\varepsilon_y = \varepsilon/(d+1) = \varepsilon_z/d$. Theorem 7 shows that the entire encoding and perturbation procedure satisfies ε-LDP.

Theorem 7 *Let $\mathcal{M}_f$ be the privacy mechanism taking $(x(t), y)$ as input and returning $\big(\boldsymbol{z}^{(\varepsilon_z)}, y^{(\varepsilon_y)}\big)$ via Equations (13)–(14). Then $\mathcal{M}_f$ satisfies ε-LDP.*

It is worth noting that we employ the Laplace mechanism here due to its simplicity and ease of interpretation, which facilitates our subsequent demonstrations of the proposed methods. As shown in Duchi et al. (2018), adding Laplacian noise to (appropriately truncated) observations is an optimal privacy mechanism in one-dimensional mean estimation under LDP. However, this optimality does not directly extend to higher-dimensional settings. For additional insights on such mechanisms, we refer readers to Nguyên et al. (2016); Duchi et al. (2018); Wang et al. (2019a); Xiao et al. (2023).

5.2 Classifier Construction with MRMA

Once the server collects $\{(\boldsymbol{z}_i^{(\varepsilon_z)}, y_i^{(\varepsilon_y)})\}_{i=1}^{N_0}$ from N_0 training clients, it can train multiple weak classifiers as introduced in Section 3.3. Two general methods are outlined below.

Method I. It's straightforward to build a vector classifier by treating $\boldsymbol{z}_i^{(\varepsilon_z)} \in \mathbb{R}^d$ as the input to a standard classification model (e.g., logistic regression, SVM). Denote the estimated intercept and coefficients by $\hat{\alpha}$ and $\hat{\boldsymbol{b}} = (\hat{b}_1, \ldots, \hat{b}_d)^\top \in \mathbb{R}^d$, respectively. We then recover the slope function via $\widehat{\beta}(t) = \sum_{k=1}^d \hat{b}_k \phi_k(t)$, and define the corresponding functional classifier as

$$\hat{f}(x) = \hat{\alpha} + \int_0^1 x(t)\,\widehat{\beta}(t)\,dt.$$

Method II. Before building the classifier, we first reconstruct the function using the perturbed basis scores, i.e., treating $x_i^{(\varepsilon_z)}(t) = \sum_{k=1}^d z_{i,k}^{(\varepsilon_z)} \phi_k(t)$ as the perturbed functional

covariate, where $z_{i,k}^{(\varepsilon_z)}$ represents the kth element of $\boldsymbol{z}_i^{(\varepsilon_z)}$. Then, the classifier can be obtained using the functional conjugate gradient algorithm (CG, Kraus and Stefanucci, 2019), functional distance weighted discrimination (DWD, Sang et al., 2022), or any other established functional classification method.

With these two ways of constructing classifiers, in the Appendix A.1, we illustrate the effects of dimensionality reduction, rescaling, and perturbation on the misclassification rate of the classifiers. Experiment results indicate that the projection of functional data and rescaling of coefficient vectors have a small impact on the classifier's performance. And the performance of two types of transformations is similar in our context. Moreover, as the privacy budget $\varepsilon \to 0$, the misclassification rates of the weak classifiers based on the perturbed data tend to 50%.

MRMA. Based on the noised binary evaluations collected from the additional N_1 evaluation clients in Section 3.3, we apply the MRMA procedure introduced in Section 3.4. Specifically, for a weak classifier $f^{(b)}$, if its estimated accuracy $\tilde{r}^{(b)} < 50\%$, we invert its parameters:

$$(\hat{\alpha}^{(b)},\ \widehat{\beta}^{(b)}(t)) \ \longmapsto\ \left(-\hat{\alpha}^{(b)},\ -\widehat{\beta}^{(b)}(t)\right).$$

Let $\{f^{*(b)}\}_{b=1}^B$ be the reversed classifiers, and denote their weights by $\{w_b\}_{b=1}^B$ as in Equation (10). Our final averaged functional classifier becomes

$$f^\dagger(x) \ =\ \hat{\alpha}^\dagger \ +\ \int_0^1 x(t)\, \widehat{\beta}^\dagger(t)\, dt,$$

where $\hat{\alpha}^\dagger = \sum_{b=1}^B w_b\, \hat{\alpha}^{*(b)}, \widehat{\beta}^\dagger(t) = \sum_{b=1}^B w_b\, \widehat{\beta}^{*(b)}(t)$.

This example demonstrates how our LDP framework generalizes naturally to infinite-dimensional data via basis truncation, preserving both privacy and utility. It highlights the method's flexibility in accommodating complex data structures.

6. Experiments

In this section, we present experiments that primarily demonstrate the improvements in classification accuracy achieved by applying our proposed techniques, compared to baselines that do not use them. These results validate the effectiveness of our approach. Differences in performance across classifier types are due to their intrinsic properties and are not the focus of this study.

In each experimental trial, the functional covariate $X(\cdot)$ is generated by $X(t) = \sum_{j=1}^{50} \xi_j \zeta_j \phi_j(t)$ for $t \in [0,1]$, where ξ_j's are independently drawn from $U(-\sqrt{3}, \sqrt{3})$, $\zeta_j = (-1)^{j+1} j^{-1}, j = 1, \ldots, 50$, $\phi_1(t) = 1$ and $\phi_j(t) = \sqrt{2}\cos((j-1)\pi t)$ with $j \geq 2$. The binary response variable Y, taking values 1 or -1, is generated using the following logistic model:

$$f(X) = \alpha_0 + \int_0^1 X(t)\beta(t)dt, \quad \Pr(Y=1) = \frac{\exp\{f(X)\}}{1 + \exp\{f(X)\}},$$

where $\alpha_0 = 0.1$, $\beta(t)$ is the slope function, and $f(X)$ is referred to as the classification function. Results in Appendix A.1 show that the performance with $d = 4, 5, 6$ cubic B-Spline is comparable, and the performance based on Tanh or Max-Abs transformation is close. Thus we present results with $d = 4$ and employ the tanh transformation.

6.1 Single-Server Classification with MRMA

In this section, we demonstrate the improvements in classification accuracy brought about by model reversal and model averaging. We generate data for a server using the slope function $\beta(t) = \sum_{j=1}^{50} 4(-1)^{j+1} j^{-2} \phi_j(t)$. Assuming the server has a total of $N = 3000$ clients, we allocate $N_0 = 500$ for training and the remaining $N_1 = 2500$ for evaluation. To construct classifiers, we sequentially draw $n_0 = 50$ samples from the training data set without replacement, repeating this procedure $B = 50$ times. At the same time, we partition the evaluation set into $B = 50$ subsets of equal size, enabling us to evaluate the classification accuracy of each classifier using $n_1 = 50$ distinct samples. To assess the performance of the classifiers, we randomly generate a testing data set comprising 500 samples during each trial, repeating this procedure 500 times.

Figure 2 showcases the misclassification rates along with error bars, for various classifiers across different ε levels. In this figure, "Weak" denotes the average misclassification rate of $B = 50$ weak classifiers obtained through sampling. "MR" represents the average misclassification rate of B weak classifiers after model reversal under LDP. "MA" signifies the results when using model averaging on weak classifiers with cutoff value $r_0 = 0.8$, while "MRMA" illustrates the results of applying both model reversal and model averaging. To compare with classic ensemble methods, we train B weak classifiers under LDP. Each classifier is trained with $N/B = 60$ instances from the total sample. We then obtain the results through majority voting and equal-weight averaging, denoted as "Voting" and "Averaging", respectively. And "All data" denotes the classifier trained with N clients directly.

The classifier "All data", even if it is trained with 3000 clients, shows almost no improvement over the classifier "Weak" when ε is small (indicating substantial noise interference). And classifiers "Voting" and "Averaging" also perform similarly. However, our proposed techniques, both model reversal and model averaging, significantly improve the performance of all types of weak classifiers. And MRMA further enhances the performance of SVM and CG classifiers substantially. Figure A.5 in Appendix A.3 demonstrates that even when allocating more clients for training weak classifiers, MR and MA can still significantly enhance the classifiers' performance. From the client's perspective, this means that to achieve a target classification accuracy, our method allows for a higher level of privacy protection, that is, a smaller value of ε can be used without sacrificing performance. For further discussions regarding distinctions among different types of classifiers, please refer to Appendix A.1.

6.2 Multi-Server Classification with MRMA under Heterogeneity

In this section, we showcase the improvements in classification accuracy achieved through multi-server learning for individual servers. We consider a total of $K = 25$ servers, divided into three groups, each characterized by a distinct slope function to introduce heterogeneity. Specifically,

- Group 1: for $k = 1, \ldots, 10$, $\beta_k(t) = \sum_{j=1}^{50} \gamma_j (-1)^{j+1} j^{-2} \phi_j(t)$ with $\gamma_j \overset{i.i.d.}{\sim} U(-8, -2)$.

- Group 2: for $k = 11, \ldots, 15$, $\beta_k(t) \sim \mathrm{GP}(0, K(s, t))$ with $K(s, t) = \exp(-15|s - t|)$.

- Group 3: for $k = 16, \ldots, 25$, $\beta_k(t) = \sum_{j=1}^{50} \gamma_j (-1)^{j+1} j^{-2} \phi_j(t)$ with $\gamma_j \overset{i.i.d.}{\sim} U(2, 8)$.
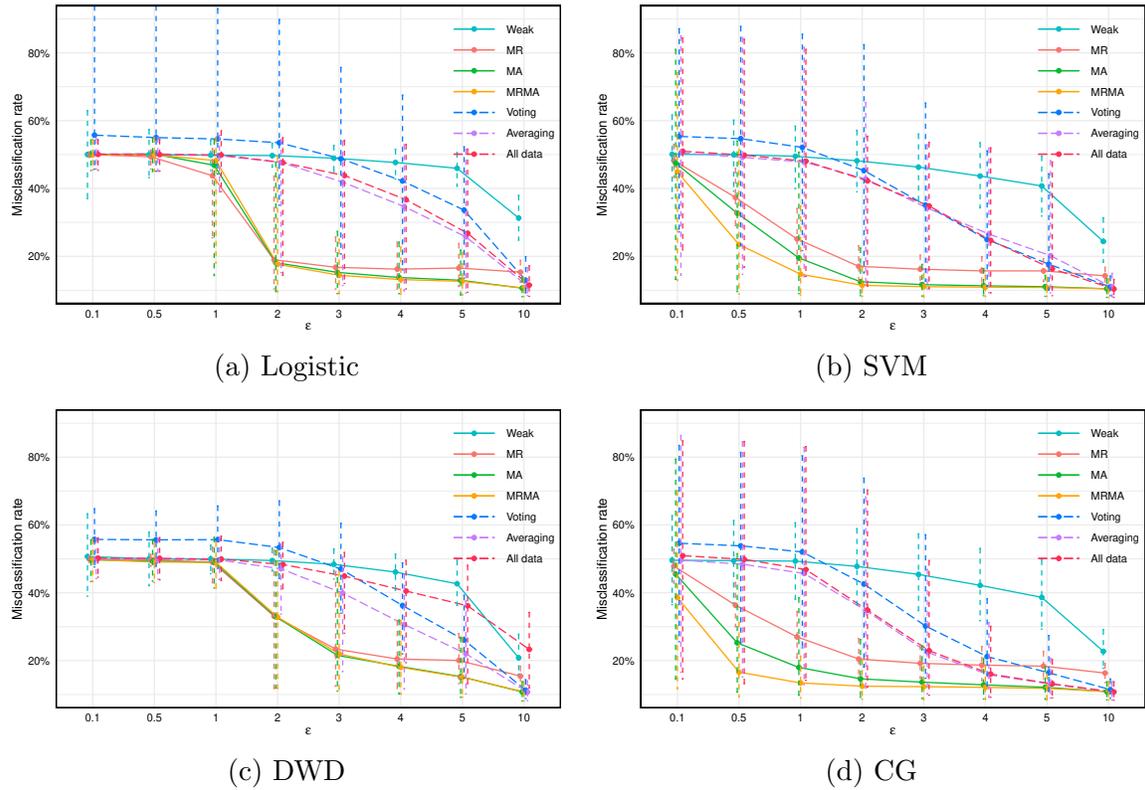
Figure 2: The misclassification rates of classifiers with a single server under ε-LDP.

