Author/s:
Wu, X;Manton, JH;Aickelin, U;Zhu, J

Title:
A Bayesian approach to (online) transfer learning: Theory and algorithms

Date:
2023-11

Citation:
Wu, X., Manton, J. H., Aickelin, U. & Zhu, J. (2023). A Bayesian approach to (online) transfer learning: Theory and algorithms. Artificial Intelligence, 324, pp.103991-103991. https://doi.org/10.1016/j.artint.2023.103991.

Persistent Link:
https://hdl.handle.net/11343/336987

Artificial Intelligence 324 (2023) 103991

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

A Bayesian approach to (online) transfer learning: Theory and algorithms ☆

Xuetong Wu [a,*], Jonathan H. Manton [a], Uwe Aickelin [b], Jingge Zhu [a]

[a] Department of Electrical and Electronic Engineering, University of Melbourne, Grattan Street, Parkville, 3010, VIC, Australia
[b] School of Computing and Information Systems, University of Melbourne, Grattan Street, Parkville, 3010, VIC, Australia

ARTICLE INFO

Article history:
Received 3 October 2021
Received in revised form 4 March 2023
Accepted 4 August 2023
Available online 11 August 2023

Keywords:
Transfer learning
Conditional mutual information
Prior information
Negative transfer

ABSTRACT

Transfer learning is a machine learning paradigm where knowledge from one problem is utilized to solve a new but related problem. While conceivable that knowledge from one task could help solve a related task, if not executed properly, transfer learning algorithms can impair the learning performance instead of improving it – commonly known as *negative transfer*. In this paper, we use a parametric statistical model to study transfer learning from a Bayesian perspective. Specifically, we study three variants of transfer learning problems, instantaneous, online, and time-variant transfer learning. We define an appropriate objective function for each problem and provide either exact expressions or upper bounds on the learning performance using information-theoretic quantities, which allow simple and explicit characterizations when the sample size becomes large. Furthermore, examples show that the derived bounds are accurate even for small sample sizes. The obtained bounds give valuable insights into the effect of prior knowledge on transfer learning, at least with respect to our Bayesian formulation of the transfer learning problem. In particular, we formally characterize the conditions under which negative transfer occurs. Lastly, we devise several (online) transfer learning algorithms that are amenable to practical implementations, some of which do not require the parametric assumption. We demonstrate the effectiveness of our algorithms with real data sets, focusing primarily on when the source and target data have strong similarities.

1. Introduction

Machine learning has been widely studied and used in many real-world applications. One crucial assumption for traditional machine learning algorithms is that the training and target data are drawn from the same distribution. However, this assumption may not always hold in practice. This may be because training and testing data are time-varying or because it is difficult to collect data from testing distributions due to annotation expenses or privacy considerations. To tackle the distribution mismatch issues, learning algorithms need to be developed that can "transfer" knowledge across different domains.

☆ This work is an extended version of the preliminary work [67] appeared in ISIT2021 conference. This paper extends the results of online transfer learning to instantaneous and time-variant transfer learning. We then propose several algorithms inspired by the mixture strategy and compare them with some traditional benchmarks.

* Corresponding author.
E-mail addresses: xuetongw1@student.unimelb.edu.au (X. Wu), jmanton@unimelb.edu.au (J.H. Manton), uwe.aickelin@unimelb.edu.au (U. Aickelin), jingge.zhu@unimelb.edu.au (J. Zhu).

https://doi.org/10.1016/j.artint.2023.103991

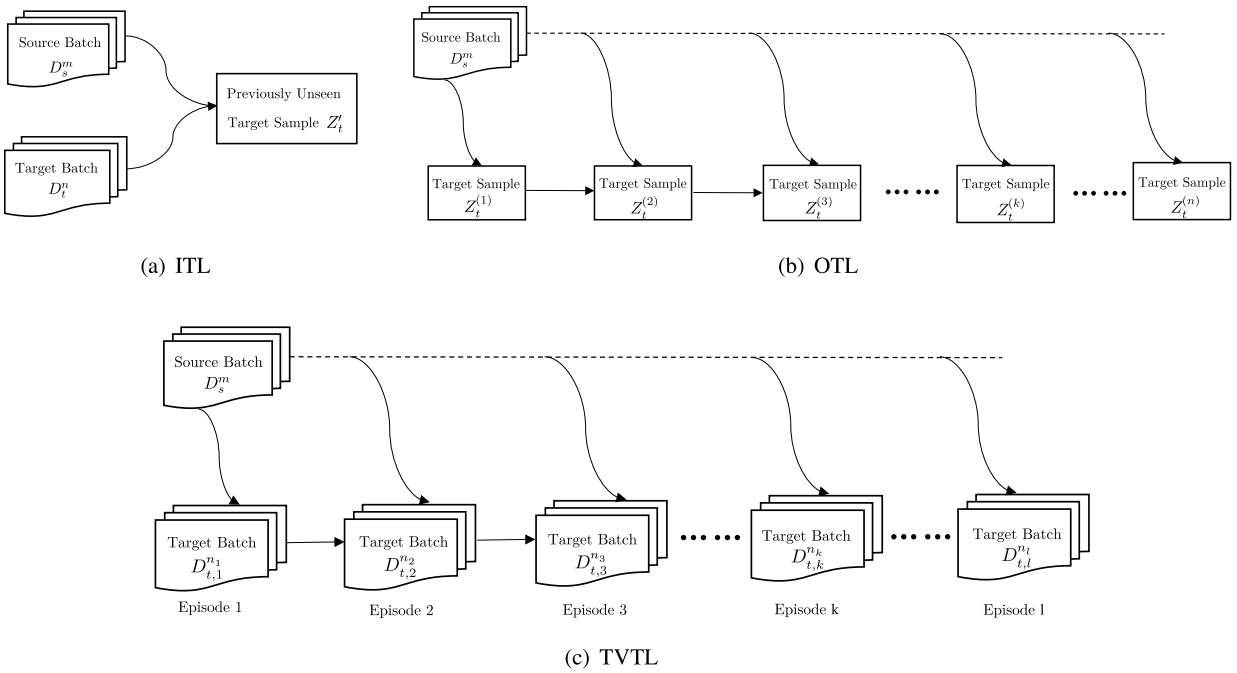


Fig. 1. We will investigate various transfer learning frameworks in this work where (a) describes the instantaneous transfer learning: given both batch target and source data, the task is to predict for a single previously unseen target data. (b) shows the learning regime for online transfer learning where batch source data is available and the target data will arrive sequentially from a fixed distribution, the task is to make sequential predictions for incoming target data. (c) illustrates the process for the time-variant transfer learning where batch source data is available and the target data will arrive sequentially from a possibly time-variant distribution, the task is to make sequential predictions for incoming target data for each episode.

Such transfer learning algorithms leverage knowledge from one or more *source* domains to resolve the problem (or improve the performance) in a related *target* domain. This problem has been brought to the fore due to the rapid growth of different types of data and the rise of complicated learning models such as pre-trained neural networks [13,73] and has been widely employed in natural language processing [50], computer vision [61] and recommender systems [45].

Transfer learning (also known as domain adaptation) problems are currently widely investigated with different setups. Most existing transfer learning methods focus on offline settings where both batch target and source data are available in the training phase [44,63,83,72]. In the testing phase, the performance is evaluated with previously unseen target data. We refer to this setup as instantaneous transfer learning (ITL); see Fig. 1(a). To accommodate applications where data arrive sequentially (e.g., predicting stock market prices), we also study online transfer learning (OTL), first proposed by [81]. The framework of OTL is illustrated in Fig. 1(b), where the decisions are sequentially made with the aid of source and historical target data. The online framework has been extended to many other problems such as multi-source transfers [65,28], multi-task problems [23] and iterative domain adaptation [6]. The third setup we consider is time-variant transfer learning (TVTL) which is an extension of OTL where the target data are drawn sequentially from a possibly time-variant distribution (see Fig. 1(c)). This assumption is relevant for some practical problems such as climate change and geographical process detection, where the data distribution may vary from time to time. Similar problems have been studied in [35,58].

In this work, we extend the results in [67] and propose a general framework for instantaneous, online, and time-variant transfer learning problems that is suitable for a very general transfer learning setup. Specifically, we formulate the transfer learning problems under the assumption that the source and target data distributions are parameterized by some unknown but fixed parameters. Then we define the corresponding evaluation criterion for each case and propose an information-theoretic based algorithm for learning in the target domain. In a nutshell, the upper bounds of the learning performance are characterized by the conditional mutual information (CMI) between the model parameters and the testing samples conditioned on training samples, and the asymptotic approximation w.r.t. the sample sizes is derived if the prior distribution is proper. Practically, the bound can also be applied to the scenario where only limited source and target data are available.

