


## Article

# An Information Theoretic Approach to Privacy-Preserving Interpretable and Transferable Learning

Mohit Kumar <sup>1,2,\*</sup> , Bernhard A. Moser <sup>2,3</sup>, Lukas Fischer <sup>2</sup> and Bernhard Freudenthaler <sup>2</sup><sup>1</sup> Faculty of Computer Science and Electrical Engineering, University of Rostock, 18051 Rostock, Germany<sup>2</sup> Software Competence Center Hagenberg GmbH, A-4232 Hagenberg, Austria<sup>3</sup> Institute of Signal Processing, Johannes Kepler University Linz, 4040 Linz, Austria

\* Correspondence: mohit.kumar@scch.at

**Abstract:** In order to develop machine learning and deep learning models that take into account the guidelines and principles of trustworthy AI, a novel information theoretic approach is introduced in this article. A unified approach to privacy-preserving interpretable and transferable learning is considered for studying and optimizing the trade-offs between the privacy, interpretability, and transferability aspects of trustworthy AI. A variational membership-mapping Bayesian model is used for the analytical approximation of the defined information theoretic measures for privacy leakage, interpretability, and transferability. The approach consists of approximating the information theoretic measures by maximizing a lower-bound using variational optimization. The approach is demonstrated through numerous experiments on benchmark datasets and a real-world biomedical application concerned with the detection of mental stress in individuals using heart rate variability analysis.

**Keywords:** privacy; interpretability; transferability; information theory; membership mappings; variational optimization; machine and deep learning



**Citation:** Kumar, M.; Moser, B.A.; Fischer, L.; Freudenthaler, B. An Information Theoretic Approach to Privacy-Preserving Interpretable and Transferable Learning. *Algorithms* **2023**, *16*, 450. <https://doi.org/10.3390/a16090450>

Academic Editors: Frank Werner and Yue Duan

Received: 7 June 2023

Revised: 30 August 2023

Accepted: 8 September 2023

Published: 20 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Trust in the development, deployment, and use of AI is essential in order to fully utilize the potential of AI to contribute to human well-being and society. Recent advances in machine and deep learning have rejuvenated the field of AI with an enthusiasm that AI could become an integral part of human life. However, a rapid proliferation of AI will give rise to several ethical, legal, and social issues.

### 1.1. Trustworthy AI

In response to the ethical, legal, and social challenges that accompany AI, guidelines and ethical principles have been established [1–4] in order to evaluate the responsible development of AI systems that are good for humanity and the environment. These guidelines have introduced the concept of *trustworthy AI* (TAI), and the term TAI has quickly gained attention in research and practice. TAI is based on the idea that trust in AI will allow AI to realize its full potential in contributing to societies, economies, and sustainable development. As “trust” is a complex phenomenon being studied in diverse disciplines (i.e., psychology, sociology, economics, management, computer science, and information systems), the definition and realization of TAI remains challenging. While forming trust in technology, users express expectations about the technology’s *functionality*, *helpfulness* and *reliability* [5]. The authors in [6] state that “AI is perceived as trustworthy by its users (e.g., consumers, organizations, and society) when it is developed, deployed, and used in ways that not only ensure its compliance with all relevant laws and its robustness but especially its adherence to general ethical principles”.

Academicians, industries, and policymakers have developed in recent times for TAI several frameworks and guidelines including “Asilomar AI Principles” [7], “Montreal

Declaration of Responsible AI” [8], “UK AI Code” [9], “AI4People” [4], “Ethics Guidelines for Trustworthy AI” [1], “OECD Principles on AI” [10], “Governance Principles for the New Generation Artificial Intelligence” [11], and “Guidance for Regulation of Artificial Intelligence Applications” [12]. However, it was argued in [13] that AI ethics lack a reinforcement mechanism, and economic incentives could easily override commitment to ethical principles and values.

The five principles of ethical AI [4] (i.e., *beneficence*, *non-maleficence*, *autonomy*, *justice*, and *explicability*) have been adopted for TAI [6]. Beneficence refers to promoting the well-being of humans, preserving dignity, and sustaining the planet. Non-maleficence refers to avoiding bringing harm to people and is especially concerned with the protection of people’s privacy and security. Autonomy refers to the promotion of human autonomy, agency, and oversight including the restriction of AI systems’ autonomy, where necessary. Justice refers to using AI for correcting past wrongs, ensuring shared benefits through AI, and preventing the creation of new harms and inequities by AI. Explicability comprises an epistemological sense and an ethical sense. Explicability refers in the epistemological sense to the explainable AI developed by creating interpretable AI models with high levels of performance and accuracy. In the ethical sense, explicability refers to accountable AI.

### 1.2. Motivation and Novelty

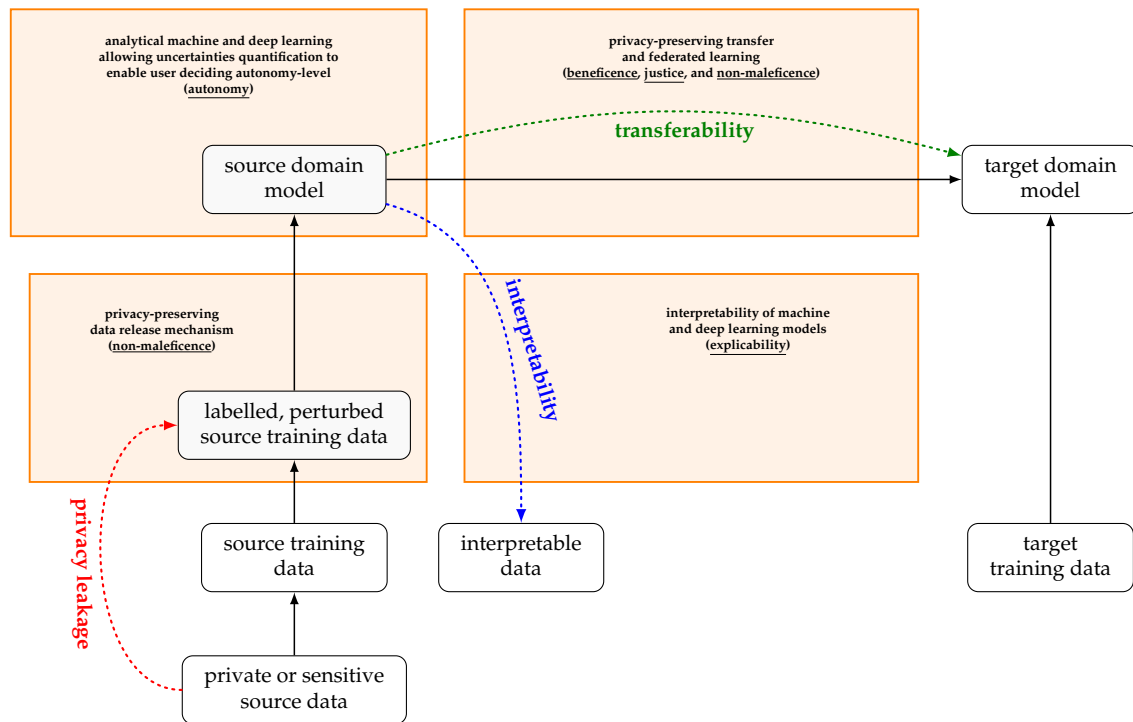
The core issues related to machine and deep learning that need to be addressed in order to fulfill the five principles of trustworthy AI are listed in Table 1.

**Table 1.** Core issues with TAI principles and solution approaches.

| TAI Principle   | Issue  | Solution Approach                                |
|-----------------|--|--|
| Beneficence     | I1: non-availability of large high-quality training data   | transfer learning                                |
|                 | I2: models (intellectual properties) are not widely available  | federated learning                               |
| Non-maleficence | I3: leakage of private information embedded in training data   | privacy-preserving data release mechanism        |
|                 | I4: leakage of private information embedded in model parameters and model outputs  | privacy-preserving machine and deep learning     |
| Autonomy        | I5: user’s inability to quantify model uncertainties lead to indecisiveness regarding the level of autonomy given to AI system | analytical quantification of model uncertainties |
| Justice         | I6: bias of training data toward certain groups of people leads to discrimination  | federated learning                               |
| Explicability   | I7: user’s inability to understand model functionality leads to mistrust and obstruction in establishing accountability        | interpretable machine and deep learning models   |

Solution approaches to address issues concerning TAI have been identified in Table 1; however, a unified solution approach addressing all major issues does not exist. Despite the importance of the outlined TAI principles, their major limitation, as identified in [6], concerns the fact that the principles are highly general and provide little to no guidance for how they can be transferred into practice. To address this limitation, a data-driven research framework for TAI was outlined in [6]. However, to the best knowledge of the authors, no previous study presented a unified information theoretic approach to study the privacy, interpretability, and transferability aspects of trustworthy AI in a rigorous analytical manner.

This motivated us in this study to develop a novel information theoretic approach for addressing the privacy, interpretability, and transferability aspects of trustworthy AI in a rigorous analytical manner. This study introduces a unified information theoretic approach to “privacy-preserving interpretable and transferable learning”, as represented in Figure 1, for addressing trustworthy AI issues, which is the novelty of this study.



**Figure 1.** An information theoretic unified approach to “privacy-preserving interpretable and transferable learning” for studying the privacy–interpretability–transferability trade-offs while addressing beneficence, non-maleficence, autonomy, justice, and explicability principles of TAI.

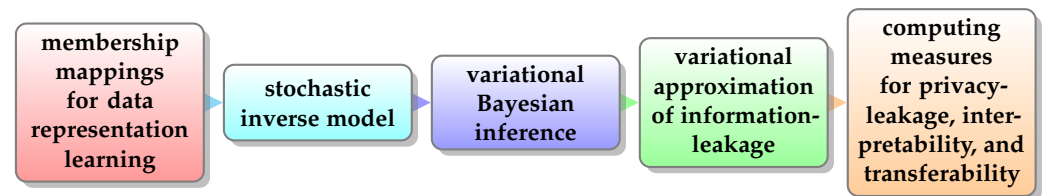
### 1.3. Goal and Aims

Our goal is to develop a novel approach to trustworthy AI based on the hypothesis that information theory enables taking into account the privacy, interpretability, and transferability aspects of trustworthy AI principles during the development of machine learning and deep learning models by providing a way to study and optimize the inherent trade-offs. The aims focused on the development of our approach are the following:

- Aim 1:** To develop an information theoretic approach to privacy that enables the quantification of privacy leakage in terms of the mutual information between sensitive private data and the data released to the public without the availability of prior knowledge about data statistics (such as joint distributions of public and private variables).
- Aim 2:** To develop an information theoretic criterion for evaluating the interpretability of a machine learning model in terms of the mutual information between non-interpretable model outputs/activations and corresponding interpretable parameters.
- Aim 3:** To develop an information theoretic criterion for evaluating the transferability (of a machine learning model from source to target domain) in terms of the mutual information between source domain model outputs/activations and target domain model outputs/activations.
- Aim 4:** To develop analytical approaches to machine and deep learning allowing for the quantification of model uncertainties.
- Aim 5:** To develop a unified approach to “privacy-preserving interpretable and transferable learning” for an analytical optimization of privacy–interpretability–transferability trade-offs.

### 1.4. Methodology

Figure 2 outlines the methodological workflow. For an information theoretic evaluation of the privacy leakage, interpretability, and transferability, we provide a novel method that consists of following three steps:



**Figure 2.** The proposed methodology to evaluate privacy leakage, interpretability, and transferability in terms of the information leakages.

#### 1.4.1. Defining Measures in Terms of the Information Leakages

The privacy, interpretability, and transferability measures are defined in terms of the information leakages:

- Privacy leakage is measured as the amount of information about private/sensitive variables leaked by the shared variables;
- Interpretability is measured as the amount of information about interpretable parameters leaked by the model;
- Transferability is measured as the amount of information about the source domain model output leaked by the target domain model output.

#### 1.4.2. Variational Membership Mapping Bayesian Models

In order to derive analytical expressions for the defined privacy leakage, interpretability, and transferability measures, the stochastic inverse models (governing the relationships amongst variables) will be required. In this study, the variational membership mappings are leveraged to build the required stochastic inverse models. Membership mappings [14,15] have been introduced as an alternative to deep neural networks in order to address the issues such as determining the optimal model structure, smaller training dataset, and iterative time-consuming nature of numerical learning algorithms [16–22]. A membership mapping represents data through a fuzzy set (characterized by a membership function such that the dimension of the membership function increases with an increasing data size). A remarkable feature of membership mappings is that these allow an analytical approach to the variational learning of a membership-mappings-based data representation model. Our idea is to employ membership mappings for defining a stochastic inverse model, which is then inferred using variational Bayesian methodology.

#### 1.4.3. Variational Approximation of Information Theoretic Measures

The variational membership-mapping Bayesian models are used to determine the lower bounds on the defined information theoretic measures for privacy leakage, interpretability, and transferability. The lower bounds are then maximized using variational optimization methodology to derive analytically the expressions that approximate the privacy leakage, interpretability, and transferability measures. The analytically derived expressions form the basis of an algorithm that practically computes the measures using available data samples, where expectations over unknown distributions are approximated by sample averages.

### 1.5. Contributions

The main contributions of this study are following:

### 1.5.1. A Unified Approach to Study the Privacy, Interpretability, and Transferability Aspects of Trustworthy AI

The study introduces a novel information theoretic unified approach (as represented in Figure 1) to address the

1. Issues **I1** and **I2** of beneficence principle by means of transfer and federated learning;
2. Issues **I3** and **I4** of non-maleficence principle by means of privacy-preserving data release mechanisms;
3. Issue **I5** of autonomy principle by means of analytical machine and deep learning algorithms that enable the user to quantify model uncertainties and hence to decide the level of autonomy given to AI systems;
4. Issue **I6** of justice principle by means of federated learning;
5. Issue **I7** of explicability principle by means of interpretable machine and deep learning models.

### 1.5.2. Information Theoretic Quantification of Privacy, Interpretability, and Transferability

The most important feature of our approach is that the notions of privacy, interpretability, and transferability are quantified by information theoretic measures allowing for the study and optimization of trade-offs (such as trade-off between privacy and transferability or trade-off between privacy and interpretability) in a practical manner.

### 1.5.3. Computation of Information Theoretic Measures without Requiring the Knowledge of Data Distributions

It is possible to derive analytical expressions for the defined measures provided that the knowledge regarding the data distributions is available. However, in practice, the data distributions are unknown, and thus, a way to approximate the defined measures is required. Therefore, a novel method that employs recently introduced membership mappings [14–22], is presented for approximating the defined privacy leakage, interpretability, and transferability measures. The method relies on inferring a variational Bayesian model that facilitates an analytical approximation of the information theoretic measures through variational optimization methodology. A computational algorithm is provided for practically calculating the privacy leakage, interpretability, and transferability measures. Finally, an algorithm is presented that provides

1. Information theoretic evaluation of privacy leakage, interpretability, and transferability in a semi-supervised transfer and multi-task learning scenario;
2. An adversary model for estimating private data and for simulating privacy attacks; and
3. An interpretability model for estimating interpretable parameters and for providing an interpretation to the non-interpretable data vectors.

## 1.6. Organization

This text is organized into sections. The proposed methodology relies on the membership mappings for data representation learning. Therefore, Section 2 has been dedicated to the review of membership mappings. An application of membership mappings to solve an inverse modeling problem by developing a variational membership-mapping Bayesian model is considered in Section 3. Section 4 presents the most important result of this study on the variational approximation of information leakage and development of a computational algorithm for calculating information leakage. The measures for privacy leakage, interpretability, and transferability are formally introduced in Section 5. Section 5 further provides an algorithm to study the privacy, interpretability, and transferability aspects in a unified manner. The application of proposed measures to study the trade-offs is also demonstrated through the experiments made on the widely used MNIST and “Office+Caltech256” datasets in Section 6. Section 6 further considers a biomedical application concerned with the detection of mental stress in individuals using heart rate variability analysis. Finally, the concluding remarks are provided in Section 7.

## 2. Mathematical Background

This section reviews the membership mappings and transferable deep learning from [14,15,22]. For a detailed mathematical study of the concepts used in this section, the readers are referred to previous works [14,15,22].