Here, $GP(0, K(s, t))$ represents a Gaussian process with zero mean and kernel $K(s, t)$. It is essential to highlight that the slope functions of servers within the same group are not identical. Moreover, the directions of the slope functions in groups 1 and 3 are opposite, serving as a test to assess the potential negative transfer impact during multi-server learning. In group 2, the slope function is randomly generated, resulting in an approximate 50% misclassification rate for classifiers built on servers in this group. This design is purposefully implemented to gauge its potential disruption to the multi-server learning process.

For each server, we generate $N = 3000$ clients, with an additional 500 clients designated for testing. Each server employs algorithms with the parameters $N_0 = 500$, $N_1 = 2500$, $n_0 = 50$, and $B = 50$. Initially, each server independently obtains its own classifier using MRMA, with parameters set to $\varepsilon_v = \varepsilon$ and $r_0 = 0.8$. Subsequently, all servers proceed to perform multi-server learning. The parameters for this phase are set as $\varepsilon_v = \varepsilon_v^* = \varepsilon/2$, with $r_0^* = 0.8$ to mitigate potential negative transfer effects. As expected, servers in group 2 exhibit misclassification rates around 50%. The average misclassification rates for classifiers of servers in groups 1 and 3 under these two scenarios are illustrated in Figure 3. The results show that multi-server learning significantly improves the performance of both logistic and DWD classifiers, even with server heterogeneity. While SVM and CG classifiers

21

already perform well within a single server setting, multi-server learning shows comparable or slightly better performance.
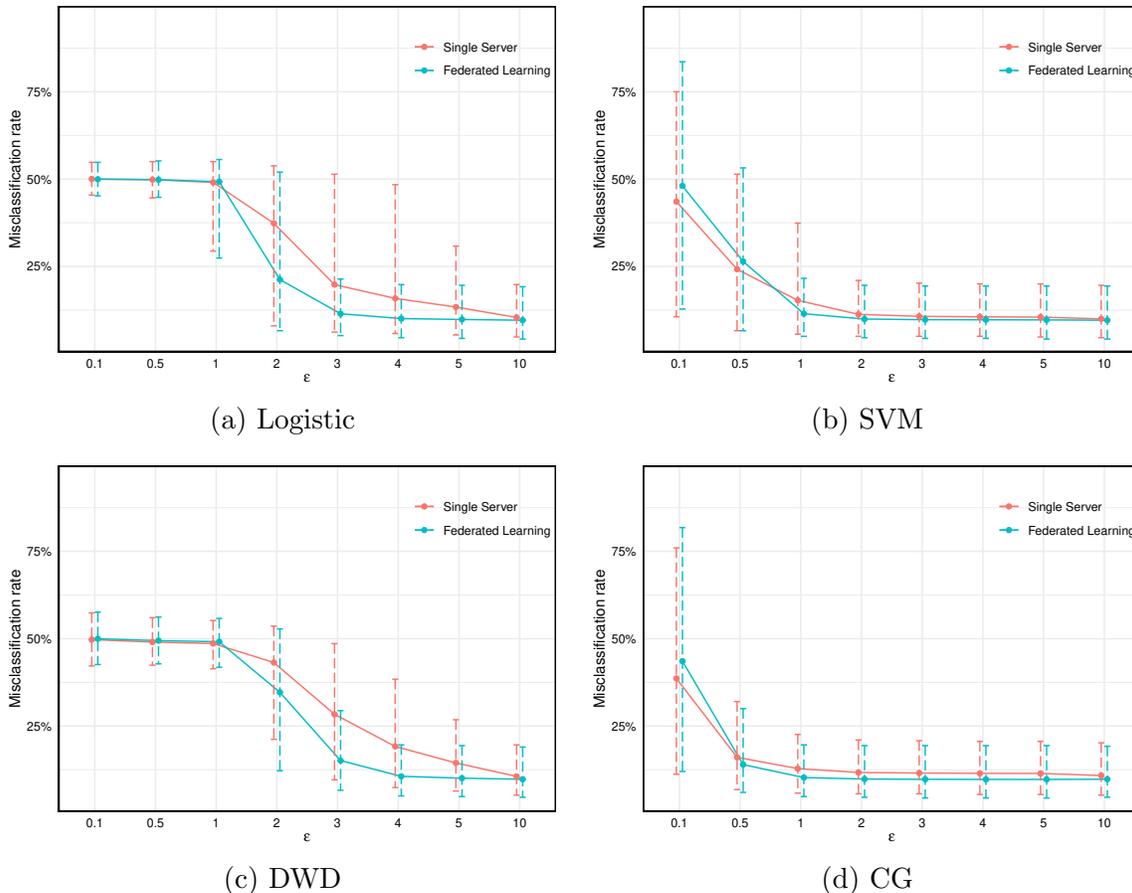

(a) Logistic

(b) SVM

(c) DWD

(d) CG

Figure 3: The misclassification rates of classifiers with multi-server under ε-LDP.

7. Real Application

To demonstrate the practical utility of our proposed classification methods under LDP, we evaluate their performance on both vector-valued and functional datasets, each involving sensitive information that necessitates privacy protection. Section 7.1 presents results from two real-world datasets with vector predictors related to diabetes and employee attrition, while Section 7.2 focuses on applications involving functional predictors derived from physical activity and speech recordings.

7.1 Vector-valued Predictors

We first evaluate our methods on two publicly available datasets: a healthcare dataset related to diabetes risk and an employee attrition dataset. Both datasets contain sensitive

information, making them suitable candidates for privacy-preserving classification. We compare our methods with the histogram-based classifier under LDP proposed by Berrett and Butucea (2019), referred to as "Histogram". We fix the number of splits to $h^{-1} = 4$ along each axis, since tuning h under LDP would require an additional privacy-preserving validation step and further complicate the procedure.

The Diabetes dataset, published by Pore (2023), comprises 2768 samples and includes eight health-related attributes used to predict whether an individual has diabetes. We randomly split the dataset into 2214 training samples (approximately 80%) and 554 testing samples. To assess classifier performance under LDP, we repeat this process 500 times. In each repetition, we draw $N_0 = 414$ samples from the training samples to serve as the training set, while the remaining $N_1 = 1800$ samples are used for evaluation. Specifically, we randomly sample $n_0 = 60$ observations (without replacement) from the training set to construct a classifier, repeating this procedure $B = 30$ times to generate multiple weak classifiers. Then we partition the evaluation set into $B = 30$ subsets of equal size, enabling us to evaluate the classification accuracy of each classifier using $n_1 = 60$ distinct samples.

The Employee dataset, introduced by Elmetwally (2023), contains employment-related records for 4653 individuals, each described by eight variables, including education, job history, and workplace factors. The task is to predict whether an employee leaves the company. We split the dataset into 3722 training samples and 931 testing samples, again repeating the process 500 times. Within each repetition, we use $N_0 = 722$ for training and $N_1 = 3000$ for evaluation. Classifier construction and evaluation follow the same procedure as described for the Diabetes dataset, using $n_0 = n_1 = 100$ and $B = 30$.

Tables 1 and 2 report the average misclassification rates and standard deviations across 500 repetitions under various privacy levels (ε) for both datasets. We set $r_0 = 0.7$ to ensure the existence of weak classifiers satisfying this criterion. Our MRMA-based classifier consistently achieves superior accuracy, particularly when ε is small, corresponding to stronger privacy guarantees. As ε increases, indicating weaker privacy constraints, the performance differences among our proposed methods, the ensemble strategies ("Voting" and "Averaging"), and the "All data" method become negligible, with all achieving similar accuracy and significantly outperforming the histogram-based approach. Notably, even when $\varepsilon = 1000$, the histogram method remains less effective, likely due to the substantial Laplace noise introduced, whose scale is $2^{d+1}/\varepsilon$, where $d = 8$ is the number of predictors. This level of noise, although reduced for large ε, still introduces enough distortion to degrade classification performance.

7.2 Functional Predictors

We evaluate our LDP methods on two real-world classification tasks involving functional predictors. The first task uses wearable-sensor-based physical activity data to predict cardiovascular health status, while the second revisits the phoneme classification problem using frequency-domain speech features.

The first task draws on data from the National Health and Nutrition Examination Survey 2013–2014 cycle. We consider adult female participants aged 20 years or older, for whom both laboratory-measured high-density lipoprotein (HDL) cholesterol and minute-level physical activity recordings are available. HDL is known to be protective against

ε	Weak	MR	MA	MRMA	Voting	Averaging	All data	Histogram
0.1	49.55	48.22	46.22	**43.44**	55.25	49.03	50.82	49.75
	(0.078)	(0.032)	(0.153)	(0.144)	(0.221)	(0.157)	(0.158)	(0.057)
0.5	49.23	40.23	37.32	**34.49**	53.93	45.61	46.33	49.75
	(0.066)	(0.031)	(0.095)	(0.030)	(0.226)	(0.152)	(0.153)	(0.057)
1	48.48	34.90	34.59	**34.31**	48.09	41.50	41.57	49.75
	(0.055)	(0.020)	(0.033)	(0.018)	(0.197)	(0.132)	(0.132)	(0.057)
5	42.95	34.16	**34.31**	**34.31**	34.38	**34.31**	**34.31**	49.64
	(0.047)	(0.017)	(0.018)	(0.018)	(0.019)	(0.018)	(0.018)	(0.057)
10	38.06	**34.01**	34.29	34.30	34.31	34.31	34.31	49.58
	(0.030)	(0.018)	(0.018)	(0.018)	(0.018)	(0.018)	(0.018)	(0.058)
1000	26.60	26.02	23.03	23.03	22.81	22.23	**22.05**	39.80
	(0.010)	(0.013)	(0.018)	(0.018)	(0.016)	(0.016)	(0.015)	(0.036)

Table 1: Mean and standard deviation (in parentheses) of misclassification rates for various classification methods, computed over 500 random sample splits of the Diabetes dataset and evaluated under different values of ε for ε-LDP, with bold values indicating the best performance.