1.1. Related work

Transfer learning We refer to three excellent surveys on transfer learning, [44], [63] and [83], which formally define the problem of transfer learning and provide a comprehensive overview of transfer learning methodologies. We mainly focus on the condition that both source and target data have the same feature and label spaces, known as *homogeneous transfer learning*. Most technologies to conquer the distribution shifting are roughly categorized into instance-based, feature-based, parameter-based, and deep learning-based methods. Instance-based methods identify source samples that bear a likeness

to target samples by importance re-weighting [20,11]. Feature-based methods map both the source and target data to a new latent space where the discrepancy of their (empirical) distribution embeddings (e.g., kernel embeddings) is small under some metric and then construct learning models with the new representations [45,37,77]. The idea of parameter-based methods is to construct a model using the source data initially and then learned model parameters are shared with the target domain as a pre-obtained model for further fine-tuning or regularization [14,30]. Deep learning-based methods use deep neural networks to either learn new representations in both source and target domains for efficient knowledge transfer [36,56,38] or pre-train a model from the source domain that generalizes well in the target domain [13,73], which takes advantage of both feature-based and parameter-based methods. However, most of these methods focus on empirical verification of source samples' effects instead of rigorous theoretical analysis of their algorithms. For example, we lack the understanding of how the source data explicitly affects the generalization error in the target domain, and it is not clear when the negative transfer happens given a specific algorithm [49]. Moreover, there is no unified framework for analyzing this type of problem. Current theoretical analyses for transfer learning focus on either the co-variate shift or conditional distribution alignment [48]. To this end, various metrics, such as $\mathcal{H}\Delta\mathcal{H}$ divergence [4,80], maximum mean discrepancy [45,37,77], KL divergence [66] and density ratios of the joint distribution between the source and target [62], are developed to measure the similarity between the source and target domains. It is widely recognized that minimizing such divergence is more likely to bring about successful knowledge transfer, and prior knowledge over the source and target domains effectively improves the prediction, as demonstrated in many papers [3,7,41]. However, to the best of our knowledge, there is no work that theoretically defines this prior knowledge and its effect on the learning guarantees.

Online transfer learning In contrast to conventional batch settings, online transfer learning has attracted more attention in recent research, where the target data may arrive sequentially. In this particular setting, key questions include the following. Can the source data can help reduce the prediction loss for the target? Under what conditions will the source data be helpful? How does the source data interfere with the prediction of target data, and how does the learning performance vary quantitatively? To answer these questions formally, Zhao et al. [81] first proposed the OTL framework for binary classification with linear models, while the learning metrics and loss functions are limited, i.e., performance is evaluated using very specific metrics such as the number of mistakes. Such a learning framework, in general, does not exploit the structures (or distributions) of the data or model parameters. Wu et al. [65] extended the OTL framework in [81] and derive a similar algorithm for multiple source domains. Some deep neural networks based methods such as [74,25,39,35] consider the time-evolving target domains and empirically show the usefulness of their proposed methods. Yet, the theoretical guarantees on when or whether negative transfer happens are less investigated. Kumagai and Iwata [29] studies the time-variant domain adaptation problem using variational Bayesian inference, which is close to our framework. However, the latest learned hypothesis is restricted to normal distributions with a linear combination of previously learned hypotheses, which may not always be the case in real problems. Wu and He [64] proposed $\mathcal{C}$-divergence to measure label-informed domain discrepancy between the source along with the previous target domains and the current target domain and provided a theoretical bound on the number of the mistakes. Among the many other papers on OTL algorithms, relatively few focus on rigorous theoretical analysis of the learning performance.

Negative transfer The critical points for successful transfer learning are knowing how to transfer, what to transfer and when to transfer [72]. Researchers have focused more on how and what to transfer but have paid less attention to when to transfer. Characterizing and determining when to transfer is of great importance since the source data is not always beneficial. If the source is very different from the target or if we do not execute the learning procedures properly, the source data will instead impair the performance on the target domain [49,62]. As the effectiveness of transfer learning is not always guaranteed, there is a need to develop a robust methodology to overcome the *negative transfer* problem (see [79] for an overview). Roughly speaking, the negative transfer can result from poor source and target data quality, the "distance" between the source and target domains, or a difference between the source and target learning objectives. The idea of avoiding negative transfer is to minimize the aforementioned *domain divergence* between the source and target in terms of the underlying distributions. Even though the effectiveness of source data depends on such divergences, the negative transfer is inevitable in many empirical experiments. Notably, few papers formally study the problem of negative transfer, which is a crucial question in transfer learning.

Information-theoretic framework and universal prediction The information-theoretic framework has been established and studied in many online and reinforcement learning problems (see [32,31,75,55,40,51] for references). One advantage of this framework is that information-theoretic tools are powerful in studying asymptotic behaviors as well as deriving learning performance bounds for various statistical problems. Additionally, information-theoretic quantities such as mutual information and KL divergence (relative entropy) give natural interpretations for the learning bounds. This paper establishes learning bounds for various transfer learning setups using the information-theoretic concept of universal prediction [40]. Here, "universal" means that the predictor does not depend on the unknown underlying distribution and performs essentially as well as if the distribution was known in advance.

Previous studies, such as [15,40,12], mainly focused on situations where data is drawn independently and identically from a single parametric distribution, which is similar to traditional online learning problems. In contrast, we extended their work by deriving excess risk bounds based on conditional mutual information for various transfer learning scenarios

with batch source data introduced, which may come from a different distribution than the target data of interest. However, the bounds obtained through the conditional mutual information cannot provide more quantitative insights for analyzing the regret. To this end, the previous works such as [9,10,82] provided an asymptotic analysis for the conditional mutual information under the conventional online learning or semi-supervised learning problems, where the regret approximation is associated with the prior distribution over the distribution parameters. Building upon these works, we further derive the asymptotic approximation of regrets in different transfer learning scenarios and draw corresponding conclusions for various transfer setups. From our analysis, we identify the usefulness of source data (i.e., whether it leads to positive or negative transfer) depending on the prior knowledge of the source and target domains. Moreover, our approach differs from the related information-theoretic-based analysis for online learning setup with the side information [12,53], where the source data is sequentially provided together with the target data and the sample sizes of two domains are restricted to be identical. Instead, we consider a more general transfer learning regime where the source data is initially offered as a batch, and its sample size can differ from the target data.

1.2. Contributions

We briefly summarize the main contributions of our paper as follows.

- We formulate the instantaneous, online and time-variant transfer learning problems under parametric distribution conditions where the source data are sampled from $P_{\theta_s^*}$ and target data are sampled from $P_{\theta_t^*}$ ($P_{\theta_{t,i}^*}$ for TVTL at episode i) from a parametric family $\mathcal{P}_\theta, \theta \in \Lambda$. With the proposed *mixture strategy*, the learning performance for each problem is characterized by the conditional mutual information (CMI) between the distribution parameters and data samples. Furthermore, we quantitatively give an asymptotic estimation of CMI for each problem, which clearly shows how the learning performance depends on the target and source sample sizes. These bounds also give an explicit connection between the learning performance and the prior knowledge over θ_t^* ($\theta_{t,i}^*$ for TVTL) and θ_s^* along with their common parameters.
- Based upon the asymptotic bounds, we define the concept of *proper prior* and show that the improper prior will lead to the negative transfer. We also identify scenarios when the positive transfer will happen, which provides a practical guideline to avoid the negative transfer. To the best of our knowledge, this is the first work that theoretically and quantitatively characterizes negative and positive transfer phenomena.
- To extend our theoretical study to practical implementation, we devise efficient transfer learning algorithms inspired by the mixture strategy. Specifically, the algorithm can be extended to a broader family of models where the model parameter θ_s and θ_t are not necessarily the true data generating distribution, thus extending our results to more general applications. Experimenting on both the synthetic and real data sets, the results empirically confirm the theoretical results and show the usefulness of our proposed algorithms.

This paper is structured as follows. In Section 2, we formally formulate the problems for instantaneous, online, and time-variant transfer learning. The main theoretical results and discussions are presented in Section 3. The proposed algorithms and experiments are presented and discussed in Section 4 along with some practical applications. Section 5 concludes the work and carries out some future works.

2. Problem formulation

This section formally formulated the ITL, OTL, and TVTL problems. We will use the convention that capital letters denote random variables and lower-case letters as their realizations. We define $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$, and denote the support of a probability distribution by $\text{supp}(\cdot)$. We use $f(n) \asymp g(n)$ to signify $f(n)$ grows at the same asymptotic rate as $g(n)$, meaning there exists some positive integer n_0 such that for all $n > n_0$, $c_1 g(n) \leq f(n) \leq c_2 g(n)$ always holds for some positive values c_1 and c_2. We also use $f(n) \lesssim g(n)$ (for convenience, we will also use $f(n) = O(g(n))$ interchangeably) to mean $f(n)$ grows asymptotically no faster than $g(n)$, that is, there exists some integer n_0 such that for all $n > n_0$, $f(n) \leq c_3 g(n)$ always holds for some positive value c_3.

Instantaneous transfer learning Assume the source data $D_s^m = (Z_s^{(1)}, \cdots, Z_s^{(m)}) \in \mathcal{Z}^m$ and the target data $D_t^n = (Z_t^{(1)}, \cdots, Z_t^{(n)}) \in \mathcal{Z}^n$ are given, where each sample takes on a value in $\mathcal{Z}$. Note that $\mathcal{Z}$ is a separable metric space with probability measures assumed to be defined on the Borel subsets of $\mathbb{R}^k$. Note that our framework naturally applies to both the supervised learning problem where $Z = (X, Y)$ is a feature-label pair and the unsupervised learning problem where $Z = X$ only denotes the feature. We will predict a single previously unseen target sample Z_t' using D_s^m and D_t^n with the predictor $b : \mathcal{Z}^m \times \mathcal{Z}^n \to \mathcal{B}$ where $\mathcal{B}$ denotes the set of the predictor. Associated with this predictor b and the actual outcome z_t', we introduce a loss function $\ell : \mathcal{B} \times \mathcal{Z} \to \mathbb{R}$ to evaluate the prediction performance. We will later show how to construct b properly for different loss functions. We make the following assumptions on data distributions.