### 2.1. Notations

- Let  $n, N, p, M \in \mathbb{N}$ .
- Let  $\mathcal{B}(\mathbb{R}^N)$  denote the Borel  $\sigma$ -algebra on  $\mathbb{R}^N$ , and let  $\lambda^N$  denote the Lebesgue measure on  $\mathcal{B}(\mathbb{R}^N)$ .
- Let  $(\mathcal{X}, \mathcal{A}, \rho)$  be a probability space with unknown probability measure  $\rho$ .
- Let us denote by  $\mathcal{S}$  the set of finite samples of data points drawn i.i.d. from  $\rho$ , i.e.,

$$\mathcal{S} := \{(x^i \sim \rho)_{i=1}^N \mid N \in \mathbb{N}\}. \quad (1)$$

- For a sequence  $x = (x^1, \dots, x^N) \in \mathcal{S}$ , let  $|x|$  denote the cardinality, i.e.,  $|x| = N$ .
- If  $x = (x^1, \dots, x^N)$ ,  $a = (a^1, \dots, a^M) \in \mathcal{S}$ , then  $x \wedge a$  denotes the concatenation of the sequences  $x$  and  $a$ , i.e.,  $x \wedge a = (x^1, \dots, x^N, a^1, \dots, a^M)$ .
- Let us denote by  $\mathbb{F}(\mathcal{X})$  the set of  $\mathcal{A}$ - $\mathcal{B}(\mathbb{R})$  measurable functions  $f: \mathcal{X} \rightarrow \mathbb{R}$ , i.e.,

$$\mathbb{F}(\mathcal{X}) := \{f: \mathcal{X} \rightarrow \mathbb{R} \mid f \text{ is } \mathcal{A}\text{-}\mathcal{B}(\mathbb{R}) \text{ measurable}\}. \quad (2)$$

- For convenience, the values of a function  $f \in \mathbb{F}(\mathcal{X})$  at points in the collection  $x = (x^1, \dots, x^N)$  are represented as  $f(x) = (f(x^1), \dots, f(x^N))$ .
- Let  $\zeta_x: \mathbb{R}^{|x|} \rightarrow [0, 1]$  be a membership function satisfying the following properties:

**Nowhere Vanishing:**  $\zeta_x(y) > 0$  for all  $y \in \mathbb{R}^{|x|}$ , i.e.,

$$\text{supp}[\zeta_x] = \mathbb{R}^{|x|}. \quad (3)$$

**Positive and Bounded Integrals:** The functions  $\zeta_x$  are absolutely continuous and Lebesgue integrable over the whole domain such that for all  $x \in \mathcal{S}$ , we have

$$0 < \int_{\mathbb{R}^{|x|}} \zeta_x \, d\lambda^{|x|} < \infty. \quad (4)$$

**Consistency of Induced Probability Measure:** The membership function induced probability measures  $\mathbb{P}_{\zeta_x}$ , defined on any  $A \in \mathcal{B}(\mathbb{R}^{|x|})$ , as

$$\mathbb{P}_{\zeta_x}(A) := \frac{1}{\int_{\mathbb{R}^{|x|}} \zeta_x \, d\lambda^{|x|}} \int_A \zeta_x \, d\lambda^{|x|} \quad (5)$$

are consistent in the sense that for all  $x, a \in \mathcal{S}$ :

$$\mathbb{P}_{\zeta_{x \wedge a}}(A \times \mathbb{R}^{|a|}) = \mathbb{P}_{\zeta_x}(A). \quad (6)$$

The collection of membership functions satisfying the aforementioned assumptions is denoted by

$$\Theta := \{\zeta_x: \mathbb{R}^{|x|} \rightarrow [0, 1] \mid (3), (4), (6), x \in \mathcal{S}\}. \quad (7)$$

### 2.2. Review of Variational Membership Mappings

**Definition 1** (Student-t Membership Mapping [14]). A Student-t membership-mapping,  $\mathcal{F} \in \mathbb{F}(\mathcal{X})$ , is a mapping with input space  $\mathcal{X} = \mathbb{R}^n$  and a membership function  $\zeta_x \in \Theta$  that is Student-t like:

$$\zeta_x(y) = \left(1 + 1/(v-2)(y - m_y)^T K_{xx}^{-1}(y - m_y)\right)^{-\frac{v+|x|}{2}} \quad (8)$$



where  $x \in \mathcal{S}$ ,  $y \in \mathbb{R}^{|\mathcal{X}|}$ ,  $v \in \mathbb{R}_+ \setminus [0, 2]$  is the degrees of freedom,  $m_y \in \mathbb{R}^{|\mathcal{X}|}$  is the mean vector, and  $K_{xx} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$  is the covariance matrix with its  $(i, j)$ -th element given as

$$(K_{xx})_{i,j} = kr(x^i, x^j) \quad (9)$$

where  $kr : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  is a positive definite kernel function defined as

$$kr(x^i, x^j) = \sigma^2 \exp \left( -0.5 \sum_{k=1}^n w_k |x_k^i - x_k^j|^2 \right) \quad (10)$$

where  $x_k^i$  is the  $k$ -th element of  $x^i$ ,  $\sigma^2$  is the variance parameter, and  $w_k \geq 0$  (for  $k \in \{1, \dots, n\}$ ).

Given a dataset  $\{(x^i, y^i) \mid x^i \in \mathbb{R}^n, y^i \in \mathbb{R}^p, i \in \{1, \dots, N\}\}$ , it is assumed that there exist zero-mean Student-t membership mappings  $\mathcal{F}_1, \dots, \mathcal{F}_p \in \mathbb{F}(\mathbb{R}^n)$  such that

$$y^i \approx [\mathcal{F}_1(x^i) \ \dots \ \mathcal{F}_p(x^i)]^T. \quad (11)$$

Under modeling scenario (11), [22] presents an algorithm (stated as Algorithm 1) for the variational learning of membership mappings.

---

**Algorithm 1** Variational learning of the membership mappings [22]

---

**Require:** Dataset  $\{(x^i, y^i) \mid x^i \in \mathbb{R}^n, y^i \in \mathbb{R}^p, i \in \{1, \dots, N\}\}$  and maximum possible number of auxiliary points  $M_{max} \in \mathbb{Z}_+$  with  $M_{max} \leq N$ .

- 1: Choose  $v$  and  $w = (w_1, \dots, w_n)$  as in (12) and (14), respectively.
  - 2: Choose a small positive value  $\kappa = 10^{-1}$ .
  - 3: Set iteration count  $it = 0$  and  $M|_0 = M_{max}$ .
  - 4: **while**  $\tau(M|_{it}, 1) < \kappa$  **do**
  - 5:    $M|_{it+1} = \lceil 0.9M|_{it} \rceil$
  - 6:    $it \leftarrow it + 1$
  - 7: **end while**
  - 8: Set  $M = M|_{it}$ .
  - 9: **if**  $\tau(M, 1) \geq \frac{1}{p} \sum_{j=1}^p \text{var}(y_j^1, \dots, y_j^N)$  **then**
  - 10:    $\sigma^2 = 1$
  - 11: **else**
  - 12:    $\sigma^2 = \frac{1}{\tau(M, 1)} \frac{1}{p} \sum_{j=1}^p \text{var}(y_j^1, \dots, y_j^N)$
  - 13: **end if**
  - 14: Compute  $a = \{a^m\}_{m=1}^M$  using (13),  $K_{xx}$  using (9),  $K_{aa}$  using (15), and  $K_{xa}$  using (16).
  - 15: Set  $\beta = 1$ .
  - 16: **repeat**
  - 17:   Compute  $\alpha$  using (18).
  - 18:   Update the value of  $\beta$  using (19).
  - 19: **until** ( $\beta$  nearly converges)
  - 20: Compute  $\alpha$  using (18).
  - 21: **return** the parameters set  $\mathbb{M} = \{\alpha, a, M, \sigma, w\}$ .
- 

With reference to Algorithm 1, we have following:

- The degrees of freedom associated to the Student-t membership mapping  $v \in \mathbb{R}_+ \setminus [0, 2]$  is chosen as

$$v = 2.1 \quad (12)$$

- The auxiliary inducing points are suggested to be chosen as the cluster centroids:

$$a = \{a^m\}_{m=1}^M = \text{cluster\_centroid}(\{x^i\}_{i=1}^N, M) \quad (13)$$

- where  $cluster\_centroid(\{x^i\}_{i=1}^N, M)$  represents the k-means clustering on  $\{x^i\}_{i=1}^N$ .
- The parameters  $(w_1, \dots, w_n)$  for kernel function (10) are chosen such that  $w_k$  (for  $k \in \{1, 2, \dots, n\}$ ) is given as

$$w_k = \left( \max_{1 \leq i \leq N} (x_k^i) - \min_{1 \leq i \leq N} (x_k^i) \right)^{-2} \quad (14)$$

- where  $x_k^i$  is the  $k$ -th element of vector  $x^i \in \mathbb{R}^n$ .
- $K_{aa} \in \mathbb{R}^{M \times M}$  and  $K_{xa} \in \mathbb{R}^{N \times M}$  are matrices with their  $(i, j)$ -th elements given as

$$(K_{aa})_{i,j} = kr(a^i, a^j) \quad (15)$$

$$(K_{xa})_{i,j} = kr(x^i, a^j) \quad (16)$$

- where  $kr : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  is a positive definite kernel function defined as in (10).
- The scalar-valued function  $\tau(M, \sigma^2)$  is defined as

$$\tau(M, \sigma^2) := \frac{Tr(K_{xx}) - Tr((K_{aa})^{-1} K_{xa}^T K_{xa})}{\nu + M - 2} \quad (17)$$

- where  $a$  is given by (13),  $\nu$  is given by (12), and parameters  $(w_1, \dots, w_n)$  (which are required to evaluate the kernel function for computing matrices  $K_{xx}$ ,  $K_{aa}$ , and  $K_{xa}$ ) are given by (14).
- $\alpha = [\alpha_1 \ \dots \ \alpha_p] \in \mathbb{R}^{M \times p}$  is a matrix with its  $j$ -th column defined as

$$\alpha_j := \left( K_{xa}^T K_{xa} + \frac{Tr(K_{xx}) - Tr((K_{aa})^{-1} K_{xa}^T K_{xa})}{\nu + M - 2} K_{aa} + \frac{K_{aa}}{\beta} \right)^{-1} (K_{xa})^T y_j \quad (18)$$

- The disturbance precision value  $\beta$  is iteratively estimated as

$$\frac{1}{\beta} = \frac{1}{pN} \sum_{j=1}^p \sum_{i=1}^N |y_j^i - \widehat{\mathcal{F}_j(x^i)}|^2 \quad (19)$$

where  $\widehat{\mathcal{F}_j(x^i)}$  is the estimated membership-mapping output given as

$$\widehat{\mathcal{F}_j(x^i)} = (G(x^i)) \alpha_j. \quad (20)$$

Here,  $G(x) \in \mathbb{R}^{1 \times M}$  is a vector-valued function defined as

$$G(x) := [kr(x, a^1) \ \dots \ kr(x, a^M)] \quad (21)$$

where  $kr : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  is defined as in (10).

**Definition 2** (Membership-Mappings Prediction [22]). Given the parameters set  $\mathbb{M} = \{\alpha, a, M, \sigma, w\}$  returned by Algorithm 1, the learned membership mappings could be used to predict output corresponding to any arbitrary input data point  $x \in \mathbb{R}^n$  as

$$\hat{y}(x; \mathbb{M}) = \alpha^T (G(x))^T \quad (22)$$

where  $G(\cdot) \in \mathbb{R}^{1 \times M}$  is a vector-valued function (21).

### 2.3. Review of Membership-Mappings-Based Conditionally Deep Autoencoders

**Definition 3** (Membership-Mapping Autoencoder [15]). A membership-mapping autoencoder,  $\mathcal{G} : \mathbb{R}^p \rightarrow \mathbb{R}^p$ , maps an input vector  $y \in \mathbb{R}^p$  to  $\mathcal{G}(y) \in \mathbb{R}^p$  such that

$$\mathcal{G}(y) \stackrel{\text{def}}{=} [\mathcal{F}_1(Py) \ \dots \ \mathcal{F}_p(Py)]^T, \quad (23)$$



where  $\mathcal{F}_j$  ( $j \in \{1, 2, \dots, p\}$ ) is a Student- $t$  membership-mapping,  $P \in \mathbb{R}^{n \times p}$  ( $n \leq p$ ) is a matrix such that the product  $Py$  is a lower-dimensional encoding for  $y$ .

**Definition 4** (Conditionally Deep Membership-Mapping Autoencoder (CDMMA) [15,22]). A conditionally deep membership-mapping autoencoder,  $\mathcal{D} : \mathbb{R}^p \rightarrow \mathbb{R}^p$ , maps a vector  $y \in \mathbb{R}^p$  to  $\mathcal{D}(y) \in \mathbb{R}^p$  through a nested composition of finite number of membership-mapping autoencoders such that

$$y^l = (\mathcal{G}_l \circ \dots \circ \mathcal{G}_2 \circ \mathcal{G}_1)(y), \forall l \in \{1, 2, \dots, L\} \quad (24)$$

$$l^* = \arg \min_{l \in \{1, 2, \dots, L\}} \|y - y^l\|^2 \quad (25)$$

$$\mathcal{D}(y) = y^{l^*}, \quad (26)$$

where  $\mathcal{G}_l(\cdot)$  is a membership-mapping autoencoder (Definition 3).

CDMMA discovers layers of increasingly abstract data representation with the lowest-level data features being modeled by the first layer and the highest-level data features being modeled by the end layer [15,22]. An algorithm (stated as Algorithm 2) has been provided in [15,22] for the variational learning of CDMMA.

---

**Algorithm 2** Variational learning of CDMMA [15,22]

---

**Require:** Dataset  $\mathbf{Y} = \{y^i \in \mathbb{R}^p \mid i \in \{1, \dots, N\}\}$ ; the subspace dimension  $n \in \{1, 2, \dots, p\}$ ; maximum number of auxiliary points  $M_{\max} \in \mathbb{Z}_+$  with  $M_{\max} \leq N$ ; the number of layers  $L \in \mathbb{Z}_+$ .

- 1: **for**  $l = 1$  to  $L$  **do**
- 2: Set subspace dimension associated to  $l$ -th layer as  $n_l = \max(n - l + 1, 1)$ .
- 3: Define  $P^l \in \mathbb{R}^{n_l \times p}$  such that the  $i$ -th row of  $P^l$  is equal to the transpose of eigenvector corresponding to the  $i$ -th largest eigenvalue of a sample covariance matrix of dataset  $\mathbf{Y}$ .
- 4: Define a latent variable  $x^{l,i} \in \mathbb{R}^{n_l}$ , for  $i \in \{1, \dots, N\}$ , as

$$x^{l,i} := \begin{cases} P^l y^i & \text{if } l = 1, \\ P^l \hat{y}^{l-1}(x^{l-1,i}; \mathbb{M}^{l-1}) & \text{if } l > 1 \end{cases} \quad (27)$$

where  $\hat{y}^{l-1}$  is the estimated output of the  $(l-1)$ -th layer computed using (22) for the parameters set  $\mathbb{M}^{l-1} = \{\alpha^{l-1}, a^{l-1}, M^{l-1}, \sigma^{l-1}, w^{l-1}\}$ .

- 5: Define  $M_{\max}^l$  as

$$M_{\max}^l := \begin{cases} M_{\max} & \text{if } l = 1, \\ M^{l-1} & \text{if } l > 1 \end{cases} \quad (28)$$

- 6: Compute parameters set  $\mathbb{M}^l = \{\alpha^l, a^l, M^l, \sigma^l, w^l\}$ , characterizing the membership mappings associated to the  $l$ -th layer, using Algorithm 1 on dataset  $\{(x^{l,i}, y^i) \mid i \in \{1, \dots, N\}\}$  with the maximum possible number of auxiliary points  $M_{\max}^l$ .

7: **end for**

- 8: **return** the parameters set  $\mathcal{M} = \{\{\mathbb{M}^1, \dots, \mathbb{M}^L\}, \{P^1, \dots, P^L\}\}$ .
- 

**Definition 5** (CDMMA Filtering [15,22]). Given a CDMMA with its parameters being represented by a set  $\mathcal{M} = \{\{\mathbb{M}^1, \dots, \mathbb{M}^L\}, \{P^1, \dots, P^L\}\}$ , the autoencoder can be applied for filtering a given input vector  $y \in \mathbb{R}^p$  as follows:

$$x^l(y; \mathcal{M}) = \begin{cases} P^l y, & l = 1 \\ P^l \hat{y}^{l-1}(x^{l-1}; \mathbb{M}^{l-1}) & l \geq 2 \end{cases} \quad (29)$$

Here,  $\hat{y}^{l-1}$  is the output of the  $(l-1)$ -th layer estimated using (22). Finally, CDMMA's output,  $\mathcal{D}(y; \mathcal{M})$ , is given as

$$\widehat{\mathcal{D}}(y; \mathcal{M}) = \hat{y}^{l^*}(x^{l^*}; \mathbb{M}^{l^*}) \quad (30)$$

$$l^* = \arg \min_{l \in \{1, \dots, L\}} \|y - \hat{y}^l(x^l; \mathbb{M}^l)\|^2. \quad (31)$$

For a big dataset, the computational time required by Algorithm 2 for learning will be high. To circumvent the problem of large computation time for processing big data, it is suggested in [15,22] that the data be partitioned into subsets and corresponding to each data subset, a separate CDMMA is learned. This motivates the defining of a wide CDMMA as in Definition 6. For the variational learning of wide CDMMA, Algorithm 3 follows from [15,22], where the choice of number of subsets as  $S = \lceil N/1000 \rceil$  is driven by the consideration that each subset contains around 1000 data points, since processing the data points up to 1000 by CDMMA is not computationally a challenge.

**Definition 6** (A Wide CDMMA [15,22]). A wide CDMMA,  $\mathcal{WD} : \mathbb{R}^p \rightarrow \mathbb{R}^p$ , maps a vector  $y \in \mathbb{R}^p$  to  $\mathcal{WD}(y) \in \mathbb{R}^p$  through a parallel composition of  $S$  ( $S \in \mathbb{Z}_+$ ) number of CDMMA's such that

$$\mathcal{WD}(y) = \mathcal{D}_{s^*}(y) \quad (32)$$

$$s^* = \arg \min_{s \in \{1, 2, \dots, S\}} \|y - \mathcal{D}_s(y)\|^2, \quad (33)$$

where  $\mathcal{D}_s(y)$  is the output of the  $s$ -th CDMMA.