ε	Weak	MR	MA	MRMA	Voting	Averaging	All data	Histogram
0.1	49.71	47.86	46.43	**44.15**	58.23	48.06	48.25	49.98
	(0.082)	(0.031)	(0.152)	(0.145)	(0.230)	(0.155)	(0.155)	(0.025)
0.5	49.02	37.73	35.61	**34.43**	52.22	43.50	43.63	49.98
	(0.062)	(0.027)	(0.061)	(0.014)	(0.207)	(0.141)	(0.142)	(0.025)
1	48.04	34.64	34.48	**34.43**	45.30	38.73	39.08	49.98
	(0.053)	(0.015)	(0.020)	(0.014)	(0.167)	(0.107)	(0.109)	(0.025)
5	41.19	**34.36**	34.39	34.42	34.44	34.43	34.43	49.97
	(0.040)	(0.017)	(0.015)	(0.015)	(0.014)	(0.014)	(0.014)	(0.025)
10	36.61	**34.01**	34.18	34.16	34.43	34.43	34.43	49.96
	(0.020)	(0.018)	(0.018)	(0.018)	(0.014)	(0.014)	(0.014)	(0.025)
1000	32.43	31.62	29.69	29.69	30.21	29.51	**29.48**	47.61
	(0.009)	(0.013)	(0.017)	(0.017)	(0.014)	(0.014)	(0.014)	(0.024)

Table 2: Mean and standard deviation (in parentheses) of misclassification rates for various classification methods, computed over 500 random sample splits of the Employee dataset and evaluated under different values of ε for ε-LDP, with bold values indicating the best performance.

cardiovascular disease, and its levels are influenced by physical activity. According to clinical guidelines, HDL is considered healthy for women when its concentration is at least 50 mg/dL (Turk et al., 2009). Using this threshold, we define a binary classification task to distinguish participants with healthy versus unhealthy HDL levels. As the functional predictor, we use hour-level Monitor-Independent Movement Summary (MIMS) values averaged over seven days and projected onto a set of 6 cubic B-spline basis functions. The resulting dataset includes 2410 samples, with 1558 labeled as healthy and 852 as unhealthy. To evaluate

classifier performance under LDP, we adopt a randomized split: 410 samples are held out as a test set, and the remaining 2000 are divided into a training set of size $N_0 = 500$ and an evaluation set of size $N_1 = 1500$. For each repetition, we subsample $n_0 = n_1 = 50$ observations to train and evaluate classifiers across $B = 30$ randomized subsets. This entire process is repeated 500 times.
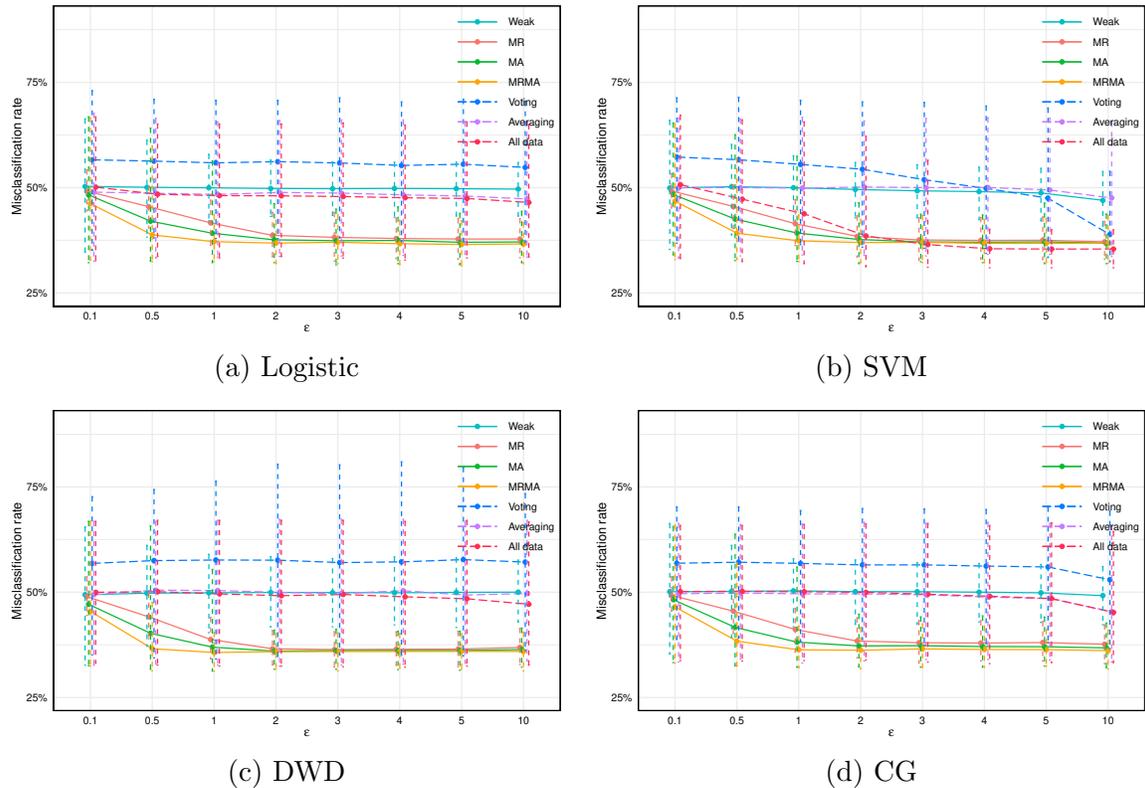

(a) Logistic

(b) SVM

(c) DWD

(d) CG

Figure 4: The misclassification rates of classifiers with a single server under ε-LDP on the physical activity dataset.

The second task uses the Phonemes dataset derived from the TIMIT Acoustic-Phonetic Continuous Speech Corpus (Garofolo, 1993). Speech frames in this dataset are extracted from the continuous speech of 50 male speakers, and log-periodograms are constructed from recordings sampled at 256 equispaced frequencies and projected onto a set of 4 Fourier basis functions. The classification task focuses on distinguishing between the phonemes "sh" and "iy", represented by 1163 and 872 curves, respectively. The log-periodogram functions are approximated using Fourier basis expansion. As in the MIMS example, we adopt a randomized split of 535 test samples, a training set of $N_0 = 300$, and an evaluation set of $N_1 = 1200$, with $n_0 = n_1 = 50$, and $B = 24$, repeating the entire process 500 times.

Visualizations of the functional predictors in these two applications are provided in Figures A.7-A.8 in the Appendix. These figures display both the raw functional observations and the curves reconstructed after finite basis projection and transformation. Despite some
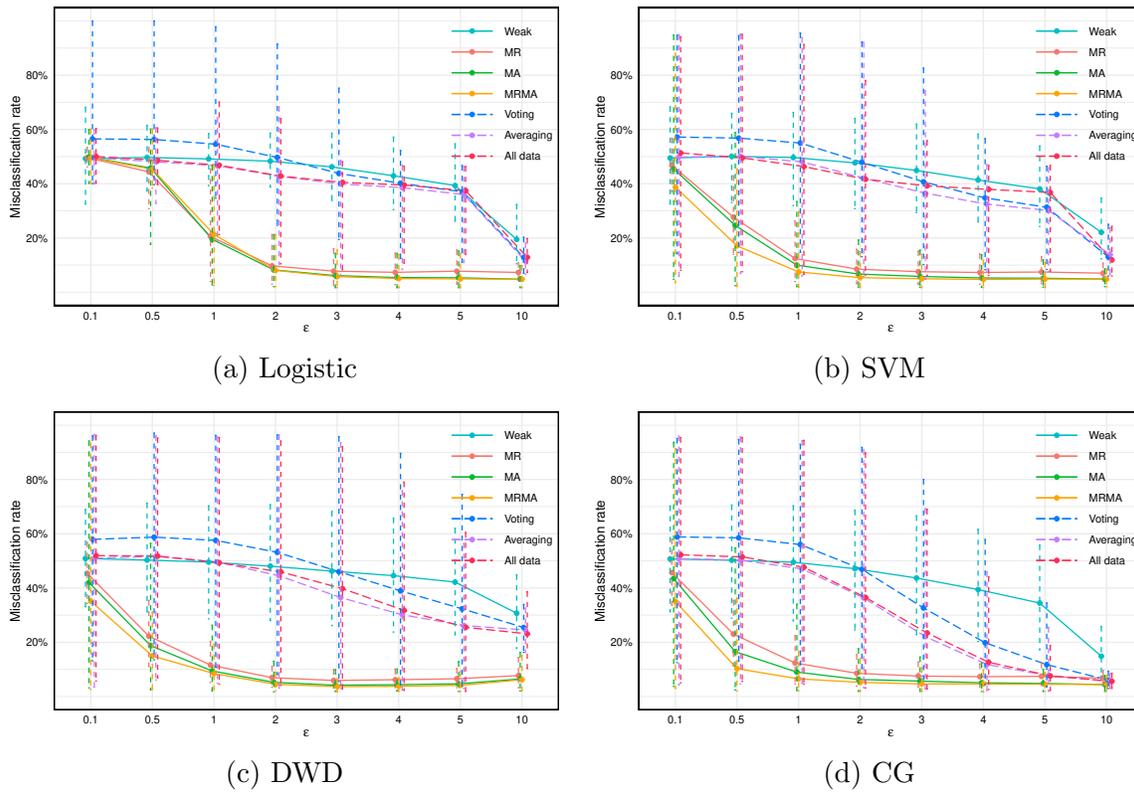
Figure 5: The misclassification rates of classifiers with a single server under ε-LDP on the Phonemes dataset.

smoothing and loss of fine-grained detail, the figures demonstrate that the key temporal patterns and population-level group differences are well preserved. Figures 4-5 report their misclassification rates. The parameter r_0 is set to 0.7 for the physical activity dataset and 0.9 for the Phonemes dataset. In both examples, classifiers based on model reversal and model averaging achieve substantial performance gains compared to other classifiers.

8. Conclusion

We proposed a novel framework for classification under LDP, addressing key challenges in data utility and model performance. By reinterpreting private learning as a transfer learning problem, we introduced a utility evaluation mechanism based on privatized binary feedback, enabling accurate assessment of classifier performance without accessing unperturbed data. Building on this, we developed two techniques: model reversal, which rescues underperforming classifiers by inverting decision boundaries, and model averaging, which combines multiple weak classifiers using utility-based weights. Together, these methods form a robust strategy, MRMA, that significantly enhances classification accuracy under LDP constraints. We provided theoretical excess risk bounds demonstrating the effectiveness of these techniques, and showed that the framework generalizes naturally to functional data settings

through basis projection. Empirical evaluations on both simulated and real-world datasets confirmed substantial improvements over baseline approaches. Future directions include further theoretical analysis of specific classifiers under particular LDP mechanisms, as well as extending the framework to multi-class classification, regression, and other structured data modalities under LDP constraints.

Acknowledgments

Appendix A. Additional Results

In this section, we provide additional results from our experiments to demonstrate the impacts of encoding and perturbation, compare various sample allocation strategies, and confirm that the performance of weak classifiers improves slightly with an increase in training sample size.

A.1 Encoding and Perturbation

In this section, we discuss the effects of dimensionality reduction, rescaling, and perturbation introduced in Section 5.1, on the misclassification rates of different types of classifiers.