Assumption 1 *(Parametric Distributions).* We assume that source and target data are generated independently in an i.i.d. fashion. Specifically, the joint distribution of the source and target data $P_{\theta_s^*, \theta_t^*}(D_t^n, D_s^m, Z_t')$ can be written as

$$P_{\theta_s^*,\theta_t^*}(D_t^n, D_s^m, Z_t') = P_{\theta_t^*}(Z_t') \prod_{i=1}^{n} P_{\theta_t^*}(Z_t^{(i)}) \prod_{j=1}^{m} P_{\theta_s^*}(Z_s^{(j)}), \tag{1}$$

where $P_{\theta_t^*}(Z)$ and $P_{\theta_s^*}(Z)$ are two probability density functions in a parametrized family of distributions $\mathcal{P} = \{P_\theta\}_{\theta\in\Lambda}$ with respect to a fixed σ-finite measure $\mu(dz)$. Here Λ is a closed set on $\mathbb{R}^d$ and θ_t^*, θ_s^* are the interior points of Λ.

After observing n target samples and m source samples, assume ℓ is integrable w.r.t. the measure μ given any b, we want to minimize the corresponding *excess risk* defined as

$$\mathcal{R}_I := \mathbb{E}_{\theta_s^*,\theta_t^*}\left[\ell\left(b, Z_t'\right) - \ell(b^*, Z_t')\right], \tag{2}$$

where b is the predictor we made based on the source data D_s^m and target data D_t^n but without the knowledge of θ_s^* and θ_t^*. Under certain loss functions, the predictor b^* is set to be the optimal one that can depend on true target distributions $P_{\theta_t^*}$, which will be specified later and shown to have a unique optimal. For other general loss functions, under suitable continuity conditions, we do have an optimal predictor depending on $P_{\theta_t^*}$, see [21,40] for examples. If not otherwise specified, the notation $\mathbb{E}_{\theta_s,\theta_t}[\cdot]$ (similarly, $\mathbb{E}_{\theta_t}[\cdot]$ and $\mathbb{E}_{\theta_s}[\cdot]$) denotes the expectation taken over all source and target samples that are drawn from P_{θ_s} and P_{θ_t}. We will call this problem setup "*instantaneous transfer learning*" where the subscript of R_I originates. In this model, both target and source data are given in batches in the training phase, and the learned predictor will be applied to a single unseen target sample.

Online transfer learning Assume the source data $D_s^m = (Z_s^{(1)}, \cdots, Z_s^{(m)}) \in \mathcal{Z}^m$ are given in batch while the target data are received sequentially as $Z_t^{(1)}, Z_t^{(2)}, \cdots, Z_t^{(k)}, \cdots$ where each sample takes value in $\mathcal{Z}$. At each time instant k, after having seen $D_t^{k-1} = (Z_t^{(1)}, Z_t^{(2)}, \cdots, Z_t^{(k-1)})$, we predict $Z_t^{(k)}$ using D_s^m and D_t^{k-1} with the predictor $b_k : \mathcal{Z}^m \times \mathcal{Z}^{k-1} \to \mathcal{B}$. Assume the target sequence and source batch are sampled in an i.i.d. way as described in Assumption 1. After observing n target samples, we want to minimize the corresponding *expected regret* defined as

$$\mathcal{R}_O := \mathbb{E}_{\theta_s^*,\theta_t^*}\left[\sum_{k=1}^{n} \ell(b_k, Z_t^{(k)}) - \sum_{k=1}^{n} \ell(b_k^*, Z_t^{(k)})\right], \tag{3}$$

where b_k is the predictor we made based on the source data D_s^m and target data D_t^{k-1} but without the knowledge of θ_s^* and θ_t^*. Similarly, under certain loss functions, the predictor b_k^* is set to be the optimal one at each time k that can depend on true target distributions $P_{\theta_t^*}$. The subscript "O" is short for "Online".

Time-variant transfer learning In the above OTL scenario, we assume the distributions of target samples are time-invariant, which may not always be the case in many real-world applications. In this model, we consider the *time-evolving* target data that the distributions are parametrized by $\theta_{t,i}^*$ where $i \in \mathbb{N}+$ represents the episode of the sequential target distributions. That is, at each i, we will receive n_i target samples sequentially drawn from an unknown but fixed distribution $P_{\theta_{t,i}^*}$. Furthermore, at episode i, we assume that $\theta_{t,i}^*$ shares c_i common parameters with $\theta_{t,i-1}^*$, known as the target common parameters. We also assume that the source parameter θ_s^* shares j_i common parameters with $\theta_{t,i}^*$ and $\theta_{t,i-1}^*$, known as the source sharing parameters. For simplicity, we suppose that the target common parameters and the source-sharing parameters are not overlapped. Denote D_s^m the source dataset and denote $D_{t,i}^{n_i} = (Z_{t,i}^{(k)})_{k=1,2,\cdots,n_i}$ the received target dataset till time n_i at episode i, we predict $Z_{t,i}^{(n_i+1)}$ using the source data D_s^m and target data $D_{t,i-1}^{n_{i-1}}$ in previous episode and $D_{t,i}^{n_i}$ in current episode with the predictor $b_{n_i+1,i} : \mathcal{Z}^m \times \mathcal{Z}^{n_{i-1}} \times \mathcal{Z}^{n_i} \to \mathcal{B}$. We further make the following assumption.

Assumption 2. In time-variant transfer learning, we assume that source and time-variant target data are generated independently in an i.i.d. fashion within each episode. More precisely, the joint distribution of the data sequence till time n_l in episode l can be factorized as,

$$P_{\theta_{t,1}^*,\theta_{t,2}^*,\cdots,\theta_{t,l}^*,\theta_s^*}(D_{t,1}^{n_1}, D_{t,2}^{n_2}, \cdots, D_{t,l}^{n_l}, D_s^m) = \prod_{i=1}^{l}\prod_{k=1}^{n_i} P_{\theta_{t,i}^*}(Z_{t,i}^{(k)}) \prod_{j=1}^{m} P_{\theta_s^*}(Z_s^{(j)}), \tag{4}$$

where $(P_{\theta_{t,i}^*})_{i=1,2,\cdots,l}$ and $P_{\theta_s^*}$ are in a parametrized family of distributions $\mathcal{P} = \{P_\theta\}_{\theta\in\Lambda}$. Here $\Lambda \subseteq \mathbb{R}^d$ is some measurable space, and $(\theta_{t,i}^*)_{i=1,2,\cdots,l}$ and θ_s^* are points in the interior of Λ. To be consistent, we define $n_0 = 0$ and $\theta_{t,0}^*$ is arbitrarily chosen in Λ as a trivial initialization.

Assume the number of target samples n_i at each episode i is known, we are interested in minimizing the expected regret till episode l as:

$$\mathcal{R}_{TV} := \sum_{i=1}^{l} \mathbb{E}_{\theta_s^*, \theta_{t,i}^*, \theta_{t,i-1}^*} \left[\sum_{k=1}^{n_i} \ell(b_{k,i}, Z_{t,i}^{(k)}) - \sum_{k=1}^{n_i} \ell(b_{k,i}^*, Z_{t,i}^{(k)}) \right], \tag{5}$$

where $b_{k,i}$ is chosen based on the source data D_s^m and target data $D_{t,i}^{k-1}$ and $D_{t,i-1}^{n_{i-1}}$ but without the knowledge of θ_s^*, $\theta_{t,i-1}^*$ and $\theta_{t,i}^*$. The predictor $b_{k,i}^*$ is the optimal decision at each time k that can depend on true target distributions $P_{\theta_{t,i}^*}$ and $P_{\theta_{t,i-1}^*}$. The subscript "TV" stands for "Time-Variant".

3. Main results

In this section, we will present our main theoretical results. Under the Assumption 1, the parametric conditions allow us to characterize the excess risk in Equation (2) and the expected regrets in Equation (3), (5) using information-theoretic quantities under the logarithmic loss or any bounded loss functions. To be specific, the CMI captures the performance of both the excess risk and expected regret, and their asymptotic estimations are derived when we have sufficient source and target samples.

3.1. Information-theoretic characterization

Instantaneous transfer learning For the sake of simplicity, we first present our main results under the *logarithmic loss* for the ITL setup, which is formally defined as follows.