---

#### Algorithm 3 Variational learning of wide CDMMA [15,22]

---

**Require:** Dataset  $\mathbf{Y} = \{y^i \in \mathbb{R}^p \mid i \in \{1, \dots, N\}\}$ ; the subspace dimension  $n \in \{1, 2, \dots, p\}$ ; ratio  $r_{max} \in (0, 1]$ ; the number of layers  $L \in \mathbb{Z}_+$ .

- 1: Apply k-means clustering to partition  $\mathbf{Y}$  into  $S$  subsets,  $\{\mathbf{Y}^1, \dots, \mathbf{Y}^S\}$ , where  $S = \lceil N/1000 \rceil$ .
  - 2: **for**  $s = 1$  to  $S$  **do**
  - 3: Build a CDMMA,  $\mathcal{M}^s$ , by applying Algorithm 2 on  $\mathbf{Y}^s$  taking  $n$  as the subspace dimension; maximum number of auxiliary points as equal to  $r_{max} \times \#\mathbf{Y}^s$  (where  $\#\mathbf{Y}^s$  is the number of data points in  $\mathbf{Y}^s$ ); and  $L$  is the number of layers.
  - 4: **end for**
  - 5: **return** the parameters set  $\mathcal{P} = \{\mathcal{M}^s\}_{s=1}^S$ .
- 

**Definition 7** (Wide CDMMA Filtering [15,22]). Given a wide CDMMA with its parameters being represented by a set  $\mathcal{P} = \{\mathcal{M}^s\}_{s=1}^S$ , the autoencoder can be applied for filtering a given input vector  $y \in \mathbb{R}^p$  as follows:

$$\widehat{\mathcal{WD}}(y; \mathcal{P}) = \widehat{\mathcal{D}}(y; \mathcal{M}^{s^*}) \quad (34)$$

$$s^* = \arg \min_{s \in \{1, 2, \dots, S\}} \|y - \widehat{\mathcal{D}}(y; \mathcal{M}^s)\|^2, \quad (35)$$

where  $\widehat{\mathcal{D}}(y; \mathcal{M}^s)$  is the output of the  $s$ -th CDMMA estimated using (30).

#### 2.4. Membership Mappings for Classification

A classifier (i.e., Definition 8) and an algorithm for its variational learning (stated as Algorithm 4) follows from [15,22].

**Definition 8** (A Classifier [15,22]). A classifier,  $\mathcal{C} : \mathbb{R}^p \rightarrow \{1, 2, \dots, C\}$ , maps a vector  $y \in \mathbb{R}^p$  to  $\mathcal{C}(y) \in \{1, 2, \dots, C\}$  such that

$$\mathcal{C}(y; \{\mathcal{P}_c\}_{c=1}^C) = \arg \min_{c \in \{1, 2, \dots, C\}} \|y - \widehat{\mathcal{WD}}(y; \mathcal{P}_c)\|^2 \quad (36)$$

where  $\widehat{WD}(y; \mathcal{P}_c)$ , computed using (34), is the output of the  $c$ -th wide CDMMA. The classifier assigns to an input vector the label of that class whose associated autoencoder best reconstructs the input vector.

---

**Algorithm 4** Variational learning of the classifier [15,22]
 

---

**Require:** Labeled dataset  $\mathbf{Y} = \{\mathbf{Y}_c \mid \mathbf{Y}_c = \{y^{i,c} \in \mathbb{R}^p \mid i \in \{1, \dots, N_c\}\}, c \in \{1, \dots, C\}\}$ ; the subspace dimension  $n \in \{1, \dots, p\}$ ; ratio  $r_{max} \in (0, 1]$ ; the number of layers  $L \in \mathbb{Z}_+$ .

- 1: **for**  $c = 1$  to  $C$  **do**
  - 2:   Build a wide CDMMA,  $\mathcal{P}_c = \{\mathcal{M}_c^s\}_{s=1}^{S_c}$ , by applying Algorithm 3 on  $\mathbf{Y}_c$  for the given  $n, r_{max}$ , and  $L$ .
  - 3: **end for**
  - 4: **return** the parameters set  $\{\mathcal{P}_c\}_{c=1}^C$ .
- 

### 2.5. Review of Membership-Mappings-Based Privacy-Preserving Transferable Learning

A privacy-preserving semi-supervised transfer and multi-task learning problem has been recently addressed in [22] by means of variational membership mappings. The method, as suggested in [22], involves the following steps:

#### 2.5.1. Optimal Noise Adding Mechanism for Differentially Private Classifiers

The approach suggested in [22] relies on a tailored noise adding mechanism to achieve a given level of differential privacy loss bound with the minimum perturbation of the data. In particular, Algorithm 5 is suggested for a differentially private approximation of data samples and Algorithm 6 is suggested for building a differentially private classifier.

---

**Algorithm 5** Differentially private approximation of data samples [22]
 

---

**Require:** Dataset  $\mathbf{Y} = \{y^i \in \mathbb{R}^p \mid i \in \{1, \dots, N\}\}$ ; differential privacy parameters:  $d \in \mathbb{R}_+$ ,  $\epsilon \in \mathbb{R}_+$ ,  $\delta \in (0, 1)$ .

- 1: A differentially private approximation of data samples is provided as

$$y_j^{+i} = y_j^i + F_{v_j^i}^{-1}(r_j^i; \epsilon, \delta, d), \quad r_j^i \in (0, 1) \quad (37)$$

$$F_{v_j^i}^{-1}(r_j^i; \epsilon, \delta, d) = \begin{cases} \frac{d}{\epsilon} \log\left(\frac{2r_j^i}{1-\delta}\right), & r_j^i < \frac{1-\delta}{2} \\ 0, & r_j^i \in \left[\frac{1-\delta}{2}, \frac{1+\delta}{2}\right], \quad r_j^i \in (0, 1). \\ -\frac{d}{\epsilon} \log\left(\frac{2(1-r_j^i)}{1-\delta}\right), & r_j^i > \frac{1+\delta}{2} \end{cases} \quad (38)$$

where  $y_j^{+i}$  is the  $j$ -th element of  $y^{+i} \in \mathbb{R}^p$ .

- 2: **return**  $\mathbf{Y}^+ = \{y^{+i} \in \mathbb{R}^p \mid i \in \{1, \dots, N\}\}$ .
- 

---

**Algorithm 6** Variational learning of a differentially private classifier [22]
 

---

**Require:** Differentially private approximated dataset:  $\mathbf{Y}^+ = \{\mathbf{Y}_c^+ \mid c \in \{1, \dots, C\}\}$ ; the subspace dimension  $n \in \{1, \dots, p\}$ ; ratio  $r_{max} \in (0, 1]$ ; the number of layers  $L \in \mathbb{Z}_+$ .

- 1: Build a classifier,  $\{\mathcal{P}_c^+\}_{c=1}^C$ , by applying Algorithm 4 on  $\mathbf{Y}^+$  for the given  $n, r_{max}$ , and  $L$ .
  - 2: **return**  $\{\mathcal{P}_c^+\}_{c=1}^C$ .
- 

#### 2.5.2. Semi-Supervised Transfer Learning Scenario

The aim is to transfer the knowledge extracted by a classifier trained using a source dataset to the classifier of the target domain such that the privacy of the source dataset is preserved. Let  $\{\mathbf{Y}_c^{sr}\}_{c=1}^C$  be the labeled source dataset where  $\mathbf{Y}_c^{sr} = \{y_{sr}^{i,c} \in \mathbb{R}^{p_{sr}} \mid i \in$

$\{1, \dots, N_c^{sr}\}$  represents the  $c$ -th labelled samples. The target dataset consist of a few labeled samples  $\{\mathbf{Y}_c^{tg}\}_{c=1}^C$  (with  $\mathbf{Y}_c^{tg} = \{y_{tg}^{i,c} \in \mathbb{R}^{p_{tg}} \mid i \in \{1, \dots, N_c^{tg}\}\}$ ) and another set of unlabeled samples  $\mathbf{Y}_*^{tg} = \{y_{tg}^{i,*} \in \mathbb{R}^{p_{tg}} \mid i \in \{1, \dots, N_*^{tg}\}\}$ .

### 2.5.3. Differentially Private Source Domain Classifier

For a given differential privacy parameters:  $d, \epsilon, \delta$ ; Algorithm 5 is applied on  $\mathbf{Y}_c^{sr}$  to obtain the differentially private approximated data samples,  $\mathbf{Y}_c^{+sr} = \{y_{sr}^{+i,c} \in \mathbb{R}^{p_{sr}} \mid i \in \{1, \dots, N_c^{sr}\}\}$ , for all  $c \in \{1, \dots, C\}$ . Algorithm 6 is applied on  $\{\mathbf{Y}_c^{+sr}\}_{c=1}^C$  to build a differentially private source domain classifier characterized by parameters sets  $\{\mathcal{P}_c^{+sr}\}_{c=1}^C$ .

### 2.5.4. Latent Subspace Transformation Matrices

For a given subspace dimension  $n_{st} \in \{1, 2, \dots, \min(p_{sr}, p_{tg})\}$ , the source domain transformation matrix  $V^{+sr} \in \mathbb{R}^{n_{st} \times p_{sr}}$  is defined as with its  $i$ -th row equal to the transpose of the eigenvector corresponding to the  $i$ -th largest eigenvalue of the sample covariance matrix computed on differentially private approximated source samples. The target domain transformation matrix  $V^{tg} \in \mathbb{R}^{n_{st} \times p_{tg}}$  is defined as with its  $i$ -th row equal to the transpose of the eigenvector corresponding to the  $i$ -th largest eigenvalue of the sample covariance matrix computed on target samples.

### 2.5.5. Subspace Alignment

A target sample is mapped to source-data-space via following transformation:

$$y_{tg \rightarrow sr}(y_{tg}) = \begin{cases} y_{tg}, & p_{sr} = p_{tg} \\ (V^{+sr})^T V^{tg} y_{tg}, & p_{sr} \neq p_{tg} \end{cases} \quad (39)$$

Both labeled and unlabeled target datasets are transformed to define the following sets:

$$\mathbf{Y}_c^{tg \rightarrow sr} := \{y_{tg \rightarrow sr}(y_{tg}) \mid y_{tg} \in \mathbf{Y}_c^{tg}\} \quad (40)$$

$$\mathbf{Y}_*^{tg \rightarrow sr} := \{y_{tg \rightarrow sr}(y_{tg}) \mid y_{tg} \in \mathbf{Y}_*^{tg}\}. \quad (41)$$

### 2.5.6. Target Domain Classifier

The  $k$ -th iteration for building the target domain classifier, where  $k \in \{1, \dots, it\_max\}$ , consists of the following updates:

$$\{\mathcal{P}_c^{tg} \mid k\}_{c=1}^C = \text{Algorithm 4} \left( \left\{ \mathbf{Y}_c^{tg \rightarrow sr} \cup \mathbf{Y}_{*,c}^{tg \rightarrow sr} \mid k-1 \right\}_{c=1}^C, n|_k, r_{max}, L \right) \quad (42)$$

$$\mathbf{Y}_{*,c}^{tg \rightarrow sr} \mid k = \left\{ y_{tg \rightarrow sr}^{i,*} \in \mathbf{Y}_*^{tg \rightarrow sr} \mid \mathcal{C}(y_{tg \rightarrow sr}^{i,*}; \{\mathcal{P}_c^{tg} \mid k\}_{c=1}^C) = c, i \in \{1, \dots, N_*^{tg}\} \right\} \quad (43)$$

where  $\{n|_1, n|_2, \dots\}$  is a monotonically non-decreasing sequence.

### 2.5.7. source2target Model

The mapping from source to target domain is learned by means of a variational membership-mappings-based model as in the following:

$$\mathbb{M}^{sr \rightarrow tg} = \text{Algorithm 1}(\mathcal{D}, M_{max}) \quad (44)$$

$$\mathcal{D} := \left\{ \left( \widehat{\mathcal{WD}}(y; \mathcal{P}_c^{+sr}), y \right) \mid y \in \left\{ \mathbf{Y}_c^{tg \rightarrow sr} \cup \mathbf{Y}_{*,c}^{tg \rightarrow sr} \mid it\_max \right\}, c \in \{1, \dots, C\} \right\} \quad (45)$$

$$M_{max} = \min(\lceil N^{tg}/2 \rceil, 1000) \quad (46)$$

where  $N^{tg} = |\mathcal{D}|$  is the total number of target samples,  $\widehat{\mathcal{WD}}(\cdot; \cdot)$  is defined as in (34),  $\mathbf{Y}_c^{tg \rightarrow sr}$  is defined as in (40), and  $\mathbf{Y}_{*,c}^{tg \rightarrow sr}$  is defined as in (43).

### 2.5.8. Transfer and Multi-Task Learning

Both source and target domain classifiers are combined with the source2target model for predicting the label associated to a target sample  $y_{tg \rightarrow sr}$  as

$$\hat{c}(y_{tg \rightarrow sr}; \{\mathcal{P}_c^{tg}\}_{c=1}^C, \{\mathcal{P}_c^{+sr}\}_{c=1}^C, \mathbb{M}^{sr \rightarrow tg}) = \arg \min_{c \in \{1, 2, \dots, C\}} \left\{ \min \left( \|y_{tg \rightarrow sr} - \widehat{\mathcal{WD}}(y_{tg \rightarrow sr}; \mathcal{P}_c^{tg})\|^2, \right. \right. \\ \left. \|y_{tg \rightarrow sr} - \hat{y}(\widehat{\mathcal{WD}}(y_{tg \rightarrow sr}; \mathcal{P}_c^{+sr}); \mathbb{M}^{sr \rightarrow tg})\|^2, \right. \\ \left. \|y_{tg \rightarrow sr} - \widehat{\mathcal{WD}}(y_{tg \rightarrow sr}; \mathcal{P}_c^{+sr})\|^2 \right\}. \quad (47)$$

where  $\hat{y}(\cdot; \mathbb{M}^{sr \rightarrow tg})$  is the output of the source2target model computed using (22).

### 3. Variational Membership-Mapping Bayesian Models

We consider the application of membership mappings to solve the inverse modeling problem related to  $x = f_{t \rightarrow x}(t)$ , where  $f_{t \rightarrow x} : \mathbb{R}^q \rightarrow \mathbb{R}^n$  is a forward map. Specifically, a membership-mappings model is used to approximate the inverse mapping  $f_{t \rightarrow x}^{-1}$ .

#### 3.1. A Prior Model

Given a dataset:  $\{(x^i, t^i) \mid i \in \{1, \dots, N\}\}$ , Algorithm 1 can be used to build a membership-mappings model characterized by a set of parameters, say  $\mathbb{M}^{x \rightarrow t} = \{\alpha^{x \rightarrow t}, a, M, \sigma, w\}$  (where  $x \rightarrow t$  indicates the mapping from  $x$  to  $t$  has been approximated by the membership mappings). It follows from (22) that the membership-mappings model predicted output corresponding to an input  $x$  is given as

$$\hat{t}(x; \mathbb{M}^{x \rightarrow t}) = (\alpha^{x \rightarrow t})^T (G(x))^T \quad (48)$$

where  $G(\cdot) \in \mathbb{R}^{1 \times M}$  is a vector-valued function defined as in (21). The  $k$ -th element of  $\hat{t}$  is given as

$$\hat{t}_k(x; \mathbb{M}^{x \rightarrow t}) = (G(x)) \alpha_k^{x \rightarrow t} \quad (49)$$

where  $\alpha_k^{x \rightarrow t}$  is the  $k$ -th column of matrix  $\alpha^{x \rightarrow t}$ .