To investigate the impacts of dimensionality reduction and rescaling, we generate data based on the model described in Section 6, and the sample sizes of the training and testing data sets are 50 and 500, respectively. During the dimensionality reduction, cubic B-spline with equidistant knots are employed, and we apply the methods for varying numbers of basis functions, $d = 4, 5, 6$. Table A.1 showcases the misclassification rates of classifiers based

		$d = 4$			$d = 5$			$d = 6$		
	INI	Coefs	Tanh	MAN	Coefs	Tanh	MAN	Coefs	Tanh	MAN
Logistic	-	17.81	12.32	11.68	20.66	13.55	12.81	23.19	15.21	13.86
SVM	-	10.85	11.20	11.03	11.29	11.67	11.27	11.69	12.07	11.48
DWD	11.01	11.02	10.64	10.61	11.01	10.60	10.58	11.01	10.61	10.59
CG	15.42	11.02	11.47	11.31	11.34	11.86	11.43	11.65	12.00	11.64

Table A.1: The misclassification rate of classifiers based on actual data (INI), coefficients obtained after dimensionality reduction (Coefs), and coefficients rescaled by either the Tanh transformation (Tanh) or Max-Abs normalization (MAN).

on the actual data (INI), coefficients obtained after dimensionality reduction (Coefs), and coefficients rescaled by either the Tanh transformation (Tanh) or Max-Abs normalization (MAN). This is based on the results from 500 repeated experiments.

Table A.1 shows that there are slight variations in the performance of different types of classifiers. Overall, classifiers with $d = 4, 5, 6$ exhibit comparable results, and the impact of dimension reduction and rescaling on classifier performance is small. Also, the performances based on Tanh and Max-Abs transformations are very similar. Furthermore, the CG classifier based on actual data and the logistic classifier based on coefficients obtained after dimensionality reduction perform relatively poorer. This may be related to the inherent characteristics of the classifiers, which is beyond the scope of this paper. Our primary focus is on the changes in classifier performance before and after consider LDP and the improvements brought about by different techniques.

To further explore the impacts of perturbation, we generate data in accordance with the model in Section 6, and both the training and testing data set sample sizes are 500. Table A.1 demonstrates that classifiers with $d = 4, 5, 6$ exhibit comparable results.Therefore, we employ $d = 4$ cubic B-spline with equidistant knots to introduce as little noise as possible. We consider eight distinct privacy budget levels, specifically, $\epsilon = 0.1, 0.5, 1, \ldots, 5, 10$. Figure A.1 displays the misclassification rates of the four types of classifiers, based on both unperturbed rescaled coefficients (i.e., "Tanh", "MAN") and perturbed rescaled coefficients across various levels of ϵ.
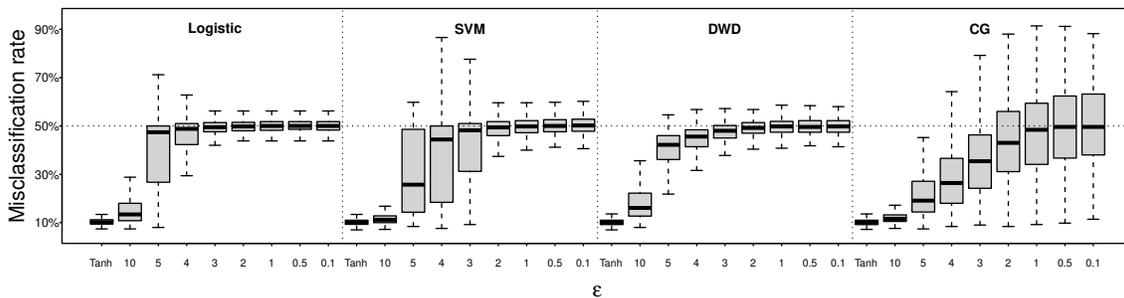
(a) Tanh transformation
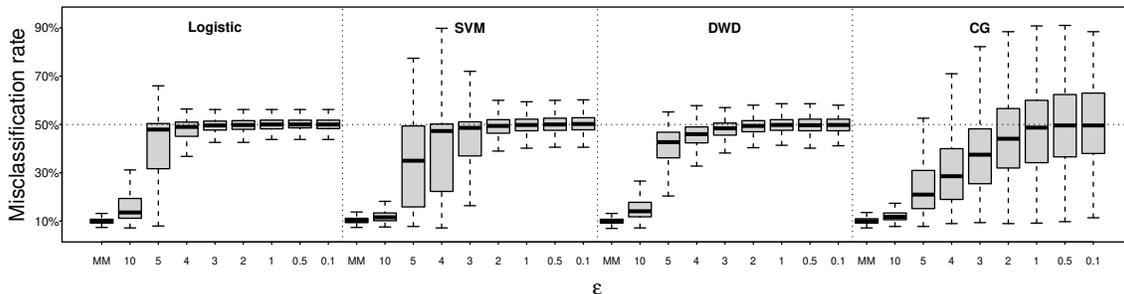
(b) Max-Abs transformation

Figure A.1: The boxplot of the misclassification rates of classifiers with Tanh and Max-Abs transformations under ε-LDP.

28

In Figure A.1, with decreasing ε, the misclassification rates of different types of classifiers tend to 50%. At the same time, there is a variation in the performance of classifiers under the influence of noise. Notably, the misclassification rate of CG classifier remains highly volatile at smaller ε values, ranging between 10% and 90%, instead of converging around 50% as other methods do. This indicates that the performance can be enhanced through model reversal and model averaging. Additionally, it can be observed from Figure A.1 that classifiers based on both Tanh and Max-Abs transformations exhibit very similar performances. Only results based on the Tanh transformation will be presented hereafter.

In Theorem 5, we quantify how model reversal improves the excess risk bound of a classifier. In Figure A.2, we show the empirical distributions of classification accuracy for the different classifiers under our experimental settings. Figure A.2 illustrates that different classifiers exhibit varying degrees of sensitivity to noise. Among them, the DWD classifier is the most affected, followed by logistic and SVM classifiers. In contrast, the CG classifier is relatively less impacted by noise interference. As ε decreases, which corresponds to increased noise, the classification accuracy distributions for logistic, SVM, and DWD classi-

(a) Logistic
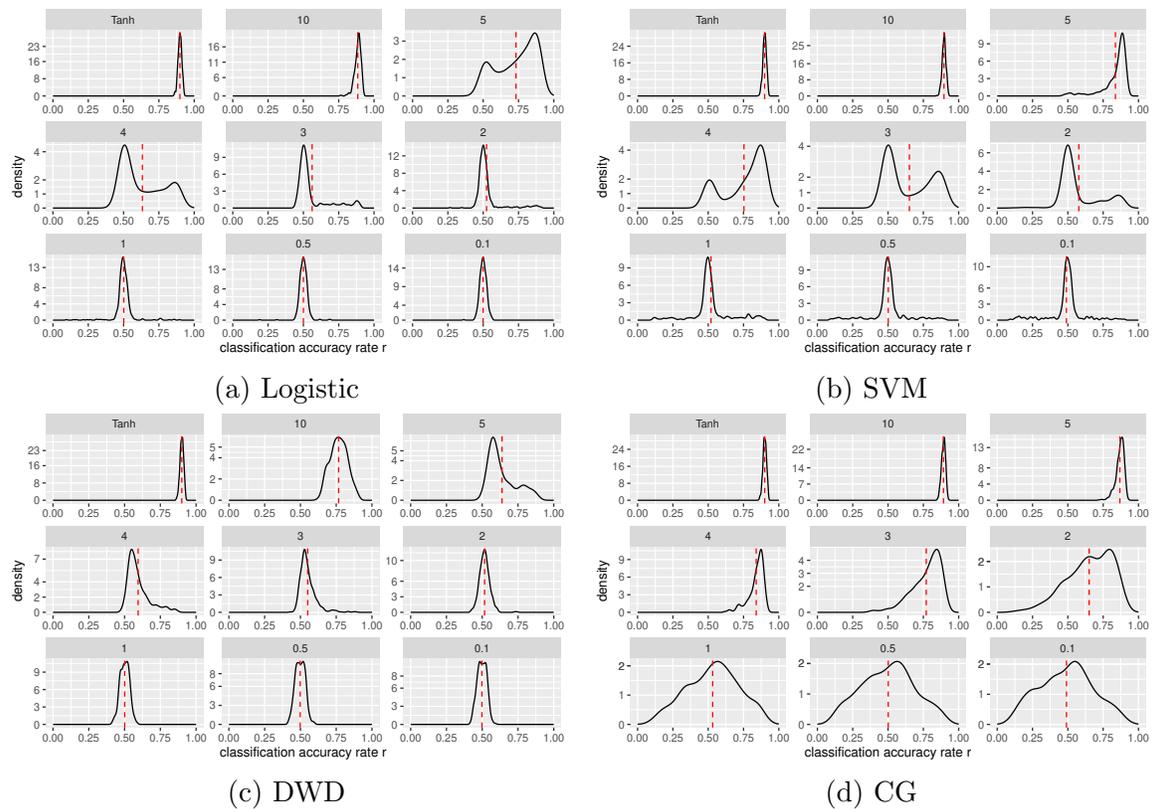
(b) SVM

(c) DWD

(d) CG

Figure A.2: The empirical distributions of classification accuracy for classifiers trained with $N = 3000$ clients and perturbed by Tanh transformation, where $\epsilon = 0.1, 0.5, 1, \ldots, 5, 10$, and the red dashed line represents the mean accuracy rate.

fiers gradually converge around 0.5. However, the distribution for the CG classifier remains more dispersed, indicating greater potential for improvement through model reversal.

A.2 Sample Size Balancing

In this section, we assess the performance of classifiers over varying values of the parameters N, N_0, N_1, n_0, n_1, B. We generate data based on the settings presented in Section 6. Specifically, we consider five distinct combinations of these parameter values, which are listed in Table A.2. Figures A.3 and A.4 display the misclassification rates of model-averaged classifiers using the cutoff value $r_0 = 0.6$, with and without model reversal, respectively, for the different parameter combinations.

Case	N	N_0	N_1	n_0	n_1	B
1	3000	500	2500	100	100	25
2	5500	500	5000	100	100	50
3	5500	500	5000	50	100	50
4	3000	500	2500	100	50	50
5	3000	500	2500	50	50	50

Table A.2: Five different combinations of parameters N, N_0, N_1, n_0, n_1, B

In Figure A.3, the differences arising from various parameter combinations on different types of classifiers manifest across distinct intervals of ε. Overall, Case 2 performs slightly better than Case 1, indicating that increasing the number of weak classifiers B and the sample size of the evaluation set N_1 can enhance the performance. The preference for a relatively smaller n_0 is evident as Case 3 outperforms Case 2, and Case 5 outperforms Case 4. Additionally, Case 2 is distinctly superior to Case 4, and Case 3 is markedly better than Case 5, indicating a preference for a larger n_1. Importantly, by comparing Case 1 and Case 4, we discern that, given N_0 and N_1, the allocation should favor a relatively smaller B and a larger n_1. This result is consistent with expectations, as a larger n_1 aids in more accurately estimating the performance of individual weak classifiers, thereby facilitating a more accurate model averaging.

Compared to Figure A.3, Figure A.4 incorporates model reversal prior to model averaging. It can be observed that the results of the various parameter combinations in Figure A.4 are consistent with those in Figure A.3. Notably, the introduction of model reversal has significantly enhanced the performance of both SVM and CG classifiers, particularly under more stringent privacy protection levels, characterized by smaller intervals of ε.