Definition 1 *(Logarithmic Loss).* Let the predictor b be a probability distribution over the target sample Z_t'. The logarithmic loss is then defined as

$$\ell(b, Z_t') = -\log b(Z_t'). \tag{6}$$

Under the log loss, the predictor b can be naturally viewed as a conditional distribution Q over the unseen target data given the training data D_s^m and D_t^n. With this interpretation in mind, we define the expected loss on test data as

$$L := -\mathbb{E}_{\theta_t^*, \theta_s^*} \left[\log Q(Z_t' | D_t^n, D_s^m) \right]. \tag{7}$$

From [40], the optimal predictor b^* is given by the underlying target distribution as $b^*(Z_t') = P_{\theta_t^*}(Z_t'|D_t^n) = P_{\theta_t^*}(Z_t')$ under the Assumption 1. Then the excess risk can be expressed as

$$\mathcal{R}_I = \mathbb{E}_{\theta_t^*, \theta_s^*} \left[\ell(b, Z_t') | D_t^n, D_s^m \right] - \mathbb{E}_{\theta_t^*, \theta_s^*} \left[\ell(b^*, Z_t') | D_t^n, D_s^m \right] \tag{8}$$

$$= \mathbb{E}_{\theta_t^*, \theta_s^*} \left[\log \frac{P_{\theta_t^*}(Z_t')}{Q(Z_t' | D_t^n, D_s^m)} \right]. \tag{9}$$

We define Θ_s and Θ_t as random variables over Λ, which can be interpreted as a random guess of θ_s^* and θ_t^*, and we choose some probability distribution ω over Θ_s and Θ_t with respect to Lebesgue measure as our prior knowledge on these parameters. The predictor Q is then defined as

$$Q(Z_t' | D_t^n, D_s^m) = \frac{\int P_{\theta_t}(D_t^n, Z_t') P_{\theta_s}(D_s^m) \omega(\theta_t, \theta_s) d\theta_t d\theta_s}{\int P_{\theta_t}(D_t^n) P_{\theta_s^*}(D_s^m) \omega(\theta_t, \theta_s) d\theta_t d\theta_s} \tag{10}$$

$$= \int P_{\theta_t}(Z_t') Q(\theta_t, \theta_s | D_s^m, D_s^n) d\theta_s d\theta_t. \tag{11}$$

This choice is known as *mixture strategy* [40,68]. We assign a probability distribution ω over Θ_s and Θ_t w.r.t. the Lebesgue measure to represent our prior knowledge, and we update the posterior with the incoming data to approximate the underlying distributions. The Equation (11) gives a natural interpretation of a two-stage prediction method on Z_t'. In the first stage, the joint posterior $Q(\theta_s, \theta_t | D_s, D_t)$ gives the estimation of θ_s and θ_t. In the second stage, the learned θ_t is applied for prediction in terms of the parametric distribution $P_{\theta_t}(Z_t')$. One way to comprehend the mixture strategy is that we encode our prior knowledge over target and source domain distributions in terms of the prior distribution $\omega(\Theta_s, \Theta_t)$, and its induced conditional distribution $\omega(\Theta_t | \Theta_s)$ shows our belief over target parameters given the source parameters, e.g., how close Θ_t and Θ_s are.

Remark 1. Different from many other transfer learning algorithms, the predictive hypothesis is learned via the empirical risk minimization (ERM) algorithm [4,76,47], we learn the distribution parameters and make the prediction from a Bayesian approach. On the one hand, with the mixture strategy, we could encode the prior knowledge of the distribution parameters with the prior distribution ω and estimate the posterior from the data, providing insights on how to incorporate the source

with the prediction in the target domain. Because the ERM algorithm does not take the data distribution into account, thus it is not easy to see the usefulness of the source data. On the other hand, the mixture strategy is optimal under minimax settings [40], and we later show that it can achieve the excess risk with the optimal rate $O(\frac{1}{n})$ under specific priors, while the bounds for ERM algorithm will usually involve the domain divergence term where it does not converge to zero even with sufficient data [48].

The following theorem gives an exact characterization of the excess risk of the predictor b under the logarithmic loss.

Theorem 1 *(Excess Risk with Logarithmic Loss for ITL). Under the logarithmic loss, let the predictor* $b = Q(Z_t'|D_s^m, D_t^n) = \frac{\int P_{\theta_t^*}(D_t^n, Z_t')P_{\theta_s^*}(D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s}{\int P_{\theta_t^*}(D_t^n)P_{\theta_s^*}(D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s}$ *with the prior distribution* ω *described in Equation (11), the excess risk can be written as*

$$\mathcal{R}_I = \mathbb{E}_{\theta_t^*,\theta_s^*}\left[\frac{P_{\theta_t^*}(Z_t')}{Q(Z_t'|D_s^m, D_t^n)}\right] = I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m, D_t^n). \tag{12}$$

All proofs in this paper can be found in Appendices. A similar learning strategy can be used for more general loss functions. Given a general bounded loss function ℓ, we define the predictor b to be

$$b = \operatorname{argmin}_b \mathbb{E}_{Q(Z_t', D_t^n, D_s^m)}\left[\ell(b, Z_t')|D_s^m, D_t^n\right], \tag{13}$$

with the choice of the mixture strategy $Q(Z_t', D_t^n, D_s^m) = \int P_{\theta_t,\theta_s}(Z_t', D_t^n, D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s$ for some prior ω. The optimal predictor is then given by

$$b^* = \operatorname{argmin}_b \mathbb{E}_{P_{\theta_t^*}(D_t^n, D_s^m, Z_t')}\left[\ell(b, Z_t')|D_s^m, D_t^n\right]. \tag{14}$$

We have the following theorem for general bounded loss functions.

Theorem 2 *(Excess Risk with Bounded Loss for ITL). Assume the loss function satisfies* $|\ell(b, z) - \ell(b^*, z)| \le M$ *for any observation* z *and any two predictors* b, b^*. *Then the excess risk induced by* b *and* b^* *in Equation (13) and (14) can be bounded as*

$$\mathcal{R}_I \le M\sqrt{2I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m, D_s^n)}. \tag{15}$$

Theorem 3 *(Excess Risk with Exponentially Concave Loss for ITL). Assume the loss function is* β*-exponentially concave in* b *for any* $z \in \mathcal{Z}$, *namely, that the function* $\exp(-\beta\ell(b, z))$ *is concave in the first argument. Then the excess risk induced by* b *and* b^* *in Equation (13) and (14) can be bounded as*

$$\mathcal{R}_I \le \frac{1}{\beta} I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m, D_s^n). \tag{16}$$

Many loss functions such as can be shown to be exponentially concave. For example, the squared loss $\ell(b, z) = (b - z)^2$ is $1/(8a^2)$-exponentially concave if the absolute value of b, z are no larger than a. The discrete entropy and Renyi entropy are also satisfying the exponentially concave with appropriate scaling. Furthermore, the logarithmic loss $\ell(b, z) = -\log b(z)$ that we used throughout the paper satisfies the 1-exponentially concave. See [1] and [82] for references.

The above theorems imply that under both logarithmic loss and other bounded or exponentially concave loss functions, with a specific prior ω, the excess risk induced by the mixture strategy is captured by the conditional mutual information between the sample Z_t' and Θ_t, Θ_s that are evaluated at θ_t^* and θ_s^* given the source and target data. However, the expressions involving CMI are not very informative in the sense that they do not clearly show the effect of source data in transfer learning. Our analysis in asymptotic estimation will provide insight into the usefulness of source data. We will give the asymptotic estimation of this quantity later.

Online transfer learning Techniques for instantaneous transfer learning can be extended to handle the online transfer learning problem. We first examine the expected regret under the *logarithmic loss*. Assume we have m source samples, at each time k, we may now view the predictor as a conditional probability distribution $b_k(z_t^{(k)}) = Q(z_t^{(k)}|D_t^{k-1}, D_s^m)$ conditioned on both source and target data. Using the same argument as in ITL, the optimal predictor b_k^* is naturally given by the true target distribution over $z_t^{(k)}$ as $b_k^*(z_t^{(k)}) = P_{\theta_t^*}(z_t^{(k)})$. Then the expected regret till time n can be written explicitly as

$$\mathcal{R}_O = \mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{1}{Q(D_t^n|D_s^m)} - \log\frac{1}{P_{\theta_t^*}(D_t^n)}\right]. \tag{17}$$

The effect of source data is reflected in the conditional distribution $Q(D_t^n|D_s^m)$. In particular, we choose the predictor $Q(D_t^n|D_s^m)$ with the mixture strategy as follows.

$$\begin{aligned} Q(D_t^n|D_s^m) &= \frac{\int P_{\theta_t,\theta_s}(D_t^n, D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s}{\int P_{\theta_s}(D_s^m)\omega(\theta_s)d\theta_s} \\ &= \frac{\int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s)P_{\theta_s}(D_s^m)\omega(\theta_s)d\theta_t d\theta_s}{Q(D_s^m)} \\ &= \int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s)d\theta_t \frac{P_{\theta_s}(D_s^m)\omega(\theta_s)}{Q(D_s^m)}d\theta_s \\ &= \int\int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s)d\theta_t Q(\theta_s|D_s^m)d\theta_s, \end{aligned} \tag{18}$$

where $\omega(\theta_s)$ and $\omega(\theta_t|\theta_s)$ are induced by the joint distribution $\omega(\theta_s,\theta_t)$. From Eq (18), the mixture strategy quantitatively explains how transfer learning is implemented via the posterior updates of θ_t sequentially. Intuitively speaking, the posterior $Q(\theta_s|D_s^m)$ firstly gives an estimate of θ_s^* from the source data, then the conditional prior $\omega(\theta_t|\theta_s)$ reflects our belief upon θ_t^* given θ_s estimated from source data. With the choice of $Q(D_t^n|D_s^m)$, the expected regret can be explicitly characterized in the following theorem.