Expression (49) allows estimating for any arbitrary  $x$  the corresponding  $t$  using a membership-mappings model. This motivates introducing the following prior model:

$$t_k = (G(x)) \theta_k + e_k \quad (50)$$

$$\theta_k \sim \mathcal{N}(\alpha_k^{x \rightarrow t}, \Lambda_k^{-1}) \quad (51)$$

$$e_k \sim \mathcal{N}(0, \gamma^{-1}) \quad (52)$$

$$\gamma \sim \text{Gamma}(a_\gamma, b_\gamma) \quad (53)$$

where  $k \in \{1, \dots, q\}$ ;  $\mathcal{N}(\alpha_k^{x \rightarrow t}, \Lambda_k^{-1})$  is the multivariate normal distribution with mean  $\alpha_k^{x \rightarrow t}$  and covariance  $\Lambda_k^{-1}$ ; and  $\text{Gamma}(a_\gamma, b_\gamma)$  is the Gamma distribution with shape parameter  $a_\gamma$  and rate parameter  $b_\gamma$ . The estimation provided by membership-mappings model  $\mathbb{M}^{x \rightarrow t}$  (i.e., (49)) is incorporated by the prior model (50)–(53), since

$$\mathbb{E}[t_k] = \hat{t}_k(x; \mathbb{M}^{x \rightarrow t}). \quad (54)$$

#### 3.2. Variational Bayesian Inference

Given the dataset,  $\{(x^i \in \mathbb{R}^n, t^i \in \mathbb{R}^q) \mid i \in \{1, 2, \dots, N\}\}$ , the variational Bayesian method is considered for an inference of the stochastic model (50), with priors as (51)–(53). For all  $i \in \{1, \dots, N\}$  and  $k \in \{1, \dots, q\}$ , we have

$$t_k^i = (G(x^i)) \theta_k + e_k^i, \quad (55)$$

where  $\theta_k \sim \mathcal{N}(\alpha_k^{x \rightarrow t}, \Lambda_k^{-1})$  and  $e_k^i \sim \mathcal{N}(0, \gamma^{-1})$ . Define  $\mathbf{t}_k \in \mathbb{R}^N$ ,  $\mathbf{e}_k \in \mathbb{R}^N$ , and  $R_x \in \mathbb{R}^{N \times M}$  as

$$\mathbf{t}_k = [t_k^1 \ \cdots \ t_k^N]^T \quad (56)$$

$$\mathbf{e}_k = [e_k^1 \ \cdots \ e_k^N]^T \quad (57)$$

$$R_x = \left[ \left( G(x^1) \right)^T \ \cdots \ \left( G(x^N) \right)^T \right]^T. \quad (58)$$

For all  $k \in \{1, \dots, q\}$ , we have

$$\mathbf{t}_k = R_x \theta_k + \mathbf{e}_k \quad (59)$$

$$p(\theta_k; \alpha_k^{x \rightarrow t}, \Lambda_k) = \frac{1}{\sqrt{(2\pi)^M |(\Lambda_k)^{-1}|}} \exp \left( -0.5 (\theta_k - \alpha_k^{x \rightarrow t})^T \Lambda_k (\theta_k - \alpha_k^{x \rightarrow t}) \right) \quad (60)$$

$$p(\mathbf{e}_k; \gamma) = \frac{1}{\sqrt{(2\pi)^N (\gamma)^{-N}}} \exp \left( -0.5 \gamma \|\mathbf{e}_k\|^2 \right) \quad (61)$$

$$p(\gamma; a_\gamma, b_\gamma) = \left( b_\gamma^{a_\gamma} / \Gamma(a_\gamma) \right) (\gamma)^{a_\gamma - 1} \exp(-b_\gamma \gamma). \quad (62)$$

Define the following sets:

$$\mathbf{t} = \{\mathbf{t}_1, \dots, \mathbf{t}_q\} \quad (63)$$

$$\theta = \{\theta_1, \dots, \theta_q\} \quad (64)$$

and consider the marginal probability of data  $\mathbf{t}$  which is given as

$$p(\mathbf{t}) = \int d\theta d\gamma p(\mathbf{t}, \theta, \gamma). \quad (65)$$

Let  $q(\theta, \gamma)$  be an arbitrary distribution. The log marginal probability of  $\mathbf{t}$  can be expressed as

$$\log(p(\mathbf{t})) = \int d\theta d\gamma q(\theta, \gamma) \log(p(\mathbf{t})) \quad (66)$$

$$= \int d\theta d\gamma q(\theta, \gamma) \log \left( \frac{p(\mathbf{t}, \theta, \gamma)}{q(\theta, \gamma)} \right) + \int d\theta d\gamma q(\theta, \gamma) \log \left( \frac{q(\theta, \gamma)}{p(\theta, \gamma | \mathbf{t})} \right). \quad (67)$$

Define

$$\mathcal{L}(q(\theta, \gamma), \mathbf{t}) := \int d\theta d\gamma q(\theta, \gamma) \log(p(\mathbf{t}, \theta, \gamma) / q(\theta, \gamma)) \quad (68)$$

to express (67) as

$$\log(p(\mathbf{t})) = \mathcal{L}(q(\theta, \gamma), \mathbf{t}) + \text{KL}(q(\theta, \gamma) \| p(\theta, \gamma | \mathbf{t})) \quad (69)$$

where KL is the Kullback–Leibler divergence of  $p(\theta, \gamma | \mathbf{t})$  from  $q(\theta, \gamma)$  and  $\mathcal{L}$ , referred to as negative free energy, provides a lower bound on the logarithmic evidence for the data.

The variational Bayesian approach minimizes the difference (in term of KL divergence) between variational and true posteriors via analytically maximizing negative free energy  $\mathcal{L}$  over variational distributions. However, the analytical derivation requires the following widely used mean-field approximation:

$$q(\theta, \gamma) = q(\theta)q(\gamma) \quad (70)$$

$$= q(\theta_1) \cdots q(\theta_q)q(\gamma). \quad (71)$$

Applying the standard variational optimization technique (as in [23–29]), it can be verified that the optimal variational distributions maximizing  $\mathcal{L}$  are as follows:

$$q^*(\theta_k) = \frac{1}{\sqrt{(2\pi)^M |(\hat{\Lambda}_k)^{-1}|}} \exp\left(-0.5(\theta_k - \hat{\mathbf{m}}_k)^T \hat{\Lambda}_k (\theta_k - \hat{\mathbf{m}}_k)\right) \quad (72)$$

$$q^*(\gamma) = \left((\hat{b}_\gamma)^{\hat{a}_\gamma} / \Gamma(\hat{a}_\gamma)\right) (\gamma)^{\hat{a}_\gamma - 1} \exp(-\hat{b}_\gamma \gamma) \quad (73)$$

where the parameters  $(\hat{\Lambda}_k, \hat{\mathbf{m}}_k, \hat{a}_\gamma, \hat{b}_\gamma)$  satisfy the following:

$$\hat{\Lambda}_k = \Lambda_k + (\hat{a}_\gamma / \hat{b}_\gamma) (R_x)^T R_x \quad (74)$$

$$\hat{\mathbf{m}}_k = (\hat{\Lambda}_k)^{-1} \left( \Lambda_k \alpha_k^{x \rightarrow t} + (\hat{a}_\gamma / \hat{b}_\gamma) (R_x)^T \mathbf{t}_k \right) \quad (75)$$

$$\hat{a}_\gamma = a_\gamma + 0.5qN \quad (76)$$

$$\hat{b}_\gamma = b_\gamma + 0.5 \sum_{k=1}^q \left\{ \|\mathbf{t}_k - R_x \hat{\mathbf{m}}_k\|^2 + \text{Tr} \left( (\hat{\Lambda}_k)^{-1} (R_x)^T R_x \right) \right\}. \quad (77)$$

Algorithm 7 is suggested for variational Bayesian inference of the model.

---

**Algorithm 7** Variational membership-mapping Bayesian model inference

---

**Require:** Dataset  $\{(x^i \in \mathbb{R}^n, t^i \in \mathbb{R}^q) \mid i \in \{1, \dots, N\}\}$  and maximum possible number of auxiliary points  $M_{\max} \in \mathbb{Z}_+$  with  $M_{\max} \leq N$ .

- 1: Apply Algorithm 1 on the dataset to build a variational membership-mappings model  $\mathbb{M}^{x \rightarrow t} = \{\alpha^{x \rightarrow t}, a, M, \sigma, w\}$ .
  - 2: Choose non-informative priors for covariance matrix, i.e.,  $\Lambda_k = 10^{-3} I_M, \forall k \in \{1, \dots, q\}$ .
  - 3: Choose non-informative priors for noise variance, i.e.,  $a_\gamma = 10^{-3}, b_\gamma = 10^{-3}$ .
  - 4: Initialize  $\hat{a}_\gamma / \hat{b}_\gamma = 1$ .
  - 5: **repeat**
  - 6:   update  $\{\hat{\Lambda}_k, \hat{\mathbf{m}}_k \mid k \in \{1, \dots, q\}\}, \hat{a}_\gamma, \hat{b}_\gamma$  using (74), (75), (76), (77).
  - 7: **until** convergence.
  - 8: **return** the parameters set  $\mathbb{B}\mathbb{M}^{x \rightarrow t} = \{\{\hat{\mathbf{m}}_k, \hat{\Lambda}_k \mid k \in \{1, \dots, q\}\}, \hat{a}_\gamma, \hat{b}_\gamma\}$ .
- 

The functionality of Algorithm 7 is as follows:

- Step 1 builds a variational membership-mappings model using Algorithm 1 from previous work [22].
- Algorithm 7 chooses at step 2 and 3 relatively non-informative priors.
- The loop between step 5 and step 7 applies variational Bayesian inference to iteratively estimate the parameters of optimal distributions until convergence.

**Remark 1** (Computational Complexity). *The computational complexity of Algorithm 7 is asymptotically dominated by the computation of inverse of  $M \times M$  dimensional matrix  $\hat{\Lambda}_k$  in (75) to calculate  $\hat{\mathbf{m}}_k$ . Thus, the computational complexity of Algorithm 7 is given as  $\mathcal{O}(M^3)$ , where  $M$  is the number of auxiliary points.*

The optimal distributions determined using Algorithm 7 define the so-called *Variational Membership-Mapping Bayesian Model* (VMMBM) as stated in Remark 2.



**Remark 2** (Variational Membership-Mapping Bayesian Model (VMMBM)). *The inverse mapping,  $f_{t \rightarrow x}^{-1}$ , is approximated as*

$$t_k = (G(x))\theta_k + e_k, \quad (78)$$

$$\theta_k \sim \mathcal{N}(\hat{\mathbf{m}}_k, \hat{\Lambda}_k^{-1}) \quad (79)$$

$$e_k \sim \mathcal{N}(0, \gamma^{-1}) \quad (80)$$

$$\gamma \sim \text{Gamma}(\hat{a}_\gamma, \hat{b}_\gamma) \quad (81)$$

where  $k \in \{1, \dots, q\}$  and  $(\hat{\mathbf{m}}_k, \hat{\Lambda}_k, \hat{a}_\gamma, \hat{b}_\gamma)$  are returned by Algorithm 7.

**Remark 3** (Estimation by VMMBM). *Given any  $x^*$ , the variational membership-mapping Bayesian model  $\mathbb{BM}^{x \rightarrow t}$  (returned by Algorithm 7) can be used to estimate corresponding  $t^*$  (such that  $x^* = f_{t \rightarrow x}(t^*)$ ) as*

$$\tilde{t}(x^*; \mathbb{BM}^{x \rightarrow t}) = [(G(x))\hat{\mathbf{m}}_1 \dots (G(x))\hat{\mathbf{m}}_q]^T. \quad (82)$$

#### 4. Evaluation of the Information-Leakage

Consider a scenario where a variable  $t$  is related to another variable  $x$  through a mapping  $f_{t \rightarrow x}$  such that  $x = f_{t \rightarrow x}(t)$ . The mutual information  $I(t; x)$  measures the amount of information obtained about variable  $t$  through observing variable  $x$ . Since  $x = f_{t \rightarrow x}(t)$ , the entropy  $H(t)$  remains fixed independent of mapping  $f_{t \rightarrow x}$ , and thus, the quantity  $I(t; x) - H(t)$  is a measure of the amount of information about  $t$  leaked by the mapping  $f_{t \rightarrow x}$ .

**Definition 9** (Information Leakage). *Under the scenario that  $x = f_{t \rightarrow x}(t)$ , a measure of the amount of information about  $t$  leaked by the mapping  $f_{t \rightarrow x}$  is defined as*

$$IL_{f_{t \rightarrow x}} := I(t; f_{t \rightarrow x}(t)) - H(t) \quad (83)$$

$$= I(t; x) - H(t). \quad (84)$$

The quantity  $IL_{f_{t \rightarrow x}}$  is referred to as the information leakage.

This section is dedicated to answering the question: *How to calculate the information leakage without knowing data distributions?*

##### 4.1. Variational Approximation of the Information Leakage

The mutual information between  $t$  and  $x$  is given as

$$I(t; x) = H(t) - H(t|x) \quad (85)$$

$$= H(t) + \int p(t, x) \log(p(t|x)) \, dt \, dx \quad (86)$$

$$= H(t) + \langle \log(p(t|x)) \rangle_{p(t, x)} \quad (87)$$

where  $\langle g(x) \rangle_{p(x)}$  denotes the expectation of a function of random variable  $g(x)$  with respect to the probability density function  $p(x)$ ;  $H(t)$  and  $H(t|x)$  are marginal and conditional entropies, respectively. Consider the conditional probability of  $t$  which is given as

$$p(t|x) = \int d\theta \, d\gamma \, p(\theta, \gamma, t|x) \quad (88)$$

where  $\theta$  is a set defined as in (64). Let  $q(\theta, \gamma)$  be an arbitrary distribution. The log conditional probability of  $t$  can be expressed as

$$\log(p(t|x)) = \int d\theta d\gamma q(\theta, \gamma) \log(p(t|x)) \quad (89)$$

$$= \int d\theta d\gamma q(\theta, \gamma) \log\left(\frac{p(\theta, \gamma, t|x)}{p(\theta, \gamma|t, x)}\right) \quad (90)$$

$$= \int d\theta d\gamma q(\theta, \gamma) \log\left(\frac{p(\theta, \gamma, t|x)}{q(\theta, \gamma)}\right) + \int d\theta d\gamma q(\theta, \gamma) \log\left(\frac{q(\theta, \gamma)}{p(\theta, \gamma|t, x)}\right). \quad (91)$$

Define

$$\mathcal{L}(q(\theta, \gamma), t, x) := \int d\theta d\gamma q(\theta, \gamma) \log\left(\frac{p(\theta, \gamma, t|x)}{q(\theta, \gamma)}\right) \quad (92)$$

to express (91) as

$$\log(p(t|x)) = \mathcal{L}(q(\theta, \gamma), t, x) + \text{KL}(q(\theta, \gamma) \| p(\theta, \gamma|t, x)) \quad (93)$$

where KL is Kullback–Leibler divergence of  $p(\theta, \gamma|t, x)$  from  $q(\theta, \gamma)$ . Using (87),

$$I(t; x) = H(t) + \langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t, x)} + \langle \text{KL}(q(\theta, \gamma) \| p(\theta, \gamma|t, x)) \rangle_{p(t, x)}. \quad (94)$$

That is,

$$IL_{f_{t \rightarrow x}} = \langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t, x)} + \langle \text{KL}(q(\theta, \gamma) \| p(\theta, \gamma|t, x)) \rangle_{p(t, x)}. \quad (95)$$

Since Kullback–Leibler divergence is always non-zero, it follows from (95) that  $\langle \mathcal{L} \rangle_{p(t, x)}$  provides a lower bound on  $IL_{f_{t \rightarrow x}}$  i.e.,

$$IL_{f_{t \rightarrow x}} \geq \langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t, x)}. \quad (96)$$

Our approach to approximate  $IL_{f_{t \rightarrow x}}$  is to maximize its lower bound with respect to variational distribution  $q(\theta, \gamma)$ . That is, we seek to solve

$$\widehat{IL}_{f_{t \rightarrow x}} = \max_{q(\theta, \gamma)} \langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t, x)}. \quad (97)$$

**Result 1** (Analytical Expression for the Information Leakage). *Given the model (78)–(81),  $\widehat{IL}_{f_{t \rightarrow x}}$  is given as*

$$\begin{aligned} \widehat{IL}_{f_{t \rightarrow x}} = & -0.5q \log(2\pi) + 0.5q \{F(\bar{a}_\gamma) - \log(\bar{b}_\gamma)\} \\ & - \frac{\bar{a}_\gamma}{2\bar{b}_\gamma} \sum_{k=1}^q \left\langle |t_k - G(x)\bar{\mathbf{m}}_k|^2 \right\rangle_{p(t, x)} - \frac{\bar{a}_\gamma}{2\bar{b}_\gamma} \sum_{k=1}^q \left\langle \text{Tr}\left((\bar{\Lambda}_k)^{-1}(G(x))^T G(x)\right) \right\rangle_{p(x)} \\ & - \frac{1}{2} \sum_{k=1}^q \left\{ (\hat{\mathbf{m}}_k - \bar{\mathbf{m}}_k)^T \hat{\Lambda}_k (\hat{\mathbf{m}}_k - \bar{\mathbf{m}}_k) + \text{Tr}\left(\hat{\Lambda}_k (\bar{\Lambda}_k)^{-1}\right) - \log\left(\frac{|\bar{\Lambda}_k|^{-1}}{|\hat{\Lambda}_k|^{-1}}\right) \right\} + \frac{qM}{2} \\ & - \hat{a}_\gamma \log\left(\bar{b}_\gamma / \hat{b}_\gamma\right) + \log(\Gamma(\bar{a}_\gamma) / \Gamma(\hat{a}_\gamma)) - (\bar{a}_\gamma - \hat{a}_\gamma) \Psi(\bar{a}_\gamma) + (\bar{b}_\gamma - \hat{b}_\gamma) (\bar{a}_\gamma / \bar{b}_\gamma). \end{aligned} \quad (98)$$

Here,  $F(\cdot)$  is the digamma function and the parameters  $(\bar{\Lambda}_k, \bar{\mathbf{m}}_k, \bar{a}_\gamma, \bar{b}_\gamma)$  satisfy the following:

$$\bar{\Lambda}_k = \hat{\Lambda}_k + (\bar{a}_\gamma / \bar{b}_\gamma) \langle (G(x))^T G(x) \rangle_{p(x)} \quad (99)$$

$$\bar{\mathbf{m}}_k = (\bar{\Lambda}_k)^{-1} \left( \hat{\Lambda}_k \bar{\mathbf{m}}_k + \frac{\bar{a}_\gamma}{\bar{b}_\gamma} \langle (G(x))^T t_k \rangle_{p(t,x)} \right) \quad (100)$$

$$\bar{a}_\gamma = \hat{a}_\gamma + 0.5q \quad (101)$$

$$\bar{b}_\gamma = \hat{b}_\gamma + \frac{1}{2} \sum_{k=1}^q \langle |t_k - G(x) \bar{\mathbf{m}}_k|^2 \rangle_{p(t,x)} + \frac{1}{2} \sum_{k=1}^q \langle \text{Tr} \left( (\bar{\Lambda}_k)^{-1} (G(x))^T G(x) \right) \rangle_{p(x)}. \quad (102)$$

**Proof of Result 1.** Consider

$$\mathcal{L}(q(\theta, \gamma), t, x) = \langle \log(p(t|\theta, \gamma, x)) \rangle_{q(\theta, \gamma)} + \langle \log(p(\theta, \gamma)/q(\theta, \gamma)) \rangle_{q(\theta, \gamma)}. \quad (103)$$