A.3 Sample Size per Weak Classifier

To demonstrate how the number of clients used to train weak classifiers influences the efficacy of MRMA in improving classifier performance, we consider four distinct settings, which are listed in Table A.3. Figure A.5 shows how the misclassification rates of the classifier CG vary with different parameter settings.
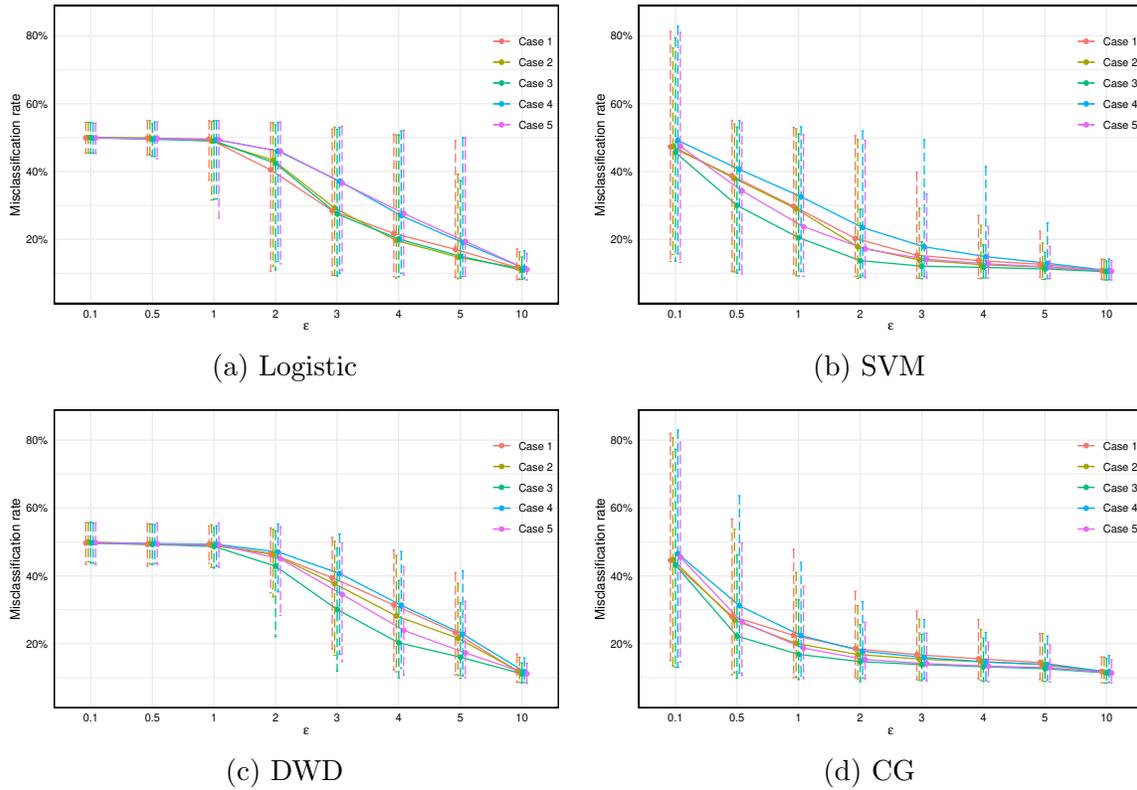
(a) Logistic

(b) SVM

(c) DWD

(d) CG

Figure A.3: The misclassification rates classifiers with model averaging under ε-LDP.

We can see from the results of cases 5, 6, and 7 in Figure A.5 that using more clients to train weak classifiers improves their performance when ε is large, implying low noise levels. However, when ε is small, which is our primary concern, increasing the sample size has little effect on the weak classifiers. In contrast, the classifiers based on MRMA consistently achieve significant improvements under different cases. Moreover, when we compare cases 7 and 8 in Figure A.5, where the training and evaluation data sets have different proportions, we find that allocating more data for evaluation, i.e., for MRMA, leads to better results than using it to enhance weak classifiers.

Case	N	N_0	N_1	n_0	n_1	B
5	3000	500	2500	50	50	50
6	5000	2500	2500	250	50	50
7	7500	5000	2500	500	50	50
8	7500	2500	5000	250	100	50

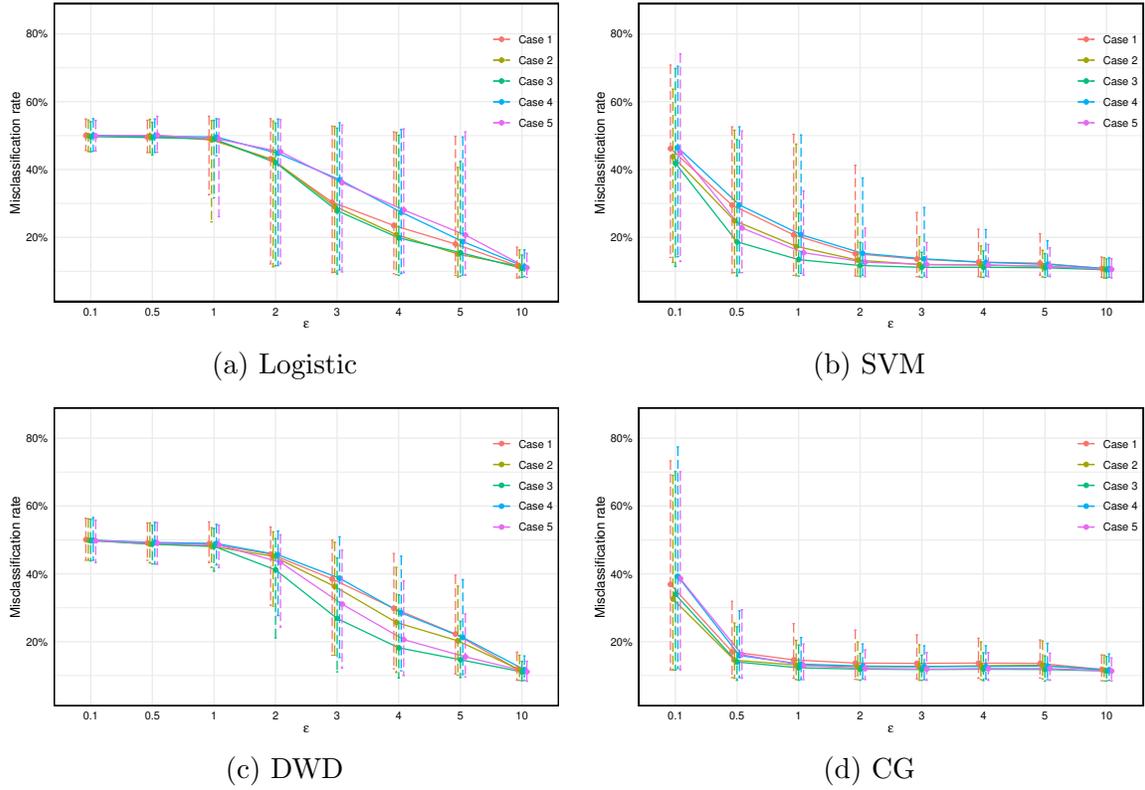Table A.3: Four different combinations of parameters N, N_0, N_1, n_0, n_1, B

Figure A.4: The misclassification rates of classifiers with model reversal and model averaging under ε-LDP.
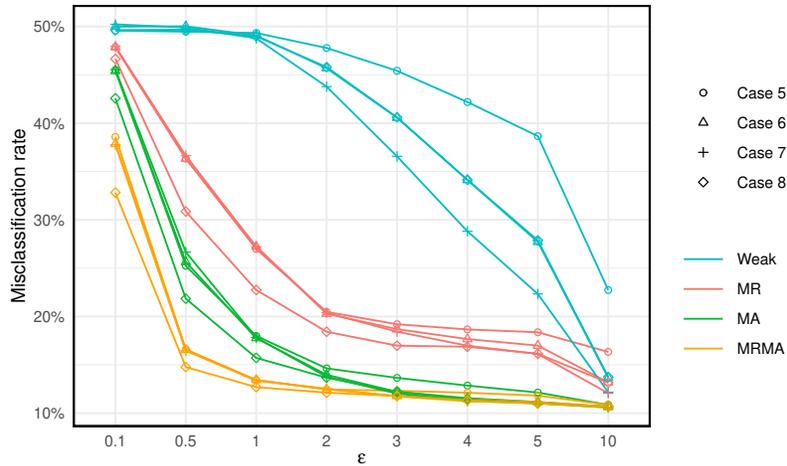

Figure A.5: The misclassification rates of classifier CG with a single server under ε-LDP.

A.4 Numerical Illustration of the Total Variation Distance

The excess risk bounds in Theorems 4-6 involve the term $d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)})$, which measures the discrepancy between the marginal distributions of Z and its privatized version $Z^{(\varepsilon_z)}$. Closed-form expressions for this total variation distance are generally unavailable, even under simple additive-noise mechanisms. To provide intuition regarding the behavior of this term, we present a simple numerical illustration.

We consider $Z \sim \mathrm{Unif}([-1,1]^d)$ and apply the Laplace mechanism (Dwork et al., 2006), which adds i.i.d. noise $\delta_k \sim \mathrm{Laplace}(0, \lambda_z)$ to each coordinate, where $\lambda_z = d(b-a)/\varepsilon_z$ with $[a,b] = [-1,1]$. The distribution of $Z^{(\varepsilon_z)} = Z + \delta$ typically has no simple closed form, so we approximate $d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)})$ via Monte Carlo: the empirical distributions are estimated on a common grid, and the total variation distance is computed as

$$d_{\mathrm{TV}}(P,Q) \approx \frac{1}{2} \sum_k |\hat{P}_k - \hat{Q}_k|.$$

Figure A.6 reports the estimated values for several dimensions $d \in \{1,2,5,10\}$ and a range of privacy levels ε_z. The results reveal two intuitive patterns. First, as $\varepsilon_z \to 0$ (i.e., stronger privacy and heavier noise), the discrepancy between P_Z and $P_Z^{(\varepsilon)}$ increases, and the total variation distance approaches 1. This corresponds to the case where the privatized distribution becomes almost disjoint from the original distribution. Second, for fixed ε_z, the distance increases with the dimension d, because the per-coordinate privacy budget effectively decreases as d grows. This is reflected in the scale parameter $\lambda_z = d(b-a)/\varepsilon_z$, which increases with d when the total privacy level ε_z is fixed, and thus induces larger Laplace noise in each coordinate.

A.5 Visualization of Functional Predictors

Figures A.7-A.8 provide visualizations of the functional predictors used in the physical activity and speech datasets, respectively, under different stages of processing: raw functional data, basis projection, and two types of transformations. In Figure A.7, we display the group mean MIMS curves for participants with healthy (HDL $\geq$ 50) and unhealthy (HDL $<$ 50) cholesterol levels, along with 10 randomly selected individual curves shown in gray. We limit the number of individual curves to reduce visual clutter, as the raw MIMS functions exhibit substantial variability across participants. Figure A.8 presents 200 randomly selected individual log-periodogram curves from the speech dataset along with the group means. Despite some smoothing and loss of fine-scale detail, these visualizations demonstrate that both temporal fluctuations and population-level differences are preserved throughout the functional representation pipeline.