Theorem 4 *(Expected Regret with Logarithmic Loss for OTL). With the mixture strategy $Q(D_t^n|D_s^m)$ in (18), the expected regret in (17) can be written as*

$$\mathcal{R}_O = \mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right] = I(D_t^n;\Theta_t=\theta_t^*,\Theta_s=\theta_s^*|D_s^m), \tag{19}$$

where $I(D_t^n;\Theta_t=\theta_t^*,\Theta_s=\theta_s^*|D_s^m)$ denotes the conditional mutual information $I(D_t^n;\Theta_t,\Theta_s|D_s^m)$ evaluated at $\Theta_t=\theta_t^*,\Theta_s=\theta_s^*$.

For general loss functions, we define the predictor b_k at time k to be

$$b_k = \text{argmin}_b \mathbb{E}_{Q(D_t^k,D_s^m)}\left[\ell(b,z_t^{(k)})|D_s^m, D_t^{k-1}\right], \tag{20}$$

with the choice of the mixture strategy $Q(D_t^k,D_s^m)=\int P_{\theta_t,\theta_s}(D_t^k,D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s$ for some prior ω. The optimal predictor is then given by

$$b_k^* = \text{argmin}_b \mathbb{E}_{P_{\theta_t^*,\theta_s^*}(D_t^k,D_s^m)}\left[\ell(b,z_t^{(k)})|D_s^m, D_t^{k-1}\right]. \tag{21}$$

As a consequence, we arrive at the following theorem for bounded and exponentially concave loss functions.

Theorem 5 *(Expected Regret with Bounded Loss for OTL). Assume the loss function satisfies $|\ell(b,z)-\ell(b^*,z)|\le M$ for any observation z and the predictors b, b^*. Then the true expected regret induced by b_k and b_k^* in Equation (20) and (21) can be bounded as*

$$\mathcal{R}_O \le M\sqrt{2nI(D_t^n;\Theta_t=\theta_t^*,\Theta_s=\theta_s^*|D_s^m)}. \tag{22}$$

Theorem 6 *(Excess Risk with Exponentially Concave Loss for OTL). Assume the loss function is β-exponentially concave in b for any $z\in\mathcal{Z}$. Then the excess risk induced by b and b^* in Equation (20) and (21) can be bounded as*

$$\mathcal{R}_O \le \frac{1}{\beta}I(D_t^n;\Theta_t=\theta_t^*,\Theta_s=\theta_s^*|D_s^m). \tag{23}$$

The proof follows the same procedures as for the Theorem 3, which we do not repeat here. We can see the analogy from the above theorem that under both the logarithmic loss, bounded and exponentially concave loss, the expected regret induced by the mixture strategy is also characterized by CMI evaluated at θ_t^* and θ_s^*. The expected regret can be considered the accumulated excess risk from sequential prediction, where every single prediction is made from the posterior of the target parameters. Nevertheless, it is not easy to directly spectate the effects of the prior and the sample sizes.

Time-variant transfer learning The treatment of time-variant transfer learning is similar. Under the logarithmic loss, we rewrite the objective function in Equation (5) as

$$\mathcal{R}_{TV} = \sum_{i=1}^{l} \mathbb{E}_{\theta_s^*, \theta_{t,i}^*, \theta_{t,i-1}^*} \left[\sum_{k=1}^{n_i} \ell(b_{k,i}, Z_{t,i}^{(k)}) - \sum_{k=1}^{n_i} \ell(b_{i,k}^*, Z_{t,i}^{(k)}) \right] \tag{24}$$

$$= \sum_{i=1}^{l} \mathbb{E}_{\theta_s^*, \theta_{t,i}^*, \theta_{t,i-1}^*} \left[\frac{P_{\theta_{t,i}^*}(D_{t,i}^{n_i})}{Q(D_{t,i}^{n_i} | D_s^m, D_{t,i-1}^{n_{i-1}})} \right]. \tag{25}$$

We will use the mixture strategy by defining the random variable Θ_s, $\Theta_{t,i}$ and $\Theta_{t,i-1}$ over Λ such that with some prior ω, we can formulate the conditional distribution as

$$Q(D_{t,i}^{n_i} | D_s^m, D_{t,i-1}^{n_{i-1}}) = \frac{Q(D_s^m, D_{t,i-1}^{n_{i-1}}, D_{t,i}^{n_i})}{Q(D_{t,i-1}^{n_{i-1}}, D_s^m)} \tag{26}$$

$$= \frac{\int P_{\theta_s, \theta_{t,i-1}, \theta_{t,i}}(D_s^m, D_{t,i-1}^{n_{i-1}}, D_{t,i}^{n_i}) \omega(\theta_s, \theta_{t,i-1}, \theta_{t,i}) d\theta_s d\theta_{t,i-1} d\theta_{t,i}}{\int P_{\theta_s, \theta_{t,i-1}}(D_s^m, D_{t,i-1}^{n_{i-1}}) \omega(\theta_s, \theta_{t,i-1}) d\theta_s d\theta_{t,i-1}} \tag{27}$$

$$= \int P_{\theta_{t,i}}(D_{t,i}^{n_i}) \omega(\theta_{t,i} | \theta_s, \theta_{t,i-1}) d\theta_{t,i} Q(\theta_s, \theta_{t,i-1} | D_s^m, D_{t,i-1}^{n_{i-1}}) d\theta_s d\theta_{t,i-1}. \tag{28}$$

The above prediction distribution suggested that the posterior $Q(\theta_s, \theta_{t,i-1} | D_s^m, D_{t,i-1}^{n_{i-1}})$ firstly gives an estimate of the source parameter and previous target parameter with marginal $\omega(\theta_s, \theta_{t,i-1})$, then the knowledge transfer is reflected on the conditional prior $\omega(\theta_{t,i} | \theta_s, \theta_{t,i-1})$ that may result in a good approximation of $\theta_{t,i}^*$. The conditional prior $\omega(\theta_{t,i} | \theta_s, \theta_{t,i-1})$ can be interpreted as our prior knowledge of the current target state given the previous state and auxiliary source parameters.

Remark 2. At each episode i, we view the sequential predictors as the conditional distribution $Q(D_{t,i}^{n_i} | D_s^m, D_{t,i-1}^{n_{i-1}})$ since we only use the previous target data from episode $i-1$, and it should be recognized that this choice is not necessarily the optimal choice. The reasons that we discard earlier target data are two-fold. On the one hand, if i becomes large, the posterior will be hard to compute using all earlier target data, and the mixture strategy will become very complicated and inefficient. On the other hand, as the relationship between the target data at episode i and the target data earlier than episode $i-1$ is not explicitly recognized, if the prior distribution is chosen improperly without prior knowledge, introducing such data may result in worse performance.

By this specific strategy, we have the expected regret in the following theorem.

Theorem 7 *(Expected Regret with Logarithmic Loss for TVTL). Under the logarithmic loss, the expected regret in (5) can be written as*

$$\mathcal{R}_{TV} = \sum_{i=1}^{l} I(D_{t,i}^{n_i}; \Theta_{t,i} = \theta_{t,i}^*, \Theta_{t,i-1} = \theta_{t,i-1}^*, \Theta_s = \theta_s^* | D_s^m, D_{t,i-1}^{n_{i-1}}). \tag{29}$$

Similarly, we can easily generalize the logarithmic loss to other bounded loss ℓ. Given any loss function ℓ, we define the predictor $b_{k,i}$ at episode i to be

$$b_{k,i} = \text{argmin}_b \mathbb{E}_{Q(D_{t,i}^k, D_{t,i-1}^{n_{i-1}}, D_s^m)} \left[\ell(b, z_{t,i}^{(k)}) | D_s^m, D_{t,i-1}^{n_{i-1}}, D_{t,i}^{k-1} \right], \tag{30}$$

with the choice of the mixture strategy

$$Q(D_{t,i}^k, D_{t,i-1}^{n_{i-1}}, D_s^m) = \int P_{\theta_{t,i}, \theta_{t,i-1}, \theta_s}(D_{t,i}^k, D_{t,i-1}^{n_{i-1}}, D_s^m) \omega(\theta_{t,i}, \theta_{t,i-1} \theta_s) d\theta_{t,i} d\theta_{t,i-1} d\theta_s \tag{31}$$

for some prior ω. The optimal predictor is then given by

$$b_{k,i}^* = \text{argmin}_b \mathbb{E}_{P_{\theta_{t,i}^*, \theta_{t,i-1}^*, \theta_s^*}(D_{t,i}^k, D_{t,i-1}^{n_{i-1}}, D_s^m)} \left[\ell(b, z_{t,i}^{(k)}) | D_s^m, D_{t,i-1}^{n_{i-1}}, D_{t,i}^{k-1} \right]. \tag{32}$$

Then following Theorem 2 and 5, we arrive at the theorem that describes the expected regret for time-variant transfer learning.