It follows from (78) and (80) that

$$\log(p(t_k|\theta_k, \gamma, x)) = -0.5 \log(2\pi) + 0.5 \log(\gamma) - 0.5\gamma |t_k - G(x)\theta_k|^2. \quad (104)$$

Since  $t = [t_1 \ \cdots \ t_q]^T$ , we have

$$\log(p(t|\theta, \gamma, x)) = -0.5q \log(2\pi) + 0.5q \log(\gamma) - 0.5\gamma \sum_{k=1}^q |t_k - G(x)\theta_k|^2. \quad (105)$$

Using (105) and (70)–(71) in (103), we have

$$\begin{aligned} \mathcal{L}(q(\theta, \gamma), t, x) = & -\frac{q}{2} \log(2\pi) + \frac{q}{2} \langle \log(\gamma) \rangle_{q(\gamma)} - \frac{\langle \gamma \rangle_{q(\gamma)}}{2} \sum_{k=1}^q \langle |t_k - G(x)\theta_k|^2 \rangle_{q(\theta_k)} \\ & + \sum_{k=1}^q \left\langle \log \left( \frac{p(\theta_k; \hat{\mathbf{m}}_k, \hat{\Lambda}_k)}{q(\theta_k)} \right) \right\rangle_{q(\theta_k)} + \left\langle \log \left( \frac{p(\gamma; a_\gamma, b_\gamma)}{q(\gamma)} \right) \right\rangle_{q(\gamma)}. \end{aligned} \quad (106)$$

Thus,

$$\begin{aligned} \langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t,x)} = & -\frac{q}{2} \log(2\pi) + \frac{q}{2} \langle \log(\gamma) \rangle_{q(\gamma)} - \frac{\langle \gamma \rangle_{q(\gamma)}}{2} \sum_{k=1}^q \langle |t_k|^2 \rangle_{p(t)} \\ & - \frac{\langle \gamma \rangle_{q(\gamma)}}{2} \sum_{k=1}^q \left\langle (\theta_k)^T \langle (G(x))^T G(x) \rangle_{p(x)} \theta_k \right\rangle_{q(\theta_k)} + \langle \gamma \rangle_{q(\gamma)} \sum_{k=1}^q \left\langle (\theta_k)^T \langle (G(x))^T t_k \rangle_{p(t,x)} \right\rangle_{q(\theta_k)} \\ & + \sum_{k=1}^q \left\langle \log \left( \frac{p(\theta_k; \hat{\mathbf{m}}_k, \hat{\Lambda}_k)}{q(\theta_k)} \right) \right\rangle_{q(\theta_k)} + \left\langle \log \left( \frac{p(\gamma; a_\gamma, b_\gamma)}{q(\gamma)} \right) \right\rangle_{q(\gamma)}. \end{aligned} \quad (107)$$

Now,  $\langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t,x)}$  can be maximized with respect to  $q(\theta_k)$  and  $q(\gamma)$  using variational optimization. It can be seen that optimal distributions maximizing  $\langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t,x)}$  are given as

$$q^*(\theta_k) = \frac{1}{\sqrt{(2\pi)^M |(\bar{\Lambda}_k)^{-1}|}} \exp \left( -0.5(\theta_k - \bar{\mathbf{m}}_k)^T \bar{\Lambda}_k (\theta_k - \bar{\mathbf{m}}_k) \right) \quad (108)$$

$$q^*(\gamma) = ((\bar{b}_\gamma)^{\bar{a}_\gamma} / \Gamma(\bar{a}_\gamma)) (\gamma)^{\bar{a}_\gamma - 1} \exp(-\bar{b}_\gamma \gamma) \quad (109)$$

where the parameters  $(\bar{\Lambda}_k, \bar{\mathbf{m}}_k, \bar{a}_\gamma, \bar{b}_\gamma)$  satisfy (99)–(102). The maximum attained value of  $\langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t,x)}$  is given as

$$\begin{aligned} \max_{q(\theta, \gamma)} \langle \mathcal{L}(q(\theta, \gamma), t, x) \rangle_{p(t, x)} &= -0.5q \log(2\pi) + 0.5q \{F(\bar{a}_\gamma) - \log(\bar{b}_\gamma)\} - \frac{\bar{a}_\gamma}{2\bar{b}_\gamma} \sum_{k=1}^q \langle |t_k - G(x) \bar{\mathbf{m}}_k|^2 \rangle_{p(t, x)} \\ &\quad - \frac{\bar{a}_\gamma}{2\bar{b}_\gamma} \sum_{k=1}^q \langle \text{Tr}((\bar{\Lambda}_k)^{-1} (G(x))^T G(x)) \rangle_{p(x)} - \sum_{k=1}^q \text{KL}(q^*(\theta_k) \| p(\theta_k; \hat{\mathbf{m}}_k, \hat{\Lambda}_k)) \\ &\quad - \text{KL}(q^*(\gamma) \| p(\gamma; \hat{a}_\gamma, \hat{b}_\gamma)) \end{aligned}$$

where  $F(\cdot)$  is the digamma function. After substituting the maximum value in (97) and calculating Kullback–Leibler divergences, we obtain (98).  $\square$

#### 4.2. An Algorithm for the Computing of Information Leakage

Result 1 forms the basis of Algorithm 8 that computes the information leakage using available data samples.

---

**Algorithm 8** Estimation of information leakage,  $IL_{f_{t \rightarrow x}} = I(t; x) - H(t)$ , using variational approximation

---

**Require:** Dataset  $\{(x^i \in \mathbb{R}^n, t^i \in \mathbb{R}^q) \mid x^i = f_{t \rightarrow x}(t^i), i \in \{1, \dots, N\}\}$ .

- 1: Apply Algorithm 7 on  $\{(x^i, t^i) \mid i \in \{1, \dots, N\}\}$  with  $M_{max} = \min(\lceil N/2 \rceil, 1000)$  (i.e., constraining the maximum possible number of auxiliary points  $M_{max}$  below 1000 for computational efficiency) to obtain variational membership-mappings Bayesian model  $\mathbb{BM}^{x \rightarrow t} = \{\{\hat{\mathbf{m}}_k, \hat{\Lambda}_k \mid k \in \{1, \dots, q\}\}, \hat{a}_\gamma, \hat{b}_\gamma\}$ .
  - 2: Initialize  $\bar{a}/\bar{b}$ , e.g., as  $\bar{a}/\bar{b} = \hat{a}/\hat{b}$ .
  - 3: **repeat**
  - 4:   Update  $\{\bar{\Lambda}_k, \bar{\mathbf{m}}_k \mid k \in \{1, \dots, q\}\}, \bar{a}, \bar{b}$  using (99)–(102) where expectations  $\langle \cdot \rangle_{p(x)}$  and  $\langle \cdot \rangle_{p(t, x)}$  are approximated via sample averages.
  - 5: **until** convergence.
  - 6: Compute  $\widehat{IL}_{f_{t \rightarrow x}}$  using (98) where expectations  $\langle \cdot \rangle_{p(x)}$  and  $\langle \cdot \rangle_{p(t, x)}$  are approximated via sample averages.
  - 7: **return**  $\widehat{IL}_{f_{t \rightarrow x}}$  and the model  $\mathbb{BM}^{x \rightarrow t}$ .
- 

The functionality of Algorithm 8 is as follows:

- Step 1 applies Algorithm 7 for the inference of a variational membership-mappings Bayesian model.
- The loop between step 3 and step 5 recursively estimates the parameters  $(\{\bar{\Lambda}_k, \bar{\mathbf{m}}_k \mid k \in \{1, \dots, q\}\}, \bar{a}, \bar{b})$  using update rules (99)–(102).
- Step 6 computes the information leakage using (98).

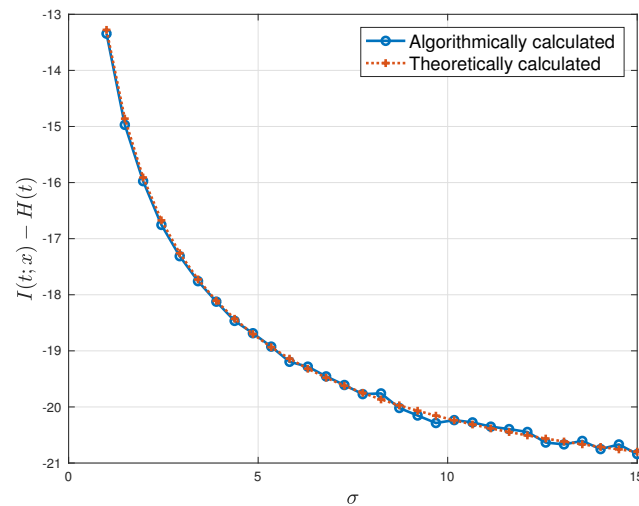
**Remark 4** (Computational Complexity). *The computational complexity of Algorithm 8 is asymptotically dominated by the computation of inverse of  $M \times M$  dimensional matrix  $\bar{\Lambda}_k$  in (100) to calculate  $\bar{\mathbf{m}}_k$ . Thus, the computational complexity of Algorithm 8 is given as  $\mathcal{O}(M^3)$ , where  $M$  is the number of auxiliary points.*

**Example 1** (Verification of Information Leakage Estimation Algorithm). *To demonstrate the effectiveness of Algorithm 8 in estimating information leakage, a scenario is generated where  $t \in \mathbb{R}^{10}$  and  $x \in \mathbb{R}^{10}$  are Gaussian distributed such that  $x = t + \omega$ ;  $t \sim \mathcal{N}(0, 5I_{10})$ ;  $\omega \sim \mathcal{N}(0, \sigma I_{10})$  with  $\sigma \in [1, 15]$ . Since the data distributions in this scenario are known, the information leakage can be theoretically calculated and is given as*

$$IL_{f_{t \rightarrow x}} = 5 \log(1 + 5/\sigma) - 0.5 \log(|(2\pi e 5 I_{10})|).$$

For a given value of  $\sigma$ , 1000 samples of  $t$  and  $x$  were simulated and Algorithm 8 was applied for estimating information leakage. The experiments were carried out at different values of  $\sigma$  ranging from 1 to 15.

Figure 3 compares the plots of estimated and theoretically calculated values of information leakage against  $\sigma$ . A close agreement between the two plots in Figure 3 verifies the effectiveness of Algorithm 8 in estimating information leakage without knowing the data distributions.



**Figure 3.** A comparison of the estimated information leakage values with the theoretically calculated values.

## 5. Information Theoretic Measures for Privacy Leakage, Interpretability, and Transferability

### 5.1. Definitions

To define formally the information theoretic measures for privacy leakage, interpretability, and transferability; a few variables and mappings are introduced in Table 2. Definitions 10–12 provide the mathematical definitions of the information theoretic measures.

**Table 2.** Introduced variables and mappings.

| Symbol/Mapping  | Definition/Meaning   |
|---|--|
| $x_{sr} \in \mathbb{R}^{n_{sr}}$  | vector representing private/sensitive variables associated to source domain  |
| $y_{sr} \in \mathbb{R}^{p_{sr}}$  | source domain data vector  |
| $t_{sr} \in \mathbb{R}^q$   | vector representing the set of interpretable parameters associated to non-interpretable data vector $y_{sr}$                               |
| $y_{sr}^+ \in \mathbb{R}^{p_{sr}}$  | noise-added data vector (that is either publicly released or used for the training of source model) obtained from $y_{sr}$ via Algorithm 5 |
| $f_{x_{sr} \rightarrow y_{sr}^+} : \mathbb{R}^{n_{sr}} \rightarrow \mathbb{R}^{p_{sr}}$ | mapping from private variables to noise-added data vector, i.e., $y_{sr}^+ = f_{x_{sr} \rightarrow y_{sr}^+}(x_{sr})$                      |
| $f_{t_{sr} \rightarrow y_{sr}^+} : \mathbb{R}^q \rightarrow \mathbb{R}^{p_{sr}}$        | mapping from interpretable parameters to noise-added data vector, i.e., $y_{sr}^+ = f_{t_{sr} \rightarrow y_{sr}^+}(t_{sr})$               |
| $\{\mathcal{P}_c^{+sr}\}_{c=1}^C$   | differentially private source domain autoencoders, representing data features of each of $C$ classes, obtained via Algorithm 6             |
| $y_{tg} \in \mathbb{R}^{p_{tg}}$  | target domain data vector  |
| $y_{tg \rightarrow sr} \in \mathbb{R}^{p_{sr}}$   | representation of target domain data vector $y_{tg}$ in source domain via transformation (39)  |
| $\{\mathcal{P}_c^{tg}\}_{c=1}^C$  | target domain autoencoders, representing data features of each of $C$ classes, obtained via Algorithm 6                                    |

Table 2. Cont.

| Symbol/Mapping  | Definition/Meaning   |
|---|--|
| $f_{y_{tg} \rightarrow c} : \mathbb{R}^{p_{tg}} \rightarrow \{1, \dots, C\}$                                | mapping assigning class label to target domain data vector $y_{tg}$ via (47), i.e.,<br>$f_{y_{tg} \rightarrow c}(y_{tg}) = \hat{c}(y_{tg \rightarrow sr}(y_{tg}); \{\mathcal{P}_c^{tg}\}_{c=1}^C, \{\mathcal{P}_c^{+sr}\}_{c=1}^C, \mathbb{M}^{sr \rightarrow tg})$  |
| $\hat{y}_{tg}^{sr} \in \mathbb{R}^{p_{sr}}$   | transformation of $y_{tg}$ to source domain and filtering through the autoencoder that represents the source domain feature vectors of the same class as that of $y_{tg}$ , i.e.,<br>$\hat{y}_{tg}^{sr} = \widehat{\mathcal{WD}}(y_{tg \rightarrow sr}(y_{tg}); \mathcal{P}_{f_{y_{tg} \rightarrow c}}^{+sr}(y_{tg}))$ |
| $\hat{y}_{tg}^{tg} \in \mathbb{R}^{p_{sr}}$   | transformation of $y_{tg}$ to source domain and filtering through the autoencoder that represents the target domain feature vectors of the same class as that of $y_{tg}$ , i.e.,<br>$\hat{y}_{tg}^{tg} = \widehat{\mathcal{WD}}(y_{tg \rightarrow sr}(y_{tg}); \mathcal{P}_{f_{y_{tg} \rightarrow c}}^{tg}(y_{tg}))$  |
| $f_{\hat{y}_{tg}^{sr} \rightarrow \hat{y}_{tg}^{tg}} : \mathbb{R}^{p_{sr}} \rightarrow \mathbb{R}^{p_{sr}}$ | mapping from source domain feature vector $\hat{y}_{tg}^{sr}$ to target domain feature vector $\hat{y}_{tg}^{tg}$ , i.e.,<br>$\hat{y}_{tg}^{tg} = f_{\hat{y}_{tg}^{sr} \rightarrow \hat{y}_{tg}^{tg}}(\hat{y}_{tg}^{sr})$  |

**Definition 10** (Privacy Leakage). *Privacy leakage (by the mapping from private variables to noise-added data vector) is a measure of the amount of information about private/sensitive variable  $x_{sr}$  leaked by the mapping  $f_{x_{sr} \rightarrow y_{sr}^+}$  and is defined as*

$$IL_{f_{x_{sr} \rightarrow y_{sr}^+}} := I(x_{sr}; f_{x_{sr} \rightarrow y_{sr}^+}(x_{sr})) - H(x_{sr}) \quad (110)$$

$$= I(x_{sr}; y_{sr}^+) - H(x_{sr}). \quad (111)$$

**Definition 11** (Interpretability Measure). *Interpretability (of noise-added data vector) is measured as the amount of information about interpretable parameters  $t_{sr}$  leaked by the mapping  $f_{t_{sr} \rightarrow y_{sr}^+}$  and is defined as*

$$IL_{f_{t_{sr} \rightarrow y_{sr}^+}} := I(t_{sr}; f_{t_{sr} \rightarrow y_{sr}^+}(t_{sr})) - H(t_{sr}) \quad (112)$$

$$= I(t_{sr}; y_{sr}^+) - H(t_{sr}). \quad (113)$$

**Definition 12** (Transferability Measure). *Transferability (from source domain data representation learning models (i.e.,  $\mathcal{P}_1^{+sr}, \dots, \mathcal{P}_C^{+sr}$ ) to the target domain data representation learning models (i.e.,  $\mathcal{P}_1^{tg}, \dots, \mathcal{P}_C^{tg}$ )) is measured as the amount of information about source domain feature vector  $\hat{y}_{tg}^{sr}$  leaked by the mapping  $f_{\hat{y}_{tg}^{sr} \rightarrow \hat{y}_{tg}^{tg}}$  and is defined as*

$$IL_{f_{\hat{y}_{tg}^{sr} \rightarrow \hat{y}_{tg}^{tg}}} := I(\hat{y}_{tg}^{sr}; f_{\hat{y}_{tg}^{sr} \rightarrow \hat{y}_{tg}^{tg}}(\hat{y}_{tg}^{sr})) - H(\hat{y}_{tg}^{sr}) \quad (114)$$

$$= I(\hat{y}_{tg}^{sr}; \hat{y}_{tg}^{tg}) - H(\hat{y}_{tg}^{sr}). \quad (115)$$

Here,  $\hat{y}_{tg}^{tg}$  represents the target domain feature vector and  $f_{\hat{y}_{tg}^{sr} \rightarrow \hat{y}_{tg}^{tg}} : \mathbb{R}^{p_{sr}} \rightarrow \mathbb{R}^{p_{sr}}$  is the mapping from source domain feature vector  $\hat{y}_{tg}^{sr}$  to target domain feature vector  $\hat{y}_{tg}^{tg}$ .