Appendix B. Theoretical Analysis

Proof [of Proposition 2] We begin by noting that $g^{(\varepsilon)}(z_0) = \eta(z_0)\eta^{(\varepsilon)}(z_0) + (1 - \eta(z_0))(1 - \eta^{(\varepsilon)}(z_0))$, which yields the first expression in Equation (3).
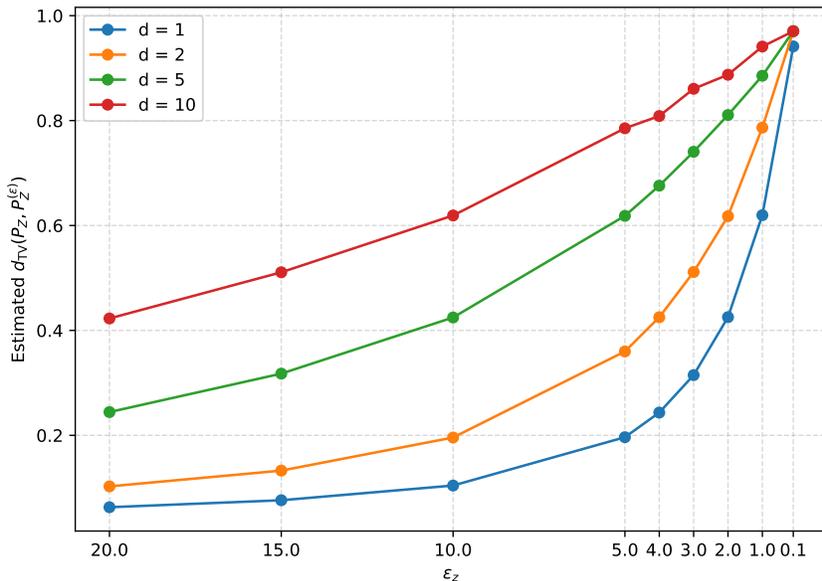
Figure A.6: Estimated total variation distance $d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)})$ under the Laplace mechanism for $Z \sim \mathrm{Unif}([-1, 1]^d)$, across dimensions $d \in \{1, 2, 5, 10\}$ and a range of privacy levels ε_z.

To compute $\eta^{(\varepsilon)}(z_0) - \frac{1}{2}$, we have that

$$\eta^{(\varepsilon)}(z_0) = \frac{P^{(\varepsilon)}(Z^{(\varepsilon_z)} = z_0, Y^{(\varepsilon_y)} = 1)}{P_Z^{(\varepsilon)}(Z^{(\varepsilon_z)} = z_0)},$$

where

$$P_Z^{(\varepsilon)}(Z^{(\varepsilon_z)} = z_0) = \mathbb{E}_{P_Z}\left(D^{(\varepsilon_z)}(\delta_z^{(\varepsilon_z)} = z_0 - z)\right),$$

$$P^{(\varepsilon)}(Z^{(\varepsilon_z)} = z_0, Y^{(\varepsilon_y)} = 1) = \mathbb{E}_{P_Z}\left(D^{(\varepsilon_z)}(\delta_z^{(\varepsilon_z)} = z_0 - z)\left(q^{(\varepsilon_y)}\eta(z) + (1 - q^{(\varepsilon_y)})(1 - \eta(z))\right)\right).$$

After some algebraic manipulation, we arrive at the expression given in Equation (4). ∎

Proof [of Theorem 3] Let $u = (x, y)$ represent the possible observation of a client, and $w = r'$ represent the possible report of a client. Let $f_{w|u}(w|u)$ be the conditional density of w given u. Then for any $w \in \{0, 1\}$, we have

$$\frac{f_{w|u}(w|u_1)}{f_{w|u}(w|u_2)} = \frac{P(r'|r_1)}{P(r'|r_2)} \leq e^{\varepsilon_v}.$$

Thus $\mathcal{M}$ satisfies ε_v-local differential privacy. Note that $r^{(b)}$ is the classification accuracy of the classifier $f^{(b)}$, then

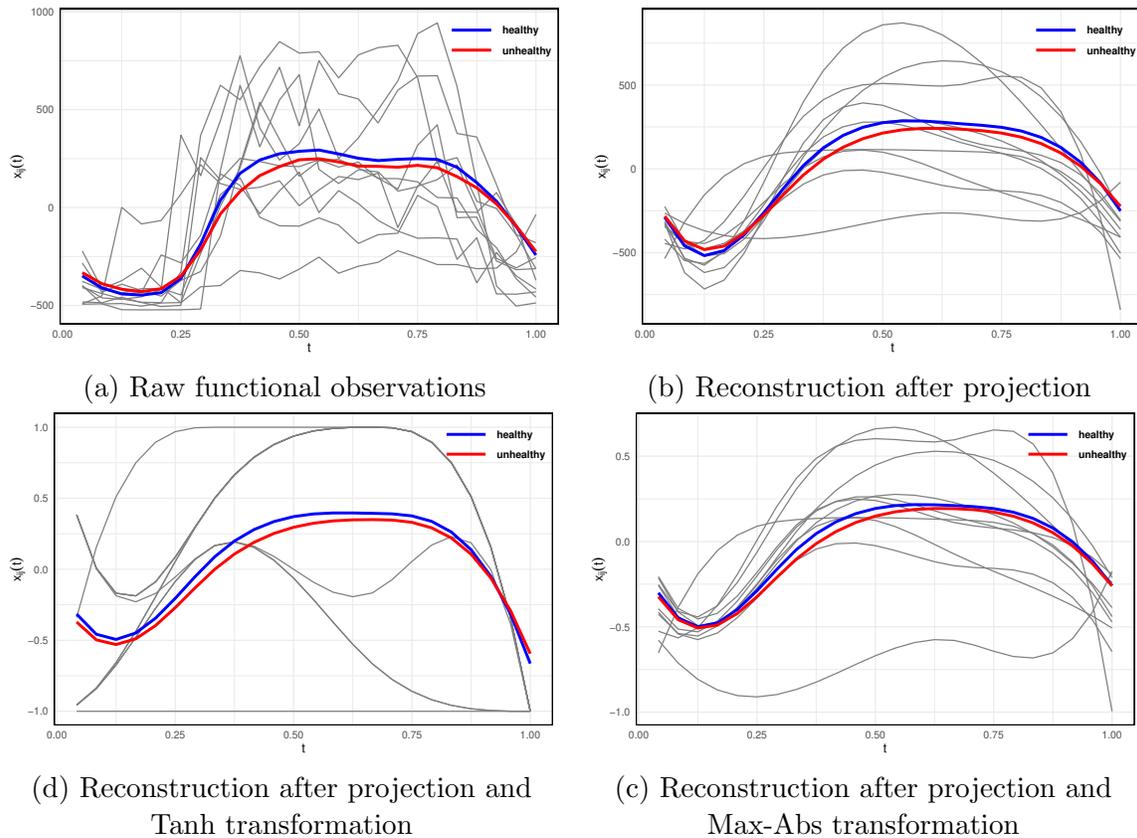$$\mathbb{E}(\hat{r}^{(b)}) = qr^{(b)} + (1 - q)(1 - r^{(b)}),$$

34

(a) Raw functional observations

(b) Reconstruction after projection

(d) Reconstruction after projection and
Tanh transformation

(c) Reconstruction after projection and
Max-Abs transformation

Figure A.7: Visualization of the mean MIMS curves for the "healthy" (HDL ≥ 50) and "unhealthy" (HDL < 50) groups in the physical activity dataset, with 10 randomly selected individual curves overlaid in gray. Each subplot shows the curves under different stages of processing the raw functional data.

where $q = e^{\varepsilon_v}/(1 + e^{\varepsilon_v})$. Thus we have $\mathbb{E}(\tilde{r}^{(b)}) = r^{(b)}$, and $\mathrm{Var}(\tilde{r}^{(b)}) = \mathrm{Var}(\hat{r}^{(b)})/(2q-1)^2 \leq ((e^{\varepsilon_v} + 1)/(e^{\varepsilon_v} - 1))^2/(4n_1^{(b)})$. This concludes the proof. ∎

Proof [of Theorem 4] By definition, we have

$$
\begin{aligned}
L_P(f^{(b)}, h^*) &= \mathbb{E}_P\{\ell(f^{(b)}) - \ell(h^*)\} \\
&= \{\mathbb{E}_P\ell(f^{(b)}) - \mathbb{E}_{P^{(\varepsilon)}}\ell(f^{(b)})\} + \mathbb{E}_{P^{(\varepsilon)}}\{\ell(f^{(b)}) - \ell(h^{*(\varepsilon)})\} + \{\mathbb{E}_{P^{(\varepsilon)}}\ell(h^{*(\varepsilon)}) - \mathbb{E}_P\ell(h^*)\} \\
&\triangleq I_1 + I_2 + I_3.
\end{aligned}
$$

To analyze $I_1 = \mathbb{E}_P\ell(f^{(b)}) - \mathbb{E}_{P^{(\varepsilon)}}\ell(f^{(b)})$, let $p(z)$ and $p^{(\varepsilon)}(z)$ denote the density functions of Z and $Z^{(\varepsilon)}$, respectively. Define

$$
p_0(z) = \min\{p(z), p^{(\varepsilon)}(z)\}, \quad p_1(z) = p(z) - p_0(z), \quad p_1^{(\varepsilon)}(z) = p^{(\varepsilon)}(z) - p_0(z).
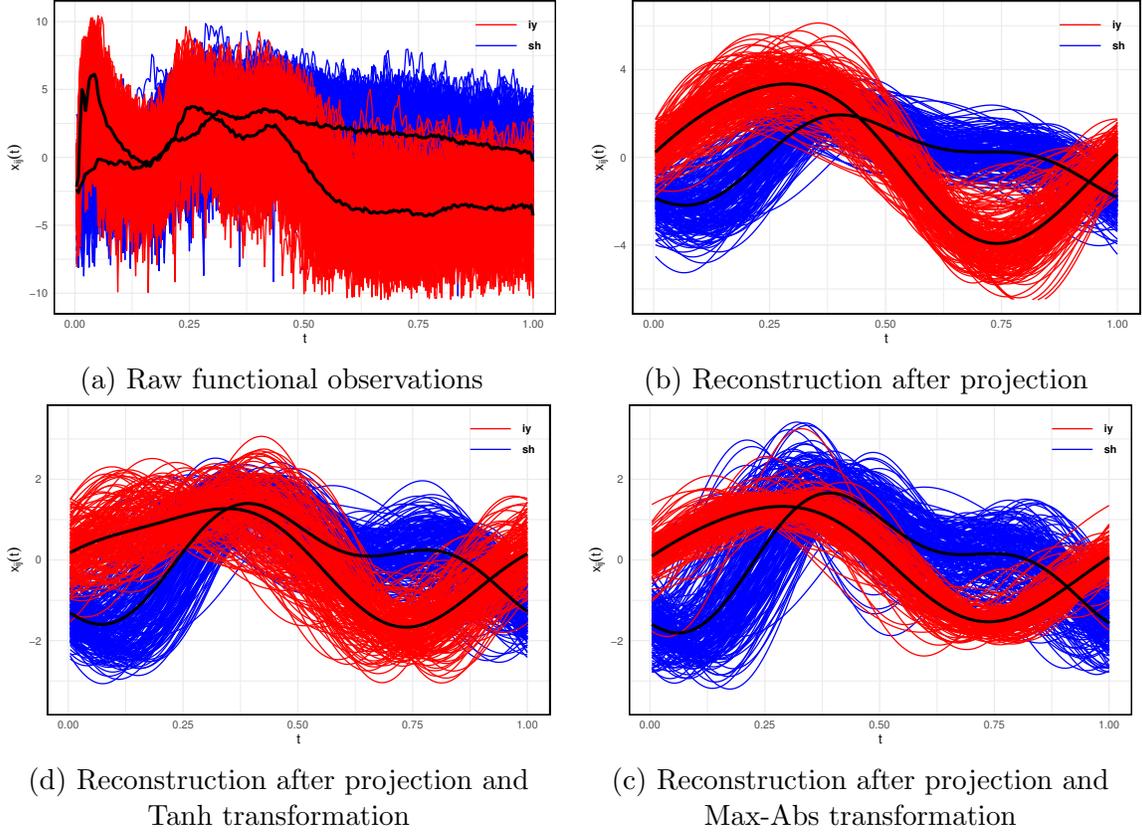$$

(a) Raw functional observations

(b) Reconstruction after projection

(d) Reconstruction after projection and Tanh transformation

(c) Reconstruction after projection and Max-Abs transformation

Figure A.8: Visualization of 200 randomly selected functional observations from groups "sh" and "iy" in the phonemes data set. The two black lines in each figure represent the mean functions of the two groups of curves.