Theorem 8 *(Expected Regret with Bounded Loss for TVTL). Assume the loss function satisfies $|\ell(b, z) - \ell(b^*, z)| \leq M$ for any observation z and the predictors b, b^*. Then the true expected regret induced by b_k and b_k^* in Equation (30) and (32) can be bounded as*

$$\mathcal{R}_{TV} \leq M \sqrt{2l \sum_{i=1}^{l} n_i I(D_{t,i}^{n_i}; \Theta_{t,i} = \theta_{t,i}^*, \Theta_{t,i-1} = \theta_{t,i-1}^*, \Theta_s = \theta_s^* | D_s^m, D_{t,i-1}^{n_{i-1}})}. \tag{33}$$

In this section, we characterize the excess risk and expected regrets for instantaneous and online transfer learning scenarios using information-theoretic quantities from a Bayesian perspective. Given bounded or logarithmic loss functions, the performance is captured by the conditional mutual information between the parameters and the test data. The bound implicitly embeds our prior knowledge over target and source parameters in the prior distribution ω. However, the bounds in their current forms are less informative as they do not show the effects of the prior knowledge ω and sample sizes of the source and target domains. To this end, we will give an asymptotic approximation for conditional mutual information in the following subsection.

3.2. Asymptotic approximation for conditional mutual information

Instantaneous transfer learning To investigate the effect of sample size and prior, first we make the regular assumptions on parametric conditions [9,10] and define the proper prior.

Assumption 3 *(Parametric Condition).* We make the following assumptions for the source and target distributions:

- The source and target distributions $P_{\theta_s^*}(Z_s)$ and $P_{\theta_t^*}(Z_t)$ are twice continuously differentiable at θ_s^* and θ_t^* for almost every Z_s and Z_t.
- For any $\theta_t, \theta_s \in \Lambda$, there exist $\delta_s, \delta_t > 0$ satisfying,

$$\mathbb{E}_{\theta_t}\left[\sup_{\|\theta_t-\theta_t^*\|\le\delta}\left|\frac{\partial}{\partial\theta_{t,i}}\log P_{\theta_t}(Z_t)\right|\right] < \infty, \tag{34}$$

$$\mathbb{E}_{\theta_s}\left[\sup_{\|\theta_s-\theta_s^*\|\le\delta}\left|\frac{\partial}{\partial\theta_{s,i}}\log P_{\theta_s}(Z_s)\right|\right] < \infty \tag{35}$$

  for $i = 1, \cdots, d$. In addition, we assume,

$$\mathbb{E}_{\theta_t}\left[\sup_{\|\theta_t-\theta_t^*\|\le\delta}\left|\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log P_{\theta_t}(Z_t)\right|^2\right] < \infty, \tag{36}$$

$$\mathbb{E}_{\theta_s}\left[\sup_{\|\theta_s-\theta_s^*\|\le\delta}\left|\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log P_{\theta_s}(Z_s)\right|^2\right] < \infty \tag{37}$$

  for any $i, j = 1, \cdots, d$.
- Let $D_{\mathrm{KL}}(P_{\theta_s^*}\|P_{\theta_s})$ and $D_{\mathrm{KL}}(P_{\theta_t^*}\|P_{\theta_t})$ denote the information (KL) divergence for source distribution P_{θ_s} and target distribution P_{θ_t}. We assume they are twice continuously differentiable at θ_s^* and θ_t^*, with the Hessian $J_s(\theta_s^*)$ and $J_t(\theta_t^*)$ positive definite, which are defined by:

$$J_s(\theta_s) = \left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}D_{\mathrm{KL}}(P_{\theta_s^*}\|P_{\theta_s})\right]_{i,j=1\cdots,d}, \tag{38}$$

$$J_t(\theta_t) = \left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}D_{\mathrm{KL}}(P_{\theta_t^*}\|P_{\theta_t})\right]_{i,j=1\cdots,d}. \tag{39}$$

- The convergence of a sequence of parameter values is equivalent to the weak convergence of the distributions they index, e.g.,

$$\theta \to \theta^* \Leftrightarrow P_\theta \to P_{\theta^*}, \tag{40}$$

  for both source and target distributions.

Definition 2 *(Proper Prior).* Given a prior $\omega(\Theta_s, \Theta_t)$, we say,

- the induced marginal density $\omega(\Theta_s)$ is proper if it is continuous and positive over the whole support $\Lambda \subseteq \mathbb{R}^d$.
- the conditional density $\omega(\Theta_t|\Theta_s)$ is proper if there exist some $\delta_s > 0$ and $\delta_t > 0$ such that $\omega(\theta_t|\theta_s) > 0$ for any θ_s and θ_t satisfying $\|\theta_s - \theta_s^*\| \le \delta_s$ and $\|\theta_t - \theta_t^*\| \le \delta_t$.
- $\omega(\Theta_s, \Theta_t)$ is proper if both $\omega(\Theta_s)$ and $\omega(\Theta_t|\Theta_s)$ are proper.

We define the proper prior to ensure that the posterior distribution of Θ_t and Θ_s given D_t^n and D_s^m will asymptotically concentrate on neighborhoods of θ_t^* and θ_s^*, respectively. Specifically, if $\omega(\Theta_s)$ is positive and continuous in Λ, the posterior of Θ_s will concentrate around θ_s^* with sufficient source data. With the posterior of Θ_s, the knowledge is transferred via

the conditional distribution $\omega(\Theta_T|\Theta_s)$ where we could eventually obtain a good estimation of θ_t^* only if $\omega(\Theta_T|\Theta_s)$ has a positive density around θ_t^*. We also say $\omega(\Theta_s)$ or $\omega(\Theta_t|\Theta_s)$ is improper if it does not satisfy conditions in Definition 2.[1]

Intuitively speaking, continuity and positivity are two conditions of the asymptotic behavior of posterior distributions. Specifically, the first point ensures that θ_s^* can be estimated accurately with sufficient source data. The second point ensures that with the accurately estimated θ_s, the conditional distribution $\omega(\Theta_t|\Theta_s)$ is proper, and we have some positive density around the true target distribution θ_t^*. If an improper prior $\omega(\Theta_t|\Theta_s)$ is posed, then we can never get an accurate estimation of θ_t^* even with sufficient source and target data as the posterior will never converge to its true distribution. Hence, such a prior will instead harm the performance and lead to negative transfer. On the other hand, if $\omega(\Theta_t|\Theta_s)$ can encourage a tighter concentration around θ_t^* given the information of θ_s, then the knowledge from the source domain is helpful, and such a prior will lead to the positive transfer. Later, we will have a more detailed discussion on both negative and positive transfer in Section 3.3, and we will also reflect on the effect of the prior with a toy Bernoulli example in Section 3.4.

Allowing both k and m to be sufficiently large, we can obtain the following asymptotic results for the excess risk when both Θ_s and Θ_t are scalars.

Theorem 9 *(Instantaneous Prediction with Scalar Parameters). Under Assumptions 1 and 3, for $\Lambda = \mathbb{R}$ and $\theta_s^* \neq \theta_t^*$, as $n, m \to \infty$, the mixture strategy with a proper prior $\omega(\Theta_s, \Theta_t)$ for logarithmic loss yields*

$$\mathcal{R}_I \asymp \frac{1}{n}. \tag{41}$$

When $\theta_s^* \neq \theta_t^*$, the above theorem characterizes the excess risk with the rate of $O(\frac{1}{n})$, which achieves the optimal convergence rate for parametric distribution estimation [22,70]. However, the expression does not involve the source sample m and the prior distribution $\omega(\Theta_s, \Theta_t)$, indicating that the source data does not help the prediction asymptotically. This is intuitive because given enough target data, we could precisely estimate the underlying target distribution. Hence the source data is not needed in this asymptotic regime. However, we will see that this is not the case anymore in the OTL setup.

The above result can be extended to a more typical transfer learning scenario where $\Theta_t, \Theta_s \in \mathbb{R}^d$ with $d > 1$ share some common parameters $\Theta_c \in \mathbb{R}^j$ for $0 \leq j \leq d$. To illustrate, we can write the parameters in the following way,

$$\Theta_s = (\Theta_{c,1}, \Theta_{c,2}, \cdots, \Theta_{c,j}, \quad \Theta_{s,1}, \cdots, \Theta_{s,d-j}) = (\Theta_c, \Theta_{sr}), \tag{42}$$

$$\Theta_t = (\underbrace{\Theta_{c,1}, \Theta_{c,2}, \cdots, \Theta_{c,j}}_{\text{common parameters}}, \quad \underbrace{\Theta_{t,1}, \cdots, \Theta_{t,d-j}}_{\text{task-specific parameters}}) = (\Theta_c, \Theta_{tr}), \tag{43}$$

where $\Theta_c \in \mathbb{R}^j$ denotes the common parameter vector and $\Theta_{sr}, \Theta_{tr} \in \mathbb{R}^{d-j}$ are task-specific parameter vectors for source and target data, respectively. Then we reach the following theorem that gives the asymptotic estimation of the excess risk with $d > 1$.