Since the defined measures are in the form of information leakages, Algorithm 8 could be directly applied for practically computing the measures provided the availability of data samples.

## 5.2. A Unified Approach to Privacy-Preserving Interpretable and Transferable Learning

The presented theory allows us to develop an algorithm that implements privacy-preserving interpretable and transferable learning methodology in a unified manner.

Algorithm 9 is presented for a systematic implementation of the proposed privacy-preserving interpretable and transferable deep learning methodology. The functionality of Algorithm 9 is as follows:

---

**Algorithm 9** Algorithm for privacy-preserving interpretable and transferable learning

---

**Require:** The labeled source dataset:  $\mathbf{Y}^{sr} = \{\mathbf{Y}_c^{sr}\}_{c=1}^C$  (where  $\mathbf{Y}_c^{sr} = \{y_{sr}^{i,c} \in \mathbb{R}^{p_{sr}} \mid i \in \{1, \dots, N_c^{sr}\}\}$  represents the  $c$ -th labeled samples); the set of private data:  $\mathbf{X}^{sr} = \{\mathbf{X}_c^{sr}\}_{c=1}^C$  (where  $\mathbf{X}_c^{sr} = \{x_{sr} \in \mathbb{R}^{n_{sr}} \mid x_{sr} = f_{x_{sr} \rightarrow y_{sr}}^{-1}(y_{sr}), y_{sr} \in \mathbf{Y}_c^{sr}\}$ ); the set of interpretable parameters:  $\mathbf{T}^{sr} = \{\mathbf{T}_c^{sr}\}_{c=1}^C$  (where  $\mathbf{T}_c^{sr} = \{t_{sr} \in \mathbb{R}^q \mid t_{sr} = f_{t_{sr} \rightarrow y_{sr}}^{-1}(y_{sr}), y_{sr} \in \mathbf{Y}_c^{sr}\}$ ); the set of a few labeled target samples:  $\{\mathbf{Y}_c^{tg}\}_{c=1}^C$  (where  $\mathbf{Y}_c^{tg} = \{y_{tg}^{i,c} \in \mathbb{R}^{p_{tg}} \mid i \in \{1, \dots, N_c^{tg}\}\}$  is the set of  $c$ -th labeled target samples); the set of unlabeled target samples:  $\mathbf{Y}_*^{tg} = \{y_{tg}^{i,*} \in \mathbb{R}^{p_{tg}} \mid i \in \{1, \dots, N_*^{tg}\}\}$ ; and the differential privacy parameters:  $d \in \mathbb{R}_+, \epsilon \in \mathbb{R}_+, \delta \in (0, 1)$ .

- 1: A differentially private approximation of source dataset,  $\mathbf{Y}^{+sr} = \{\mathbf{Y}_c^{+sr}\}_{c=1}^C$ , is obtained using Algorithm 5 on  $\mathbf{Y}^{sr}$ .
- 2: Differentially private source domain classifier,  $\{\mathcal{P}_c^{+sr}\}_{c=1}^C$ , is built using Algorithm 6 on  $\mathbf{Y}^{+sr}$  taking subspace dimension as equal to  $\min(20, p_{sr})$  (where  $p_{sr}$  is the dimension of source data samples), ratio  $r_{max}$  as equal to 0.5, and number of layers as equal to 5.
- 3: Taking subspace dimension  $n_{st} = \min(\lceil p_{sr}/2 \rceil, p_{tg})$ , the source domain transformation matrix  $V^{+sr} \in \mathbb{R}^{n_{st} \times p_{sr}}$  is defined as with its  $i$ -th row equal to the transpose of the eigenvector corresponding to the  $i$ -th largest eigenvalue of sample covariance matrix computed on differentially private approximated source samples. The target domain transformation matrix  $V^{tg} \in \mathbb{R}^{n_{st} \times p_{tg}}$  is defined as with its  $i$ -th row equal to the transpose of the eigenvector corresponding to the  $i$ -th largest eigenvalue of the sample covariance matrix computed on target samples.
- 4: For the case of heterogeneous source and target domains, the subspace alignment approach is used to transform target samples via (40) and (41) for defining the sets  $\{\mathbf{Y}_c^{tg \rightarrow sr}\}_{c=1}^C$  and  $\mathbf{Y}_*^{tg \rightarrow sr}$ .
- 5: Initial target domain classifier,  $\{\mathcal{P}_c^{tg} \mid 0\}_{c=1}^C$ , is built using Algorithm 4 on labeled target samples,  $\{\mathbf{Y}_c^{tg \rightarrow sr}\}_{c=1}^C$ , taking subspace dimension as equal to  $\min(20, \min_{1 \leq c \leq C} \{N_c^{tg} - 1\})$  (where  $N_c^{tg}$  is the number of  $c$ -th class labeled target samples), ratio  $r_{max}$  as equal to 1, and number of layers as equal to 1.
- 6: The target domain classifier is updated using (42) and (43) until 4 iterations taking the monotonically non-decreasing subspace dimension  $n$  sequence as  $\{\min(5, p_{sr}), \min(10, p_{sr}), \min(15, p_{sr}), \min(20, p_{sr})\}$  and  $r_{max}=0.5$ .
- 7: The mapping from source to target domain is learned by means of a model,  $\mathbb{M}^{sr \rightarrow tg}$ , defined as in (44).
- 8: Compute privacy leakage,  $IL_{f_{x_{sr} \rightarrow y_{sr}^+}}$ , and adversary model,  $\mathbb{B}\mathbb{M}^{y_{sr}^+ \rightarrow x_{sr}}$ , via applying Algorithm 8 on  $\{(y_{sr}^+, x_{sr}) \mid y_{sr}^+ = f_{x_{sr} \rightarrow y_{sr}^+}(x_{sr}), x_{sr} \in \mathbf{X}^{sr}, y_{sr}^+ \in \mathbf{Y}^{+sr}\}$ .
- 9: Compute interpretability measure,  $IL_{f_{t_{sr} \rightarrow y_{sr}^+}}$ , and interpretability model,  $\mathbb{B}\mathbb{M}^{y_{sr}^+ \rightarrow t_{sr}}$ , via applying Algorithm 8 on  $\{(y_{sr}^+, t_{sr}) \mid y_{sr}^+ = f_{t_{sr} \rightarrow y_{sr}^+}(t_{sr}), t_{sr} \in \mathbf{T}^{sr}, y_{sr}^+ \in \mathbf{Y}^{+sr}\}$ .
- 10: Compute transferability measure,  $IL_{f_{y_{tg}^{sr} \rightarrow y_{tg}^{tg}}}$ , via applying Algorithm 8 on  $\{(y_{tg}^{tg}, y_{tg}^{sr}) \mid y_{tg} \in \{\mathbf{Y}_c^{tg}\}_{c=1}^C \cup \mathbf{Y}_*^{tg}\}$ , where

$$\hat{y}_{tg}^{sr}(y_{tg}) = \widehat{\mathcal{W}\mathcal{D}}\left(y_{tg \rightarrow sr}(y_{tg}); \mathcal{P}_{f_{y_{tg} \rightarrow sr}}^{+sr}\right) \quad (116)$$

$$\hat{y}_{tg}^{tg}(y_{tg}) = \widehat{\mathcal{W}\mathcal{D}}\left(y_{tg \rightarrow sr}(y_{tg}); \mathcal{P}_{f_{y_{tg} \rightarrow sr}}^{tg}\right) \quad (117)$$

$$f_{y_{tg} \rightarrow sr}(y_{tg}) = \hat{c}\left(y_{tg \rightarrow sr}(y_{tg}); \{\mathcal{P}_c^{tg}\}_{c=1}^C, \{\mathcal{P}_c^{+sr}\}_{c=1}^C, \mathbb{M}^{sr \rightarrow tg}\right), \quad (118)$$

$y_{tg \rightarrow sr}(y_{tg})$  is defined as in (39), and  $\hat{c}(\cdot)$  is defined by (47).

- 11: **return** in the source domain: classifier  $\{\mathcal{P}_c^{+sr}\}_{c=1}^C$ ; privacy leakage  $IL_{f_{x_{sr} \rightarrow y_{sr}^+}}$  and adversary model  $\mathbb{B}\mathbb{M}^{y_{sr}^+ \rightarrow x_{sr}}$ ; interpretability measure  $IL_{f_{t_{sr} \rightarrow y_{sr}^+}}$  and interpretability model  $\mathbb{B}\mathbb{M}^{y_{sr}^+ \rightarrow t_{sr}}$ .
  - 12: **return** in the target domain: classifier  $\{\mathcal{P}_c^{tg}\}_{c=1}^C$ .
  - 13: **return** for transfer and multi-task learning scenario: classifiers  $\{\mathcal{P}_c^{+sr}\}_{c=1}^C$  and  $\{\mathcal{P}_c^{tg}\}_{c=1}^C$ ; source2target model  $\mathbb{M}^{sr \rightarrow tg}$ ; latent subspace transformation matrices  $V^{+sr}$  and  $V^{tg}$ ; transferability measure  $IL_{f_{y_{tg}^{sr} \rightarrow y_{tg}^{tg}}}$ .
- 

- Step 2 builds the differentially private source domain classifier following Algorithm 6 from previous work [22].
- Step 6 results in the building of the target domain classifier using the method of [22].
- An information theoretic evaluation of privacy leakage, interpretability, and transferability is undertaken at step 8, 9, and 10, respectively.
- Step 8 also provides the adversary model  $\mathbb{B}\mathbb{M}^{y_{sr}^+ \rightarrow x_{sr}}$ , which can be used to estimate private data and thus to simulate privacy attacks;
- Step 9 also provides the interpretability model  $\mathbb{B}\mathbb{M}^{y_{sr}^+ \rightarrow t_{sr}}$ , that can be used to estimate interpretable parameters and thus provide an interpretation to the non-interpretable data vectors.



## 6. Experiments

Experiments have been carried out to demonstrate the application of the proposed measures (for privacy leakage, interpretability, and transferability) to privacy-preserving interpretable and transferable learning. The methodology was implemented using MATLAB R2017b and the experiments have been made on an iMac (M1, 2021) machine with 8 GB RAM.

### 6.1. MNIST Dataset

The MNIST dataset contains  $28 \times 28$  sized images divided into a training set of 60,000 images and test set of 10,000 images. The images' pixel values were divided by 255 to normalize the values in the range from 0 to 1. The  $28 \times 28$  normalized pixel values of each image were flattened to an equivalent 784-dimensional data vector.

#### 6.1.1. Interpretable Parameters

For an MNIST digits dataset, there exist no additional interpretable parameters other than the pixel values. Thus, we defined corresponding to a pixel values vector  $y \in [0, 1]^{784}$  an interpretable parameter vector  $t \in \{0, 1\}^{10}$  such that the  $j$ -th element  $t_j = 1$ , if the  $j$ -th class label is associated to  $y$ ; otherwise,  $t_j = 0$ . That is, interpretable vector  $t$ , in our experimental setting, represents the class label assigned to data vector  $y$ .

#### 6.1.2. Private Data

Here, we assume that pixel values are private, i.e.,  $x_{sr} = y_{sr}$ .

#### 6.1.3. Semi-Supervised Transfer Learning Scenario

A transfer learning scenario was considered in the same setting as in [22,30] where 60,000 training samples constituted the source dataset; a set of 9000 test samples constituted the target dataset, and the classification performance was evaluated on the remaining 1000 test samples. Out of 9000 target samples, only 10 samples per class were labeled and the remaining 8900 target samples remained as unlabeled.

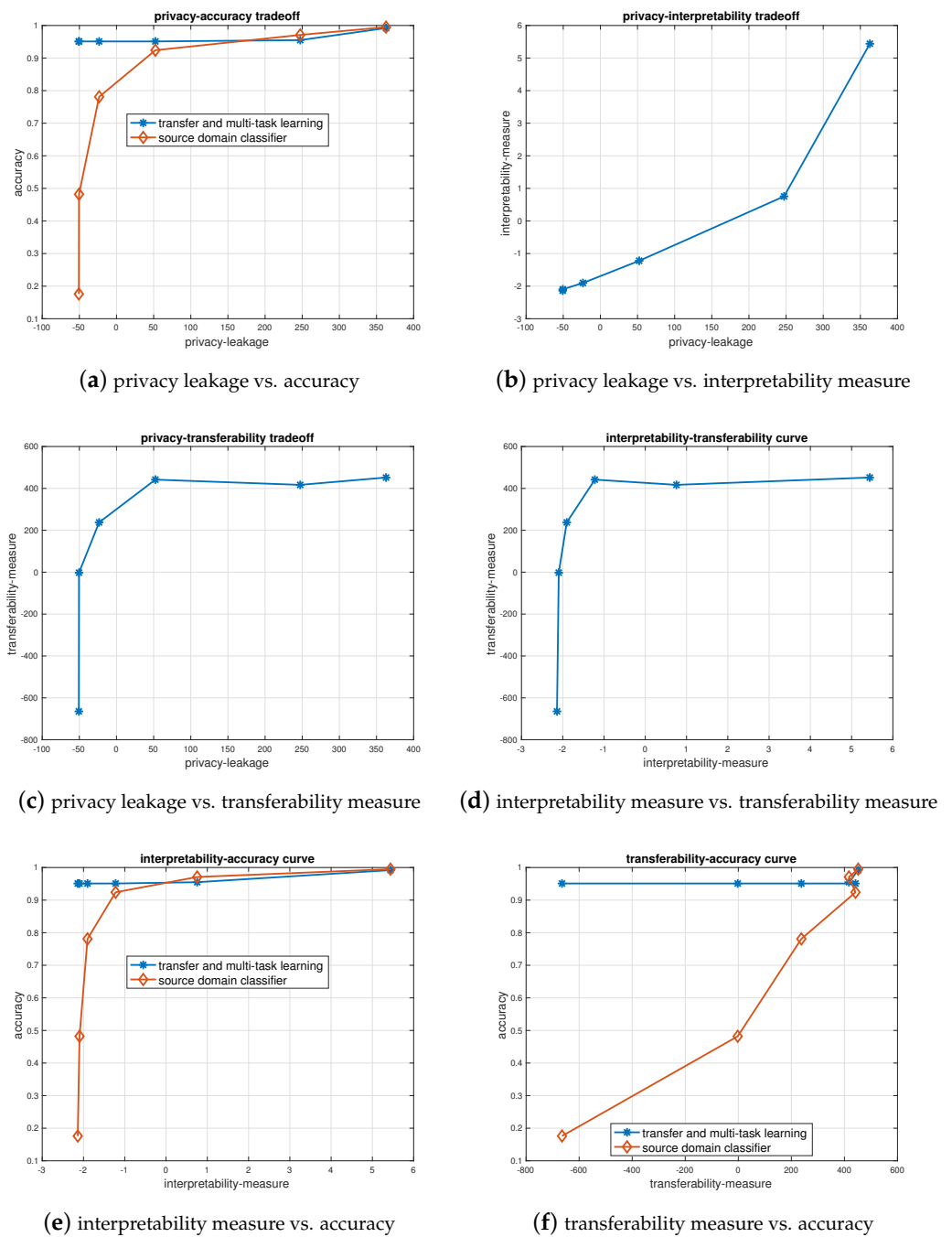
#### 6.1.4. Experimental Design

Algorithm 9 is applied with the differential privacy parameters as  $d = 1$  and  $\delta = 1 \times 10^{-5}$ . The experiment involves six different privacy-preserving semi-supervised transfer learning scenarios with privacy-loss bound values as  $\epsilon = 0.1$ ,  $\epsilon = 0.25$ ,  $\epsilon = 0.5$ ,  $\epsilon = 1$ ,  $\epsilon = 2$ , and  $\epsilon = 10$ . For the computation of privacy leakage, interpretability measure, and transferability measure in Algorithm 9, a subset consisting of 5000 randomly selected samples was considered.

#### 6.1.5. Results

The experimental results have been plotted in Figure 4. Figure 4a–c display the privacy–accuracy trade-off curve, privacy–interpretability trade-off curve, and privacy–transferability trade-off curve respectively. The following observations are made:

- As expected and observed in Figure 4f, the transferability measure is positively correlated with the accuracy of source–domain classifier on target test samples.
- Since we have defined the interpretable vector associated to a feature vector as representing the class label, the positive correlations of interpretability measure with the source domain classifier's accuracy and the transferability measure are observed in Figure 4e and Figure 4f, respectively.
- The results also verify the robust performance of Algorithm 9 under transfer and multi-task learning scenario, since the classification performance in the transfer and multi-task learning scenario, unlike the performance of the source domain classifier, is not adversely affected by a reduction in privacy leakage, interpretability measure, and transferability measure as observed in Figure 4a,e,f.



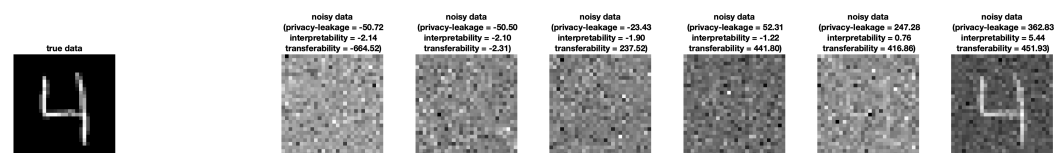
**Figure 4.** The plots between privacy leakage, interpretability measure, transferability measure, and accuracy for MNIST dataset.