Then,

$$
\begin{aligned}
I_1 &= \mathbb{E}_{P_Z}\left\{\eta(z)\ell(f^{(b)} \mid y=1) + (1-\eta(z))\ell(f^{(b)} \mid y=-1)\right\} \\
&\quad - \mathbb{E}_{P_Z^{(\varepsilon z)}}\left\{\eta^{(\varepsilon)}(z)\ell(f^{(b)} \mid y=1) + (1-\eta^{(\varepsilon)}(z))\ell(f^{(b)} \mid y=-1)\right\} \\
&= \int p_0(z)\left\{\eta(z) - \eta^{(\varepsilon)}(z)\right\}\left\{\ell(f^{(b)} \mid y=1) - \ell(f^{(b)} \mid y=-1)\right\}dz \\
&\quad + \int p_1(z)\ell(f^{(b)})dz - \int p_1^{(\varepsilon)}(z)\ell(f^{(b)})dz \\
&\le \mathbb{E}_{P_Z}\left\{|\eta(z) - \eta^{(\varepsilon)}(z)|\right\} + \int p_1(z)dz.
\end{aligned}
$$

Next, consider $I_3 = \mathbb{E}_{P^{(\varepsilon)}} \ell(h^{*(\varepsilon)}) - \mathbb{E}_P \ell(h^*)$. Note that

$$\begin{aligned}
\mathbb{E}_P \ell(h^*) &= \mathbb{E}_{P_Z} \left\{ I\{\eta(z) \geq 1/2\} P_{Y|Z}(y = -1) + I\{\eta(z) < 1/2\} P_{Y|Z}(y = 1) \right\} \\
&= \mathbb{E}_{P_Z} \left\{ I\{\eta(z) \geq 1/2\}(1 - \eta(z)) + I\{\eta(z) < 1/2\}\eta(z) \right\} \\
&= \mathbb{E}_{P_Z} \left\{ \frac{1}{2} - \left| \eta(z) - \frac{1}{2} \right| \right\},
\end{aligned}$$

and similarly,

$$\mathbb{E}_{P^{(\varepsilon)}} \ell(h^{*(\varepsilon)}) = \mathbb{E}_{P_Z^{(\varepsilon)}} \left\{ \frac{1}{2} - \left| \eta^{(\varepsilon)}(z) - \frac{1}{2} \right| \right\}.$$

Thus,

$$\begin{aligned}
I_3 &= \mathbb{E}_{P_Z} \left\{ |\eta(z) - \tfrac{1}{2}| \right\} - \mathbb{E}_{P_Z^{(\varepsilon)}} \left\{ |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}| \right\} \\
&= \int p_0(z) \left\{ |\eta(z) - \tfrac{1}{2}| - |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}| \right\} dz \\
&\quad + \int p_1(z) |\eta(z) - \tfrac{1}{2}| dz - \int p_1^{(\varepsilon)}(z) |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}| dz \\
&\leq \mathbb{E}_{P_Z} \left\{ |\eta(z) - \tfrac{1}{2}| - |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}| \right\} + \frac{1}{2} \int p_1(z) dz.
\end{aligned}$$

By the definition of total variation distance, we have

$$d_{\text{TV}}(P_Z, P_Z^{(\varepsilon)}) = 2 \int p_1(z) dz.$$

Combining the bounds for I_1 and I_3, we obtain the result stated in the theorem. ∎

Proof [of Theorem 5] Consider the noised dataset $\{(z_i^{(\varepsilon_z)}, y_i^{(\varepsilon_y)})\}_{i=1}^{n_0}$ and the classifier $f^{(b)}$ trained on it. We define a pseudo-noised dataset $\{(z_i^{(\varepsilon_z)}, \tilde{y}_i^{(\varepsilon_y)})\}_{i=1}^{n_0}$ where $\tilde{y}_i^{(\varepsilon_y)} = -y_i^{(\varepsilon_y)}$, and correspondingly define $\tilde{f}^{(b)} = -f^{(b)}$. Then $\tilde{f}^{(b)}$ can be interpreted as the classifier trained on this pseudo-noised dataset.

Let $\tilde{P}^{(\varepsilon)}$ denote the joint distribution of $(Z^{(\varepsilon_z)}, \tilde{Y}^{(\varepsilon_y)})$. Then the marginal distribution of $Z^{(\varepsilon_z)}$ under $\tilde{P}^{(\varepsilon)}$ remains the same as that under $P^{(\varepsilon)}$, i.e., $\tilde{P}_Z^{(\varepsilon)} = P_Z^{(\varepsilon)}$. The conditional distribution satisfies $\tilde{P}_{Y|Z}^{(\varepsilon)} = 1 - P_{Y|Z}^{(\varepsilon)}$, and the corresponding Bayes classifier becomes $\tilde{h}^{*(\varepsilon)}(z) = \text{sign}\{\tilde{\eta}^{(\varepsilon)}(z) \geq 1/2\} = -h^{*(\varepsilon)}(z)$, where $\tilde{\eta}^{(\varepsilon)}(z) = 1 - \eta^{(\varepsilon)}(z)$.

Applying Theorem 4, we obtain

$$\begin{aligned}
L_P(-f^{(b)}, h^*) = L_P(\tilde{f}^{(b)}, h^*) &\leq L_{\tilde{P}^{(\varepsilon)}}(\tilde{f}^{(b)}, \tilde{h}^{*(\varepsilon)}) + \frac{3}{4} d_{\text{TV}}(P_Z, \tilde{P}_Z^{(\varepsilon)}) + \mathbb{E}_{P_Z} \left(|\eta(z) - \tilde{\eta}^{(\varepsilon)}(z)| \right) \\
&\quad + \mathbb{E}_{P_Z} \left(|\eta(z) - \tfrac{1}{2}| - |\tilde{\eta}^{(\varepsilon)}(z) - \tfrac{1}{2}| \right) \\
&= L_{P^{(\varepsilon)}}(f^{(b)}, h^{*(\varepsilon)}) + \frac{3}{4} d_{\text{TV}}(P_Z, P_Z^{(\varepsilon)}) + \mathbb{E}_{P_Z} \left(|\eta(z) - \tilde{\eta}^{(\varepsilon)}(z)| \right) \\
&\quad + \mathbb{E}_{P_Z} \left(|\eta(z) - \tfrac{1}{2}| - |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}| \right).
\end{aligned}$$

Furthermore, we note that

$$\min\left\{\mathbb{E}_{P_Z}\left(|\eta(z) - \tilde{\eta}^{(\varepsilon)}(z)|\right), \mathbb{E}_{P_Z}\left(|\eta(z) - \eta^{(\varepsilon)}(z)|\right)\right\} = \mathbb{E}_{P_Z}\left(|\eta(z) - \tfrac{1}{2}| - |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}|\right).$$

This implies

$$\min\left\{L_P(f^{(b)}, h^*), L_P(-f^{(b)}, h^*)\right\} \leq L_{P^{(\varepsilon)}}(f^{(b)}, h^{*(\varepsilon)}) + \frac{3}{4}d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)})$$
$$+ 2\mathbb{E}_{P_Z}\left(|\eta(z) - \tfrac{1}{2}| - |\eta^{(\varepsilon)}(z) - \tfrac{1}{2}|\right).$$

Let $r^{(b)}$ denote the true classification accuracy of the Bayes classifier $h^{*(\varepsilon)}$ under the distribution P, and let $\tilde{r}^{(b)}$ denote the estimated classification accuracy of $f^{(b)}$ based on n_1 evaluation samples. Define the selected classifier $f^{*(b)} \in \{f^{(b)}, -f^{(b)}\}$ as the one with higher estimated accuracy: specifically, $f^{*(b)} = f^{(b)}$ if $\hat{c}_{n_1} \geq 1/2$, and $f^{*(b)} = -f^{(b)}$ otherwise. Then, the probability that model reversal correctly selects the better classifier satisfies

$$\mathbb{P}\left[L_P(f^{*(b)}, h^*) = \min\left\{L_P(f^{(b)}, h^*), L_P(-f^{(b)}, h^*)\right\}\right] = \mathbb{P}\left\{\left(\tilde{r}^{(b)} - \tfrac{1}{2}\right)\left(r^{(b)} - \tfrac{1}{2}\right) \geq 0\right\}$$
$$\approx \Phi\left\{\sqrt{n_1}\frac{|r^{(b)} - 1/2|}{\sqrt{r^{(b)}(1 - r^{(b)})}}\right\},$$

where Φ is the cumulative distribution function of the standard normal distribution. ∎

Proof [of Theorem 6] Let $f^{\dagger}(z) = \sum_{b=1}^{B} w_b f^{*(b)}(z)$ be the model-averaged classifier, where each $f^{*(b)} \in \{f^{(b)}, -f^{(b)}\}$ is selected via model reversal to maximize estimated accuracy, and the weights $\{w_b\}_{b=1}^{B}$ sum to 1. Let $\tilde{r}_z^{*(b)} = \mathbb{P}\{\mathrm{sign}[f^{*(b)}(z)] = y\}$ denote the classification accuracy of the b-th classifier at point z. Let $F_z^{(\varepsilon)}$ be the distribution of $\tilde{r}_z^{*(b)}$ under random training and evaluation sets, with support $[r_{z,0}, r_{z,1}]$. Define the aggregated mapping

$$\eta^{\dagger(\varepsilon)}(z) := \sum_{b=1}^{B} w_b \eta^{*(b)}(z),$$

where $\eta^{*(b)}(z) := \mathbb{P}\{Y = 1 \mid Z = z, f^{*(b)}\}$ is the latent regression function implied by classifier $f^{*(b)}$. Under standard assumptions, the classification accuracy of $f^{*(b)}(z)$ is

$$\tilde{r}_z^{*(b)} = \frac{1}{2} + \left|\eta^{*(b)}(z) - \frac{1}{2}\right|.$$

As $B \to \infty$, the empirical average of classifiers concentrates around those with higher accuracy, due to the weight truncation scheme that sets $w_b = 0$ if $\tilde{r}_z^{*(b)} < r_0$. Since the condition $B_0/B \to 0$ ensures that only a vanishing fraction of classifiers fall below the cutoff r_0, the weighted average function $\eta^{\dagger(\varepsilon)}(z)$ satisfies

$$\left|\eta^{\dagger(\varepsilon)}(z) - \frac{1}{2}\right| \xrightarrow{\mathbb{P}} r_{z,1} - \frac{1}{2}.$$