Theorem 10 *(Instantaneous Prediction with General Parameterization). Under Assumptions 1 and 3, assume θ_s^* and θ_t^* are characterized in (42) and (43), and $m \asymp n^p$ for some $p \geq 0$. Let $n \to \infty$, the mixture strategy with a proper prior $\omega(\Theta_s, \Theta_t)$ for logarithmic loss yields*

$$\mathcal{R}_I \asymp \frac{d-j}{n} + \frac{j}{n \vee n^p}, \tag{44}$$

Remark 3. From the theorem above, we can conclude that if there is no common parameter $j = 0$ and $d = 1$, we could then recover the result in Theorem 9. If $j > 0$, the source domain will share some parameters with the target domain and the source data will indeed help improve the "learning cost" of the common parameters. Compared with the typical result without the source as

$$\mathcal{R}_I \asymp \frac{d}{n}, \tag{45}$$

the improvement is associated with the component $\frac{j}{n \vee n^p}$, which can be interpreted as the learning cost of θ_c^*. If m is superlinear in n (e.g., $p > 1$), the source samples indeed improve the convergence rate of the estimation for θ_c^* (but does not change the rate of the estimation for θ_{tr}^*). Moreover, if we consider the extreme case $j = d$ such that the source and target have the same parameterization, the risk will be

[1] In Bayesian statistical inference (e.g., [5]), the terminology "improper prior" refers to the class of prior distribution whose sum or integral is infinite, i.e., not necessarily a distribution. With a little abuse of terminology, we use the term "improper prior" to refer to the prior that does not satisfy the Definition 2.

$$\mathcal{R}_I \asymp \frac{d}{n^p \vee n}, \tag{46}$$

where source data can yield a better convergence rate for the excess risk compared to Equation (45) if $p > 1$.

Online transfer learning For online transfer learning where both Θ_s and Θ_t are scalars, we give the asymptotic estimation for CMI as follows.

Theorem 11 *(Online Prediction with Scalar Parameters). Under Assumptions 1 and 3, for $\Lambda = \mathbb{R}$ and $\theta_s^* \neq \theta_t^*$, as $n, m \to \infty$, the mixture strategy with proper prior $\omega(\Theta_s, \Theta_t)$ for logarithmic loss yields*

$$\mathcal{R}_O - \frac{1}{2}\log\frac{n}{2\pi e} \to \frac{1}{2}\log I_t(\theta_t^*) + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)}, \tag{47}$$

where we define the Fisher information matrix $\mathbb{E}_{\Theta_t}\left[-\nabla^2_{\Theta_t}\log P_{\Theta_t}(Z_t)\right]$ evaluated at $\Theta_t = \theta_t^*$ as $I_t(\theta_t^*)$.

Remark 4. Compared to the result without the source data when target sample is abundant [10],

$$\mathcal{R}_O - \frac{1}{2}\log\frac{n}{2\pi e} \to \frac{1}{2}\log I_t(\theta_t^*) + \log\frac{1}{\hat{\omega}(\theta_t^*)} \tag{48}$$

for some prior $\hat{\omega}(\Theta_t)$, the difference between Equation (47) and (48) is $\frac{\hat{\omega}(\theta_t^*)}{\omega(\theta_t^*|\theta_s^*)}$. It says that if the distribution ω can be chosen such that $\frac{\hat{\omega}(\theta_t^*)}{\omega(\theta_t^*|\theta_s^*)} < 1$, the source data will help to reduce the regret by some constant in the scalar parameter case. However, it should be noted that ω is chosen without knowing the exact value of θ_t^* and θ_s^*, so it is not immediately clear if this is always possible. We will show later that if the conditional prior $\omega(\theta_t|\theta_s)$ is proper, it is always possible to find a distribution such that $\frac{\hat{\omega}(\theta_t^*)}{\omega(\theta_t^*|\theta_s^*)} < 1$. On the contrary, if the prior information between the source and target is incorrect, we cannot guarantee that $\frac{\hat{\omega}(\theta_t^*)}{\omega(\theta_t^*|\theta_s^*)} > 1$ always hold, hence negative transfer will occur in the worst-case scenario. This result provides a formal characterization of *negative transfer*.

Notice that the source samples change the constant from $\log\frac{1}{\hat{\omega}(\theta_t)}$ to $\log\frac{1}{\omega(\theta_t|\theta_s)}$, which is independent from n. Hence the effect of the source samples vanishes asymptotically as n goes to infinity. However, the asymptotic analysis is still useful for two reasons. Firstly, we will show later in Corollary 1 that when both n and m approach infinity, the sample complexity of the regret (i.e., how regret scales in terms of m and n) will change, depending on how fast m and n grow relative to each other. Secondly, our numerical results show that the asymptotic bound is in fact very accurate even for relatively small m and n.

Theorem 11 holds when the distributions are parametrized by scalars; that is, the source and target do not share parameters, and the knowledge transfer is reflected on the prior knowledge. Therefore, the effect of the source sample size m is not exhibited in this case. The following theorem characterizes the expected regret under general parametrization where Θ_s and Θ_t will share j common parameters.

Theorem 12 *(Online Prediction with General Parametrization). Under Assumptions 1 and 3, with $\Theta_s, \Theta_t \in \mathbb{R}^d$ defined above and as $n \to \infty$ and $m = cn^p$ for some $c > 0$, the mixture strategy with proper prior $\omega(\Theta_s, \Theta_t)$ yields*

$$\mathcal{R}_O - \frac{1}{2}\log\det(\mathbf{I}_{j\times j} + \frac{1}{cn^{p-1}}\Delta_t\Delta_s^{-1}) - \frac{1}{2}\log\det(nI_t(\theta_{tr}^*)) \to \frac{d-j}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)} \tag{49}$$

where Δ_t, Δ_s and $I_t(\theta_{tr}^*)$ are Fisher information matrices related quantities and their definitions can be found in (181) - (186) in Appendix A.9.

Corollary 1 *(Rate of $\mathcal{R}_O$). Under the conditions in Theorem 12, for $0 \leq p < 1$:*

$$\mathcal{R}_O \asymp j(1-p)\log n + (d-j)\log n. \tag{50}$$

For $p \geq 1$:

$$\mathcal{R}_O \asymp \frac{j}{n^{p-1}} + (d-j)\log n. \tag{51}$$

Remark 5. We can intuitively interpret the term $\frac{1}{2}\log\det(\mathbf{I}_{j\times j} + \frac{1}{cn^{p-1}}\Delta_t\Delta_s^{-1})$ in the Equation (49) as the learning cost of θ_c, which is captured by the source sample sizes m. Compared with the typical result without the source as

Table 1
Results on the convergence rate of the excess risk under different conditions given $\theta_s^* \neq \theta_t^*$.

Learning Type	Λ	Condition	Rate
Instantaneous TL	$\mathbb{R}$	$m = cn^p, 0 \leq p < 1$	$O(\frac{1}{n})$
Instantaneous TL	$\mathbb{R}$	$m = cn^p, p \geq 1$	$O(\frac{1}{n})$
Instantaneous TL	$\mathbb{R}^d$ in total, $\mathbb{R}^j$ in common	$m = cn^p, 0 \leq p < 1$	$O(\frac{d}{n})$
Instantaneous TL	$\mathbb{R}^d$ in total, $\mathbb{R}^j$ in common	$m = cn^p, p \geq 1$	$O(\frac{d-j}{n} + \frac{j}{m})$
Online TL	$\mathbb{R}$	$m = cn^p, 0 \leq p < 1$	$O(\log n)$
Online TL	$\mathbb{R}$	$m = cn^p, p \geq 1$	$O(\log n)$
Online TL	$\mathbb{R}^d$ in total, $\mathbb{R}^j$ in common	$m = cn^p, 0 \leq p < 1$	$O(d \log n - j \log m)$
Online TL	$\mathbb{R}^d$ in total, $\mathbb{R}^j$ in common	$m = cn^p, p \geq 1$	$O((d-j)\log n)$

$$\mathcal{R}_O - \frac{d}{2}\log\frac{n}{2\pi e} \to \frac{1}{2}\log\det(I_t(\theta_t^*)) + \log\frac{1}{\hat{\omega}(\theta_t^*)} \tag{52}$$

with the rate of $O(d\log n)$, if m is

- sublinear in n ($p < 1$), the rate is improved to $O(j(1-p)\log n)$. In this case, as the source data size is small compared to the target data size, the learning rate (for the common parameter part) is only improved by a constant factor $1-p$, changed from $j\log n$ to $j(1-p)\log n$.
- linear in n ($p = 1$), the rate (for learning in the common parameters) in this case improves from $O(j\log n)$ to a constant $O(j)$.
- superlinear in n ($p > 1$), the rate is $O(\frac{j}{n^{p-1}})$, indicating that abundant source samples indeed improve the performance and the cost for learning the common parameters vanishes in this case.

Furthermore, the prior knowledge $\omega(\theta_t^*|\theta_s^*)$ becomes involved in characterizing the expected regret as we discussed in Remark 4. However, it can only change the constant but does not change the rate.