Table 3 reports the results obtained by the models that correspond to minimum privacy leakage, maximum interpretability measure, and maximum transferability measure. The robustness of transfer and multi-task learning scenario is further highlighted in Table 3. To achieve the minimum value of privacy leakage, the accuracy of source domain classifier must be decreased to 0.1760; however, the transfer and multi-task learning scenario achieves the minimum privacy leakage value with the accuracy of 0.9510. As observed in Table 3, the maximum transferability-measure models also correspond to the maximum interpretability-measure models.

**Table 3.** Results of experiments on MNIST dataset for evaluating privacy leakage, interpretability, and transferability.

| Method  | Privacy Leakage | Interpretability Measure | Transferability Measure | Classification Accuracy |
|---|-----------------|--------------------------|-------------------------|-------------------------|
| minimum privacy leakage transfer and multi-task learning          | −50.72          | −2.14                    | −664.52                 | 0.9510                  |
| minimum privacy leakage source domain classifier                  | −50.72          | −2.14                    | −664.52                 | 0.1760                  |
| maximum interpretability measure transfer and multi-task learning | 362.83          | 5.44                     | 451.93                  | 0.9920                  |
| maximum interpretability measure source domain classifier         | 362.83          | 5.44                     | 451.93                  | 0.9950                  |
| maximum transferability measure transfer and multi-task learning  | 362.83          | 5.44                     | 451.93                  | 0.9920                  |
| maximum transferability measure source domain classifier          | 362.83          | 5.44                     | 451.93                  | 0.9950                  |

As a visualization example, Figure 5 displays noise-added data samples for different values of information theoretic measures.

**Figure 5.** An example of a source domain sample corresponding to different levels of privacy leakage, interpretability measure, and transferability measure.

## 6.2. Office and Caltech256 Datasets

The “Office+Caltech256” dataset has 10 common categories of both Office and Caltech256 datasets. The dataset has four domains: *amazon*, *webcam*, *dslr*, and *caltech256*. This dataset has been widely used [31–34] for evaluating multi-class accuracy performance in a standard domain adaptation setting with a small number of labeled target samples. Following [32], the 4096-dimensional deep-net VGG-FC6 features are extracted from the images. However, for the learning of classifiers, the 4096-dimensional feature vectors are reduced to 100-dimensional feature vectors using principal components computed from the data of *amazon* domain. Thus, corresponding to each image, a 100-dimensional data vector is constructed.

### 6.2.1. Interpretable Parameters

Corresponding to a data vector  $y \in \mathbb{R}^{100}$ , an interpretable parameter vector  $t \in \{0, 1\}^{10}$  is defined such that the  $j$ -th element  $t_j = 1$ , if the  $j$ -th class label is associated to  $y$ ; otherwise,  $t_j = 0$ . That is, interpretable vector  $t$ , in our experimental setting, represents the class-label assigned to data vector  $y$ .

### 6.2.2. Private Data

Here, we assume that the extracted image feature vectors are private, i.e.,  $x_{sr} = y_{sr}$ .

### 6.2.3. Semi-Supervised Transfer Learning Scenario

Similarly to [31–34], the experimental setup is follows:

1. The number of training samples per class in the source domain is 20 for *amazon* and is 8 for the other three domains;
2. The number of labeled samples per class in the target domain is 3 for all the four domains.

#### 6.2.4. Experimental Design

Taking a domain as the source and another domain as the target, 12 different transfer learning experiments are performed on the four domains associated to the “Office+Caltech256” dataset. Each of the 12 experiments is repeated 20 times via creating 20 random train/test splits. In all of the 240 ( $= 12 \times 20$ ) experiments, Algorithm 9 is applied three times with varying values of privacy-loss bound: first with differential privacy parameters as ( $d = 1, \epsilon = 0.01, \delta = 1 \times 10^{-5}$ ), second with differential privacy parameters as ( $d = 1, \epsilon = 0.1, \delta = 1 \times 10^{-5}$ ), and third with differential privacy parameters as ( $d = 1, \epsilon = 1, \delta = 1 \times 10^{-5}$ ). As Algorithm 9 with different values of privacy-loss bound  $\epsilon$  will result in different models, the transfer and multi-task learning models that correspond to the maximum interpretability measure and maximum transferability measure are considered for an evaluation.

#### 6.2.5. Reference Methods

This dataset has been studied previously [31–36] and thus, as a reference, the performances of the following existing methods were considered:

1. ILS (1-NN) [32]: This method learns an Invariant Latent Space (ILS) to reduce the discrepancy between domains and uses Riemannian optimization techniques to match statistical properties between samples projected into the latent space from different domains.
2. CDLS [35]: The Cross-Domain Landmark Selection (CDLS) method derives a domain-invariant feature subspace for heterogeneous domain adaptation.
3. MMDT [34]: The Maximum Margin Domain Transform (MMDT) method adapts max-margin classifiers in a multi-class manner by learning a shared component of the domain shift as captured by the feature transformation.
4. HFA [36]: The Heterogeneous Feature Augmentation (HFA) method learns common latent subspace and a classifier under the max-margin framework.
5. OBTL [33]: The Optimal Bayesian Transfer Learning (OBTL) method employs a Bayesian framework to transfer learning through the modeling of a joint prior probability density function for feature-label distributions of the source and target domains.

#### 6.2.6. Results

Tables 4–15 report the results, and the first two best performances have been marked.

**Table 4.** Accuracy (in %, averaged over 20 experiments) obtained in *amazon*→*caltech256* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <u>82.6</u>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <u>82.6</u>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | <b>83.3</b>  |
| non-private CDLS  | VGG-FC6      | 78.1         |
| non-private MMDT  | VGG-FC6      | 78.7         |
| non-private HFA   | VGG-FC6      | 75.5         |
| non-private OBTL  | SURF         | 41.5         |
| non-private ILS (1-NN)                                    | SURF         | 43.6         |
| non-private CDLS  | SURF         | 35.3         |
| non-private MMDT  | SURF         | 36.4         |
| non-private HFA   | SURF         | 31.0         |

**Table 5.** Accuracy (in %, averaged over 20 experiments) obtained in *amazon*→*dsfr* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <u>88.5</u>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <u>88.7</u>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | 87.7         |
| non-private CDLS  | VGG-FC6      | 86.9         |
| non-private MMDT  | VGG-FC6      | 77.1         |
| non-private HFA   | VGG-FC6      | 87.1         |
| non-private OBTL  | SURF         | 60.2         |
| non-private ILS (1-NN)                                    | SURF         | 49.8         |
| non-private CDLS  | SURF         | 60.4         |
| non-private MMDT  | SURF         | 56.7         |
| non-private HFA   | SURF         | 55.1         |

**Table 6.** Accuracy (in %, averaged over 20 experiments) obtained in *amazon*→*webcam* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | 89.3         |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | 89.3         |
| non-private ILS (1-NN)                                    | VGG-FC6      | <u>90.7</u>  |
| non-private CDLS  | VGG-FC6      | <u>91.2</u>  |
| non-private MMDT  | VGG-FC6      | 82.5         |
| non-private HFA   | VGG-FC6      | 87.9         |
| non-private OBTL  | SURF         | 72.4         |
| non-private ILS (1-NN)                                    | SURF         | 59.7         |
| non-private CDLS  | SURF         | 68.7         |
| non-private MMDT  | SURF         | 64.6         |
| non-private HFA   | SURF         | 57.4         |

**Table 7.** Accuracy (in %, averaged over 20 experiments) obtained in *caltech256*→*amazon* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <u>92.6</u>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <u>92.6</u>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | <u>89.7</u>  |
| non-private CDLS  | VGG-FC6      | 88.0         |
| non-private MMDT  | VGG-FC6      | 85.9         |
| non-private HFA   | VGG-FC6      | 86.2         |
| non-private OBTL  | SURF         | 54.8         |
| non-private ILS (1-NN)                                    | SURF         | 55.1         |
| non-private CDLS  | SURF         | 50.9         |
| non-private MMDT  | SURF         | 49.4         |
| non-private HFA   | SURF         | 43.8         |

Finally, Table 16 summarizes the overall performance of the top four methods. As observed in Table 16, the maximum transferability-measure model remains as best performing in the maximum number of experiments. The most remarkable result observed is that the proposed methodology, despite being privacy-preserving, ensuring the differential privacy-loss bound to be less than equal to 1 and not requiring access to source data samples, performs better than even the non-private methods.

**Table 8.** Accuracy (in %, averaged over 20 experiments) obtained in *caltech256*→*dslr* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <b>89.1</b>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <b>89.1</b>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | 86.9         |
| non-private CDLS  | VGG-FC6      | 86.3         |
| non-private MMDT  | VGG-FC6      | 77.9         |
| non-private HFA   | VGG-FC6      | 87.0         |
| non-private OBTL  | SURF         | 61.5         |
| non-private ILS (1-NN)                                    | SURF         | 56.2         |
| non-private CDLS  | SURF         | 59.8         |
| non-private MMDT  | SURF         | 56.5         |
| non-private HFA   | SURF         | 55.6         |

**Table 9.** Accuracy (in %, averaged over 20 experiments) obtained in *caltech256*→*webcam* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | 87.8         |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | 87.7         |
| non-private ILS (1-NN)                                    | VGG-FC6      | <b>91.4</b>  |
| non-private CDLS  | VGG-FC6      | <b>89.7</b>  |
| non-private MMDT  | VGG-FC6      | 82.8         |
| non-private HFA   | VGG-FC6      | 86.0         |
| non-private OBTL  | SURF         | 71.1         |
| non-private ILS (1-NN)                                    | SURF         | 62.9         |
| non-private CDLS  | SURF         | 66.3         |
| non-private MMDT  | SURF         | 63.8         |
| non-private HFA   | SURF         | 58.1         |

**Table 10.** Accuracy (in %, averaged over 20 experiments) obtained in *dslr*→*amazon* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <b>91.9</b>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <b>91.9</b>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | <b>88.7</b>  |
| non-private CDLS  | VGG-FC6      | 88.1         |
| non-private MMDT  | VGG-FC6      | 83.6         |
| non-private HFA   | VGG-FC6      | 85.9         |
| non-private OBTL  | SURF         | 54.4         |
| non-private ILS (1-NN)                                    | SURF         | 55.0         |
| non-private CDLS  | SURF         | 50.7         |
| non-private MMDT  | SURF         | 46.9         |
| non-private HFA   | SURF         | 42.9         |

### 6.3. An Application Example: Mental Stress Detection

The mental stress detection problem is considered as an application example of the proposed privacy-preserving interpretable and transferable learning approach. The dataset from [17], consisting of heart rate interval measurements of different subjects, is considered for the study of an individual stress detection problem. In [17], a membership-mappings-based interpretable deep model was applied for an estimation of stress score; however, the current study deals with application of the proposed privacy-preserving interpretable and transferable deep learning method to solve the stress classification problem. The problem

is concerned with the detection of stress in an individual based on the analysis of recorded sequence of R-R intervals,  $\{RR^i\}_i$ . The R-R data vector at  $i$ -th time-index,  $y^i$ , is defined as

$$y^i = [RR^i \ RR^{i-1} \ \dots \ RR^{i-d}]^T. \quad (119)$$

That is, the current interval and history of previous  $d$  intervals constitute the data vector. Assuming an average heartbeat of 72 beats per minute,  $d$  is chosen as equal to  $72 \times 3 = 216$  so that the R-R data vector consists of on average 3-minute-long R-R intervals sequences. A dataset, say  $\{y^i\}_i$ , is built via (1) preprocessing the R-R interval sequence  $\{RR^i\}_i$  with an impulse rejection filter [37] for artifacts detection and (2) excluding the R-R data vectors containing artifacts from the dataset. The dataset contains the stress score on a scale from 0 to 100. A label of either “no-stress” or “under-stress” is assigned to each  $y^i$  based on the stress score. Thus, we have a binary classification problem.

### 6.3.1. Interpretable Parameters

Corresponding to a R-R data vector, there exists the set of interpretable parameters: *mental demand*, *physical demand*, *temporal demand*, *own performance*, *effort*, and *frustration*. These are the six components of stress acquired using the NASA Task Load Index [38]. The NASA Task Load Index provides subjective assessment of stress where an individual provides a rating on the scale from 0 to 100 for each of the six components of stress (mental demand, physical demand, temporal demand, own performance, effort, and frustration). Thus, corresponding to each 217-dimensional R-R data vector, there exists a six-dimensional interpretable parameters vector acquired using the NASA Task Load Index.

**Table 11.** Accuracy (in %, averaged over 20 experiments) obtained in *dslr*→*caltech256* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <b>82.9</b>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <b>82.9</b>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | <u>81.4</u>  |
| non-private CDLS  | VGG-FC6      | 77.9         |
| non-private MMDT  | VGG-FC6      | 71.8         |
| non-private HFA   | VGG-FC6      | 74.8         |
| non-private OBTL  | SURF         | 40.3         |
| non-private ILS (1-NN)                                    | SURF         | 41.0         |
| non-private CDLS  | SURF         | 34.9         |
| non-private MMDT  | SURF         | 34.1         |
| non-private HFA   | SURF         | 30.9         |

**Table 12.** Accuracy (in %, averaged over 20 experiments) obtained in *dslr*→*webcam* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | 88.9         |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | 89.0         |
| non-private ILS (1-NN)                                    | VGG-FC6      | <b>95.5</b>  |
| non-private CDLS  | VGG-FC6      | <u>90.7</u>  |
| non-private MMDT  | VGG-FC6      | 86.1         |
| non-private HFA   | VGG-FC6      | 86.9         |
| non-private OBTL  | SURF         | 83.2         |
| non-private ILS (1-NN)                                    | SURF         | 80.1         |
| non-private CDLS  | SURF         | 68.5         |
| non-private MMDT  | SURF         | 74.1         |
| non-private HFA   | SURF         | 60.5         |



**Table 13.** Accuracy (in %, averaged over 20 experiments) obtained in *webcam*→*amazon* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <u>92.3</u>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <u>92.3</u>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | <u>88.8</u>  |
| non-private CDLS  | VGG-FC6      | 87.4         |
| non-private MMDT  | VGG-FC6      | 84.7         |
| non-private HFA   | VGG-FC6      | 85.1         |
| non-private OBTL  | SURF         | 55.0         |
| non-private ILS (1-NN)                                    | SURF         | 54.3         |
| non-private CDLS  | SURF         | 51.8         |
| non-private MMDT  | SURF         | 47.7         |
| non-private HFA   | SURF         | 56.5         |

**Table 14.** Accuracy (in %, averaged over 20 experiments) obtained in *webcam*→*caltech256* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | <u>81.4</u>  |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | <u>81.4</u>  |
| non-private ILS (1-NN)                                    | VGG-FC6      | <u>82.8</u>  |
| non-private CDLS  | VGG-FC6      | 78.2         |
| non-private MMDT  | VGG-FC6      | 73.6         |
| non-private HFA   | VGG-FC6      | 74.4         |
| non-private OBTL  | SURF         | 37.4         |
| non-private ILS (1-NN)                                    | SURF         | 38.6         |
| non-private CDLS  | SURF         | 33.5         |
| non-private MMDT  | SURF         | 32.2         |
| non-private HFA   | SURF         | 29.0         |

**Table 15.** Accuracy (in %, averaged over 20 experiments) obtained in *webcam*→*dslr* semi-supervised transfer learning experiments. The first and second best performances have been marked.

| Method  | Feature Type | Accuracy (%) |
|---|--------------|--------------|
| privacy-preserving maximum interpretability-measure model | VGG-FC6      | 90.8         |
| privacy-preserving maximum transferability-measure model  | VGG-FC6      | 90.2         |
| non-private ILS (1-NN)                                    | VGG-FC6      | <u>94.5</u>  |
| non-private CDLS  | VGG-FC6      | 88.5         |
| non-private MMDT  | VGG-FC6      | 85.1         |
| non-private HFA   | VGG-FC6      | 87.3         |
| non-private OBTL  | SURF         | 75.0         |
| non-private ILS (1-NN)                                    | SURF         | 70.8         |
| non-private CDLS  | SURF         | 60.7         |
| non-private MMDT  | SURF         | 67.0         |
| non-private HFA   | SURF         | 56.5         |

**Table 16.** Comparison of the methods on “Office+Caltech256” dataset.

| Method  | Number of Experiments<br>in Which Method<br>Performed Best |
|---|--|
| privacy-preserving maximum transferability-measure model  | 6  |
| privacy-preserving maximum interpretability-measure model | 5  |
| non-private ILS (1-NN)                                    | 5  |
| non-private CDLS  | 1  |

### 6.3.2. Private Data

Here, we assume that heart rate values are private. As instantaneous heart rate is given as  $HR^i = 60/RR^i$ ; thus, information about private data is directly contained in the R-R data vectors.

### 6.3.3. Semi-Supervised Transfer Learning Scenario

Out of the total subjects, a randomly chosen subject's data serve as the source domain data. Considering every other subject's data as the target domain data, the transfer learning experiment is performed independently on each target subject where 50% of the target subject's samples are labeled, and the remaining unlabeled target samples also serve as test data for evaluating the classification performance. However, only the target subjects, with data containing both the classes and at least 60 samples, were considered for experimentation. There are in total 48 such target subjects.