Now, recall from Theorem 5 that for each classifier $f^{*(b)}$, the excess risk is bounded by

$$L_P(f^{*(b)}, h^*) \leq L_{P^{(\varepsilon)}}(f^{*(b)}, h^{*(\varepsilon)}) + \frac{3}{4} d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)}) + 2 \mathbb{E}_{P_Z} \left(\left| \eta(z) - \tfrac{1}{2} \right| - \left| \eta^{*(b)}(z) - \tfrac{1}{2} \right| \right).$$

Taking a convex combination over b, and using the fact that the minimum excess risk over $\{f^{*(b)}\}$ is preserved in expectation under model averaging, we obtain

$$L_P(f^\dagger, h^*) \leq L_{P^{(\varepsilon)}}(f^\dagger, h^{*(\varepsilon)}) + \frac{3}{4} d_{\mathrm{TV}}(P_Z, P_Z^{(\varepsilon)})$$
$$+ 2 \mathbb{E}_{P_Z} \left(\left| \eta(z) - \tfrac{1}{2} \right| - \left| \eta^{\dagger,(\varepsilon)}(z) - \tfrac{1}{2} \right| \right).$$

As discussed, $\left| \eta^{\dagger,(\varepsilon)}(z) - \tfrac{1}{2} \right| \to r_{z,1} - \tfrac{1}{2}$ in probability, which establishes the theorem. ∎

Proof [of Theorem 7] Let $u = (x(t), y)$ represent the possible observation of a client, and $v = (z^{(\varepsilon_z)}, y^{(\varepsilon_y)})$ represent the possible report of a client. Let $f_{v|u}(v|u)$ be the conditional density of v given u. Then for any possible output v of $\mathcal{M}_f(u)$, by the sequential composition theorem (McSherry and Talwar, 2007), we have

$$\frac{f_{v|u}(v|u_1)}{f_{v|u}(v|u_2)} = \frac{P(y^{(\varepsilon_y)}|y_1)}{P(y^{(\varepsilon_y)}|y_2)} \prod_{k=1}^{d} \frac{f(z_k^{(\varepsilon_z)} - z_{1,k}^*)}{f(z_k^{(\varepsilon_z)} - z_{2,k}^*)}$$

$$\leq \max(1, \frac{q}{1-q}, \frac{1-q}{q}) \prod_{k=1}^{d} \exp\left(\frac{\varepsilon_z}{d\Delta} (|z_k^{(\varepsilon_z)} - z_{1,k}^*| - |z_k^{(\varepsilon_z)} - z_{2,k}^*|) \right)$$

$$\leq e^{\varepsilon_y} \prod_{k=1}^{d} \exp\left(\frac{\varepsilon_z}{d} \right) = e^{\varepsilon_z + \varepsilon_y} = e^{\varepsilon},$$

where $q = e^{\varepsilon_y}/(1 + e^{\varepsilon_y})$. Thus $\mathcal{M}_f$ satisfies ε-local differential privacy. ∎

References

Autcha Araveeporn. The higher-order of adaptive lasso and elastic net methods for classification on high dimensional data. *Mathematics*, 9(10):1091, 2021.

Héber H Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. Random sampling plus fake data: Multidimensional frequency estimates with local differential privacy. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 47–57, 2021.

Héber H Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. *Digital Communications and Networks*, 2022.

Yang Bai, Caihong Qin, and Huichen Zhu. Functional two-sample test based on projection. *Statistica Sinica Preprint*, 2023. doi: 10.5705/ss.202023.0272. URL http://www.stat.sinica.edu.tw/statistica/.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Thomas Berrett and Cristina Butucea. Classification under local differential privacy. *arXiv preprint arXiv:1912.04629*, 2019.

Gilles Blanchard, Olivier Bousquet, and Pascal Massart. Statistical performance of support vector machines. *The Annals of Statistics*, 36(2):489–531, 2008.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128, 2006.

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

Xiongtao Dai, Hans-Georg Müller, and Fang Yao. Optimal bayes classifiers for functional data and density ratios. *Biometrika*, 104(3):545–560, 2017.

Apple Differential Privacy Team. Learning with privacy at scale, 2017. URL https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf. Accessed: 20 Sep 2023.

Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *Advances in Neural Information Processing Systems*, 30, 2017.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521): 182–201, 2018.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

Tawfik Elmetwally. Employee dataset. https://www.kaggle.com/datasets/tawfikelmetwally/employee-dataset/data, 2023.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618*, 2020.

Randall L Eubank. *Nonparametric regression and spline smoothing*. CRC press, 1999.

John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

Joseph Geumlek and Kamalika Chaudhuri. Profile-based privacy for locally private computations. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 537–541. IEEE, 2019.

Yanwei Gong, Xiaolin Chang, Jelena Mišić, Vojislav B Mišić, Jianhua Wang, and Haoran Zhu. Practical solutions in fully homomorphic encryption: a survey analyzing existing acceleration methods. *Cybersecurity*, 7(1):5, 2024.

Peter Hall, Donald S Poskitt, and Brett Presnell. A functional data—analytic approach to signal discrimination. *Technometrics*, 43(1):1–9, 2001.

Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *The Journal of Machine Learning Research*, 14(1):703–727, 2013.

Lajos Horváth and Piotr Kokoszka. *Inference for functional data with applications*, volume 200. Springer Science & Business Media, 2012.

Lajos Horváth and Gregory Rice. An introduction to functional data analysis and a principal component approach for testing the equality of mean curves. *Revista Matemática Complutense*, 28(3):505–548, 2015.

Asmaul Hosna, Ethel Merry, Jigmey Gyalmo, Zulfikar Alom, Zeyar Aung, and Mohammad Abdul Azim. Transfer learning: a friendly introduction. *Journal of Big Data*, 9(1): 102, 2022.

Yiyang Huang and Clément L Canonne. Lemmas of differential privacy. *arXiv preprint arXiv:2211.11189*, 2022.

Dihong Jiang, Sun Sun, and Yaoliang Yu. Functional renyi differential privacy for generative modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Hyunouk Ko, Namjoon Suh, and Xiaoming Huo. On excess risk convergence rates of neural network classifiers, 2023. URL https://arxiv.org/abs/2309.15075.

Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

David Kraus and Marco Stefanucci. Classification of functional fragments by regularized linear classifiers with domain selection. *Biometrika*, 106(1):161–180, 2019.

Roshan Kumari and Saurabh Kr Srivastava. Machine learning: A review on binary classification. *International Journal of Computer Applications*, 160(7), 2017.

Malcolm R Leadbetter, Georg Lindgren, and Holger Rootzén. *Extremes and related properties of random sequences and processes*. Springer Science & Business Media, 2012.

Xiaoyan Leng and Hans-Georg Müller. Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22(1):68–76, 2006.

Mushan Li and Yanyuan Ma. An update on measurement error modeling. *Annual Review of Statistics and Its Application*, 11, 2024.

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

Haotian Lin and Matthew Reimherr. Transfer learning for functional linear regression with structural interpretability. *arXiv preprint arXiv:2206.04277*, 2022.

Haotian Lin and Matthew Reimherr. Differentially private functional summaries via the independent component laplace process. *arXiv preprint arXiv:2309.00125*, 2023.

Hai Liu, Changgen Peng, Youliang Tian, Shigong Long, Feng Tian, and Zhenqiang Wu. Gdp vs. ldp: A survey from the perspective of information-theoretic channel. *Entropy*, 24(3):430, 2022.

Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. Fedsel: Federated sgd under local differential privacy with top-k dimension selection. In *Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part I 25*, pages 485–501. Springer, 2020.

Yuheng Ma and Hanfang Yang. Optimal locally private nonparametric classification with public data. *Journal of Machine Learning Research*, 25(167):1–62, 2024.

Chiara Marcolla, Victor Sucasas, Marc Manzano, Riccardo Bassoli, Frank HP Fitzek, and Najwa Aaraj. Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10):1572–1609, 2022.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

Casey Meehan, Amrita Roy Chowdhury, Kamalika Chaudhuri, and Somesh Jha. Privacy implications of shuffling. In *International Conference on Learning Representations*, 2022.

Ardalan Mirshani, Matthew Reimherr, and Aleksandra Slavković. Formal privacy for functional data with gaussian perturbations. In *International Conference on Machine Learning*, pages 4595–4604. PMLR, 2019.

Thông T Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, and Junbum Shin. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053*, 2016.

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

Zhiniang Peng. Danger of using fully homomorphic encryption: A look at microsoft seal. *arXiv preprint arXiv:1906.07127*, 2019.

Gina-Maria Pomann, Ana-Maria Staicu, and Sujit Ghosh. A two sample distribution-free test for functional data with application to a diffusion tensor imaging study of multiple sclerosis. *Journal of the Royal Statistical Society. Series C, Applied statistics*, 65(3):395, 2016.

Nandita Pore. Healthcare diabetes dataset. `https://www.kaggle.com/datasets/nanditapore/healthcare-diabetes/data`, 2023.

Caihong Qin, Jinhan Xie, Ting Li, and Yang Bai. An adaptive transfer learning framework for functional classification. *Journal of the American Statistical Association*, 0(ja):1–22, 2024. doi: 10.1080/01621459.2024.2403788.

Likun Qin, Nan Wang, and Tianshuo Qiu. A survey of local differential privacy and its variants. *arXiv preprint arXiv:2309.00861*, 2023.

James O Ramsay and Bernhard W Silverman. *Functional data analysis.* Springer, New York, 2005.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

Yashar Salami, Vahid Khajevand, and Esmaeil Zeinali. Cryptographic algorithms: A review of the literature, weaknesses and open challenges. *J. Comput. Robot*, 16(2):46–56, 2023.

Peijun Sang, Adam B Kashlak, and Linglong Kong. A reproducing kernel hilbert space framework for functional classification. *Journal of Computational and Graphical Statistics*, pages 1–9, 2022.

Kl Saravanan and S Sasithra. Review on classification based on artificial neural networks. *Int J Ambient Syst Appl*, 2(4):11–18, 2014.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pages 127–140, 2015.

Lichao Sun, Jianwei Qian, and Xun Chen. Ldp-fl: Practical private aggregation in federated learning with local differential privacy. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1571–1578. International Joint Conferences on Artificial Intelligence Organization, 2021.

Ye Tian and Yang Feng. Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, 0(0):1–14, 2022.

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

Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, pages 61–66, 2020.

Melanie Warziski Turk, Patricia K Tuite, and Lora E Burke. Cardiac health: primary prevention of heart disease in women. *The Nursing clinics of North America*, 44(3):315, 2009.

Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 638–649. IEEE, 2019a.

Teng Wang, Xinyu Yang, Xuebin Ren, Wei Yu, and Shusen Yang. Locally private high-dimensional crowdsourced data release based on copula functions. *IEEE Transactions on Services Computing*, 15(2):778–792, 2019b.

Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017.

Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

Hanshen Xiao, Jun Wan, and Srinivas Devadas. Geometry of sensitivity: twice sampling and hybrid clipping in differential privacy with optimal gaussian noise and application to deep learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2636–2650, 2023.

Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam. Local differential privacy and its applications: A comprehensive survey. *arXiv preprint arXiv:2008.03686*, 2020.

Qingqing Ye and Haibo Hu. Local differential privacy: Tools, challenges, and opportunities. In *International conference on web information systems engineering*, pages 13–23. Springer, 2020.

Emre Yilmaz, Mohammad Al-Rubaie, and J Morris Chang. Naive bayes classification under local differential privacy. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 709–718. IEEE, 2020.