### 6.3.4. Experimental Design

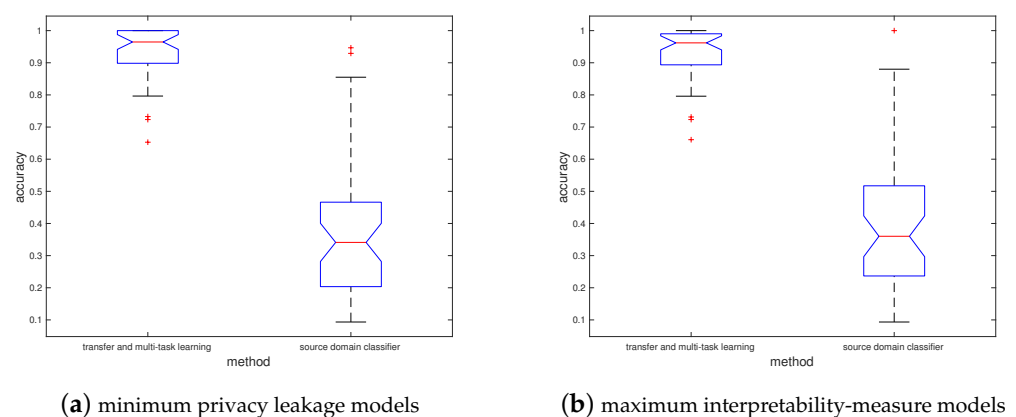
Algorithm 9 is applied with  $d = 1$ ,  $\epsilon \in \{0.1, 0.5, 1, 2, 5, 8, 20, 50, 100, \infty\}$ , and  $\delta = 1 \times 10^{-5}$ . Each of the 48 experiments involves 10 different privacy-preserving semi-supervised transfer learning scenarios with privacy-loss bound values as  $\epsilon = 0.1$ ,  $\epsilon = 0.5$ ,  $\epsilon = 1$ ,  $\epsilon = 2$ ,  $\epsilon = 5$ ,  $\epsilon = 8$ ,  $\epsilon = 20$ ,  $\epsilon = 50$ ,  $\epsilon = 100$ , and  $\epsilon = \infty$ . The following two requirements are associated to this application example:

1. The private source domain data must be protected while transferring knowledge from source to target domain; and
2. The interpretability of the source domain model should be high.

In view of the aforementioned requirements, the models that correspond to the minimum privacy leakage and maximum interpretability measure amongst all the models obtained corresponding to 10 different choices of differential privacy-loss bound  $\epsilon$  are considered for detecting stress.

### 6.3.5. Results

Figure 6 summarizes the experimental results where accuracies obtained by both minimum privacy-leakage models and maximum interpretability-measure models have been displayed as box plots.



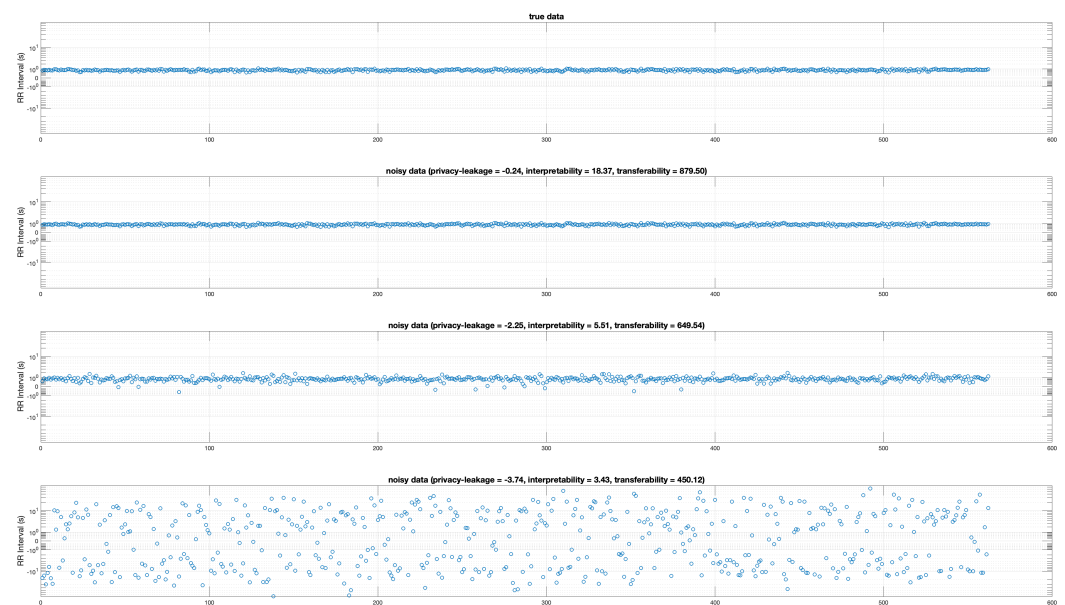
**Figure 6.** The box plots of accuracies obtained in detecting mental stress on 48 different subjects.

It is observed in Figure 6 that the transfer and multi-task learning improves considerably the performance of source domain classifier. Table 17 reports the median values (of privacy leakage, interpretability measure, transferability measure, and classification accuracy) obtained in the experiments on 48 different subjects. The robust performance of transfer and multi-task learning scenario is further observed in Table 17.

**Table 17.** Results (median values) obtained in stress detection experiments on a dataset consisting of heart rate interval measurements.

| Method  | Privacy Leakage | Interpretability Measure | Transferability Measure | Classification Accuracy |
|---|-----------------|--------------------------|-------------------------|-------------------------|
| minimum privacy leakage transfer and multi-task learning          | −3.74           | 3.47                     | 291.84                  | 0.9647                  |
| minimum privacy leakage source domain classifier                  | −3.74           | 3.47                     | 291.84                  | 0.3411                  |
| maximum interpretability measure transfer and multi-task learning | 0.43            | 23.92                    | 773.36                  | 0.9619                  |
| maximum interpretability measure source domain classifier         | 0.43            | 23.92                    | 773.36                  | 0.3602                  |

As a visualization example, Figure 7 displays the noise-added source domain heart rate interval data for different values of information theoretic measures.

**Figure 7.** A display of source domain R-R interval data corresponding to different levels of privacy leakage, interpretability measure, and transferability measure.

## 7. Concluding Remarks

The paper has introduced information theoretic measures for privacy leakage, interpretability, and transferability to study the trade-offs. This is the first study to develop an information theory-based unified approach to privacy-preserving interpretable and transferable learning. The experiments have verified that the proposed measures (for privacy leakage, interpretability, and transferability) can be used to study the trade-off curves (between privacy leakage, interpretability measure, and transferability measure) and thus to optimize the models for the given application requirements such as the requirement of minimum privacy leakage, the requirement of maximum interpretability measure, and the requirement of maximum transferability measure. The experimental results on the MNIST dataset showed that the transfer and multi-task learning scenario improved remarkable the accuracy from 0.1760 to 0.9510 while ensuring the minimum privacy leakage. The experiments on Office and Caltech256 datasets indicated that the proposed methodology, despite ensuring differential privacy-loss bound to be less than equal to 1 and not requiring an access to source data samples, performed better than even existing non-private methods in six out of 12 transfer learning experiments. The stress detection experiments on a real-world biomedical data led to the observation that the transfer and multi-task learning scenario

improved the accuracy from 0.3411 to 0.9647 (while ensuring the minimum privacy leakage) and from 0.3602 to 0.9619 (while ensuring the maximum interpretability measure). The considered unified approach to privacy-preserving interpretable and transferable learning involves membership-mappings-based conditionally deep autoencoders, albeit other data representation learning models could be explored. The future work includes the following:

- Although the text has not focused on federated learning, the transfer learning approach could be easily extended to the multi-party system and the transferability measure could be calculated for any pair of parties.
- Also, the explainability of the conditionally deep autoencoders follows, similar to in [17], via estimating interpretable parameters from non-interpretable data feature vectors using variational membership-mapping Bayesian model.
- Furthermore, the variational membership-mapping Bayesian model quantifies uncertainties on the estimation of parameters (of interest), which is also important for a user's trust on the model.

**Author Contributions:** Conceptualization, M.K.; methodology, M.K.; writing—original draft preparation, M.K.; writing—review and editing, L.F.; project administration, B.F.; funding acquisition, B.A.M. and L.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research reported in this paper has been supported by the Austrian Research Promotion Agency (FFG) COMET-Modul S3AI (Security and Safety for Shared Artificial Intelligence); FFG Sub-Project PETAI (Privacy Secured Explainable and Transferable AI for Healthcare Systems); FFG Grant SMiLe (Secure Machine Learning Applications with Homomorphically Encrypted Data); FFG Grant PRIMAL (Privacy-Preserving Machine Learning for Industrial Applications); and the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry for Digital and Economic Affairs, and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE [(FFG grant no. 892418)] part of the FFG COMET Competence Centers for Excellent Technologies Programme.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

TAI    Trustworthy Artificial Intelligence

## References

1. High-Level Expert Group on AI. *Ethics Guidelines for Trustworthy AI*; Report; European Commission: Brussels, Belgium, 2019.
2. Floridi, L. Establishing the rules for building trustworthy AI. *Nat. Mach. Intell.* **2019**, *1*, 261–262. [CrossRef]
3. Floridi, L.; Cows, J. A Unified Framework of Five Principles for AI in Society. *Harv. Data Sci. Rev.* **2019**, *1*. [CrossRef]
4. Floridi, L.; Cows, J.; Beltrametti, M.; Chatila, R.; Chazerand, P.; Dignum, V.; Luetge, C.; Madelin, R.; Pagallo, U.; Rossi, F.; et al. AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. *Minds Mach.* **2018**, *28*, 689–707. [CrossRef]
5. Mcknight, D.H.; Carter, M.; Thatcher, J.B.; Clay, P.F. Trust in a Specific Technology: An Investigation of Its Components and Measures. *ACM Trans. Manag. Inf. Syst.* **2011**, *2*, 1–25. [CrossRef]
6. Thiebes, S.; Lins, S.; Sunyaev, A. Trustworthy artificial intelligence. *Electron. Mark.* **2020**, *31*, 447–464. [CrossRef]
7. Future of Life Institute. Asilomar AI Principles. 2017. Available online: <https://futureoflife.org/ai-principles/> (accessed on 19 September 2023).
8. Université de Montréal. Montreal Declaration for a Responsible Development of AI. 2017. Available online: <https://www.montrealdeclaration-responsibleai.com/the-declaration/> (accessed on 19 September 2023).
9. UK House of Lords. AI in the UK: Ready, Willing and Able? 2017. Available online: <https://publications.parliament.uk/pa/ld201719/ldselect/ldai/100/10002.htm> (accessed on 19 September 2023).

10. OECD. OECD Principles on AI. 2019. Available online: <https://www.oecd.org/going-digital/ai/principles/> (accessed on 19 September 2023).
11. Chinese National Governance Committee for the New Generation Artificial Intelligence. Governance Principles for the New Generation Artificial Intelligence—Developing Responsible Artificial Intelligence. 2019. Available online: <https://www.chinadaily.com.cn/a/201906/17/WS5d07486ba3103dbf14328ab7.html> (accessed on 19 September 2023).
12. Vought, R.T. Guidance for Regulation of Artificial Intelligence Applications. 2020. Available online: <https://www.whitehouse.gov/wp-content/uploads/2020/01/Draft-OMB-Memo-on-Regulation-of-AI-1-7-19.pdf> (accessed on 19 September 2023).
13. Hagendorff, T. The Ethics of AI Ethics: An Evaluation of Guidelines. *Minds Mach.* **2020**, *30*, 99–120. [CrossRef]
14. Kumar, M.; Moser, B.; Fischer, L.; Freudenthaler, B. Membership-Mappings for Data Representation Learning: Measure Theoretic Conceptualization. In *Proceedings of the Database and Expert Systems Applications—DEXA 2021 Workshops*; Kotsis, G., Tjoa, A.M., Khalil, I., Moser, B., Mashkoor, A., Sametinger, J., Fensel, A., Martinez-Gil, J., Fischer, L., Czech, G., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 127–137.
15. Kumar, M.; Moser, B.; Fischer, L.; Freudenthaler, B. Membership-Mappings for Data Representation Learning: A Bregman Divergence Based Conditionally Deep Autoencoder. In *Proceedings of the Database and Expert Systems Applications—DEXA 2021 Workshops*; Kotsis, G., Tjoa, A.M., Khalil, I., Moser, B., Mashkoor, A., Sametinger, J., Fensel, A., Martinez-Gil, J., Fischer, L., Czech, G., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 138–147.
16. Kumar, M.; Freudenthaler, B. Fuzzy Membership Functional Analysis for Nonparametric Deep Models of Image Features. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 3345–3359. [CrossRef]
17. Kumar, M.; Zhang, W.; Weippert, M.; Freudenthaler, B. An Explainable Fuzzy Theoretic Nonparametric Deep Model for Stress Assessment Using Heartbeat Intervals Analysis. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3873–3886. [CrossRef]
18. Kumar, M.; Singh, S.; Freudenthaler, B. Gaussian fuzzy theoretic analysis for variational learning of nested compositions. *Int. J. Approx. Reason.* **2021**, *131*, 1–29. [CrossRef]
19. Zhang, W.; Kumar, M.; Ding, W.; Li, X.; Yu, J. Variational learning of deep fuzzy theoretic nonparametric model. *Neurocomputing* **2022**, *506*, 128–145. [CrossRef]
20. Kumar, M.; Zhang, W.; Fischer, L.; Freudenthaler, B. Membership-Mappings for Practical Secure Distributed Deep Learning. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 2617–2631. [CrossRef]
21. Zhang, Q.; Yang, J.; Zhang, W.; Kumar, M.; Liu, J.; Liu, J.; Li, X. Deep fuzzy mapping nonparametric model for real-time demand estimation in water distribution systems: A new perspective. *Water Res.* **2023**, *241*, 120145. [CrossRef] [PubMed]
22. Kumar, M. Differentially private transferrable deep learning with membership-mappings. *Adv. Comput. Intell.* **2023**, *3*, 1. [CrossRef]
23. Kumar, M.; Stoll, N.; Stoll, R. Variational Bayes for a Mixed Stochastic/Deterministic Fuzzy Filter. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 787–801. [CrossRef]
24. Kumar, M.; Stoll, N.; Stoll, R.; Thurow, K. A Stochastic Framework for Robust Fuzzy Filtering and Analysis of Signals-Part I. *IEEE Trans. Cybern.* **2016**, *46*, 1118–1131. [CrossRef]
25. Kumar, M.; Stoll, N.; Stoll, R. Stationary Fuzzy Fokker-Planck Learning and Stochastic Fuzzy Filtering. *IEEE Trans. Fuzzy Syst.* **2011**, *19*, 873–889. [CrossRef]
26. Kumar, M.; Neubert, S.; Behrendt, S.; Rieger, A.; Weippert, M.; Stoll, N.; Thurow, K.; Stoll, R. Stress Monitoring Based on Stochastic Fuzzy Analysis of Heartbeat Intervals. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 746–759. [CrossRef]
27. Kumar, M.; Insan, A.; Stoll, N.; Thurow, K.; Stoll, R. Stochastic Fuzzy Modeling for Ear Imaging Based Child Identification. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 1265–1278. [CrossRef]
28. Kumar, M.; Rossbory, M.; Moser, B.A.; Freudenthaler, B. An optimal  $(\epsilon, \delta)$ —Differentially private learning of distributed deep fuzzy models. *Inf. Sci.* **2021**, *546*, 87–120. [CrossRef]
29. Kumar, M.; Brunner, D.; Moser, B.A.; Freudenthaler, B. Variational Optimization of Informational Privacy. In *Proceedings of the Database and Expert Systems Applications*; Kotsis, G., Tjoa, A.M., Khalil, I., Fischer, L., Moser, B., Mashkoor, A., Sametinger, J., Fensel, A., Martinez-Gil, J., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 32–47.
30. Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.J.; Talwar, K. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the ICLR*, Toulon, France, 24–26 April 2017.
31. Hoffman, J.; Rodner, E.; Donahue, J.; Saenko, K.; Darrell, T. Efficient Learning of Domain-invariant Image Representations. *arXiv* **2013**, arXiv:1301.3224.
32. Herath, S.; Harandi, M.; Porikli, F. Learning an Invariant Hilbert Space for Domain Adaptation. In *Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2017.
33. Karbalayghareh, A.; Qian, X.; Dougherty, E.R. Optimal Bayesian Transfer Learning. *IEEE Trans. Signal Process.* **2018**, *66*, 3724–3739. [CrossRef]
34. Hoffman, J.; Rodner, E.; Donahue, J.; Kulis, B.; Saenko, K. Asymmetric and Category Invariant Feature Transformations for Domain Adaptation. *Int. J. Comput. Vis.* **2014**, *109*, 28–41. [CrossRef]
35. Tsai, Y.H.; Yeh, Y.; Wang, Y.F. Learning Cross-Domain Landmarks for Heterogeneous Domain Adaptation. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 5081–5090.
36. Li, W.; Duan, L.; Xu, D.; Tsang, I.W. Learning with Augmented Features for Supervised and Semi-Supervised Heterogeneous Domain Adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1134–1148. [CrossRef] [PubMed]

37. McNames, J.; Thong, T.; Aboy, M. Impulse rejection filter for artifact removal in spectral analysis of biomedical signals. In Proceedings of the The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Francisco, CA, USA, 1–5 September 2004; Volume 1, pp. 145–148. [[CrossRef](#)]
38. Hart, S.G.; Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Hum. Ment. Workload*. **1988**, *1*, 139–183.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.