As a special case, if there is no common parameters ($j = 0$), then as both m and n are sufficiently large:

$$\mathcal{R}_O - \frac{d}{2}\log\frac{n}{2\pi e} \to \frac{1}{2}\log\det(I_t(\theta_{tr}^*)) + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)} \tag{53}$$

Let $d = 1$, we can recover the results in Theorem 11 and the knowledge transfer is only reflected on the prior knowledge $\omega(\theta_t^*|\theta_s^*)$. If the number of the common parameters is d ($j = d$), that is, the source and target distributions are characterized by the same parameters, which yields the asymptotic estimation as

$$\mathcal{R}_O - \frac{1}{2}\log\det(\mathbf{I}_{d\times d} + \frac{1}{cn^{p-1}}\Delta_t\Delta_s^{-1}) \to \log\frac{1}{\omega(\theta_t^*|\theta_s^*)} \tag{54}$$

Under this case, the regret rate depends on the source sample size by setting $j = d$ in Corollary 1. Compared to the ITL case, as seen in Theorem 1 and 2, the excess risk induced by ITL is only associated with the sample number m and n since the posterior overwhelms the prior for single data prediction. While the expected regret in OTL is further determined by conditional prior distribution as shown in Theorem 11 and 12. We summarize the results for ITL and OTL in Table 1 to show the effectiveness of the common parameters between θ_s^* and θ_t^*.

From the table, it can be observed that for the scalar support domain $\Lambda = \mathbb{R}$, since $\theta_t^* \neq \theta_s^*$, the convergence rate will only depend on the target sample. The source data will affect the term $\log\frac{1}{\omega(\theta_t|\theta_s)}$ as we show in Theorem 11. However, if we consider $\Lambda = \mathbb{R}^d$ and there are j common parameters, the source data may change the convergence rate depending on the sample size and the type of the transfer learning problem. Taking ITL as an example, if we have fewer source data where $0 \leq p < 1$, the convergence rate is dominated by the target data and will not change even with the increased dimension j for common parameters. But if $p \geq 1$, the convergence rate is partially improved from $O(\frac{j}{n})$ to $O(\frac{j}{m})$ for the common parameters. For OTL, when $0 \leq p < 1$ and $d, j \geq 1$, the rate of the cumulative regret will be partially improved by $-j\log m$ compared to the case without the source (e.g., $O(d\log n)$). However, the best convergence rate we can achieve is $O((d-j)\log n)$ even if we increase the source data (to $p \geq 1$) as the domain-specific parameters can only be estimated from the target data.

Despite our conclusion being based on asymptotic analysis, we provide experimental results on the regret, such as in Section 4.2 on the logistic regression transfer problem, to demonstrate that CMI accurately captures the pattern of true regret and the impact of prior knowledge. The obtained upper bounds can also serve as a useful guideline for true regret even with limited sample sizes. However, it should be noted that providing a rigorous proof for non-asymptotic regret may require additional techniques beyond KL divergence, as discussed in [26,8], where the non-asymptotic analysis for Bayesian

posterior primarily relies on other distribution divergences such as the Wasserstein distance or maximum mean discrepancy, which suggest a possible extension as future work.

Time-variant transfer learning Under the time-variant transfer learning, with the aforementioned assumption that at episode i, the target parameter $\Theta_{t,i}$ have c_i target common parameters with $\Theta_{t,i-1}$, and Θ_s shares j_i source sharing parameters with $\Theta_{t,i-1}$ and $\Theta_{t,i}$. Let us define the random variables Θ_s, $\Theta_{t,i-1}$ and $\Theta_{t,i}$ with the following parameterization,

$$\Theta_s = (\Theta_{c,1}, \Theta_{c,2}, \cdots, \Theta_{c,j_i}, \underbrace{\Theta_{s,1}, \cdots, \Theta_{s,d-j_i}}_{\text{source-specific parameters}}) = (\Theta_{c,i}, \Theta_{sr,i}), \tag{55}$$

$$\Theta_{t,i-1} = (\Theta_{c,1}, \Theta_{c,2}, \cdots, \Theta_{c,j_i}, \quad \Theta_{v,1}, \cdots, \Theta_{v,c_i}, \quad \Theta_{t',1}, \cdots, \Theta_{t',d-j_i-c_i}) = (\Theta_{c,i}, \Theta_{v,i}, \Theta_{tr,i-1}), \tag{56}$$

$$\Theta_{t,i} = (\underbrace{\Theta_{c,1}, \Theta_{c,2}, \cdots, \Theta_{c,j_i}}_{\text{source sharing parameters}}, \underbrace{\Theta_{v,1}, \cdots, \Theta_{v,c_i}}_{\text{target common parameters}}, \underbrace{\Theta_{t,1}, \cdots, \Theta_{t,d-j_i-c_i}}_{\text{target-specific parameters}}) = (\Theta_{c,i}, \Theta_{v,i}, \Theta_{tr,i}). \tag{57}$$

Here $\Theta_{c,i}$ and $\Theta_{v,i}$ represent the source sharing parameters and the target common parameters under episode i, which are not overlapped. The parameters changing from $\Theta_{tr,i-1}$ to $\Theta_{tr,i}$ exhibit the nature of time-varying target domains. By this particular parameterization and we assume at each episode i, the sample sizes n_i are comparable with n_{i-1} (e.g., $n_{i-1} \asymp n_i$), we reach the following theorem.

Theorem 13 *(Time-variant Target Regret Bounds). Given the time-variant target domain described in the problem formulation, suppose that conditions in Theorem 5 and Assumptions 2 and 3 hold for each $\theta^*_{t,i}$ and θ^*_s for $i = 1, 2, \cdots, k$. We further assume that source parameters will share j_i parameters with every $\theta^*_{t,i}$. In addition, $\theta^*_{t,i}$, $\theta^*_{t,i-1}$ have c_i common parameters. As $n_i \to \infty$ for any i and assume $n_{i-1} \asymp n_i$ and $m \asymp n_i^p$ for some $p \geq 0$, the mixture strategy with proper prior $\omega(\theta_s, \theta_{t,i}, \theta_{t,i-1})$ yields*

$$\mathcal{R}_{TV} \lesssim \sqrt{l \sum_{i=1}^{l} n_i \Big(j_i (1 \wedge n_i^{1-p}) + c_i + (d - c_i - j_i) \log n_i + \frac{2}{\omega(\theta^*_{t,i} | \theta^*_{t,i-1}, \theta^*_s)} \Big)}. \tag{58}$$

As seen in the theorem, the estimations of the parameters are now four-fold. In summation, the first term stands for the estimation cost of source sharing parameters, and the rate depends on the sample size of source samples. Hence, a large sample size ($p > 1$) will contribute to boosting the rate. The second term, concerned with the target common parameters $\Theta_{v,i}$, is a constant c_i calculating from the ratio $\frac{n_i}{n_{i-1}} \sim O(1)$ where it entails that the target data with the previous episode improves the estimation of $\Theta_{v,i}$. For target-specific parameter, the rate is $O((d - j_i - c_i) \log(n_i))$ which coincides with the typical regret growth results. Lastly, the prior knowledge $\omega(\theta^*_{t,i} | \theta^*_{t,i-1}, \theta^*_s)$ (e.g., the knowledge over $\theta^*_{t,i}$ given the previous $\theta^*_{t,i-1}$ and θ^*_s) also plays an important role in the prediction performance for each episode i as can be seen in the OTL case.

3.3. Negative and positive transfer

As previously discussed, $\omega(\Theta_s, \Theta_t)$ should be appropriately chosen so that the posterior will asymptotically converge to the true parameter θ^*_s and θ^*_t. However, if the prior distribution (particularly $\omega(\Theta_t | \Theta_s)$) is imposed improperly, the extra source data do not necessarily help improve our prediction for target data. Roughly speaking, if our prior knowledge on θ^*_s and θ^*_t is incorrect, under our scheme, this could translate to an improper prior distribution for the mixture strategy. We will show that in the worst case, with an improper prior, the extra source data will, in fact, cause a higher regret (i.e., worse performance) compared to the case without source data, known as the *negative transfer*. We also study the *positive transfer* case under the proposed mixture strategy where the source data improve the prediction performance.

Negative transfer Let us start with a simple Bernoulli transfer example to understand the negative transfer. Consider Z_s, Z_t take values in $\{0, 1\}$ and assume θ^*_s and θ^*_t are the probabilities that the source and target samples take value in 1. Also assume that our prior knowledge on the parameters is that $|\theta^*_s - \theta^*_t| \leq 0.1$ given any $\theta^*_s \in \Lambda$. Suppose the underlying parameters are $\theta^*_t = 0.6$ and $\theta^*_s = 0.8$. In other words, our prior knowledge is incorrect. In this case, even with the perfect knowledge of θ^*_s, no algorithm can correctly estimate θ^*_t if the (incorrect) prior knowledge $|\theta_s - \theta_t| \leq 0.1$ is imposed, even with infinitely many target samples. Consequently, the expected risk becomes higher than the case without knowing such prior. In section 3.4, we present detailed analytical and experimental analyses for Bernoulli examples with different priors. Generally speaking, it is recognized that learning performance is captured by the divergence between the estimated distribution $P_{\theta_{est}}$ and the true distribution P_{θ^*}. Firstly, we consider the ITL problem under logarithmic loss and then concretely point out that improper prior will lead to the negative transfer.

Proposition 1 *(Negative Transfer for Instantaneous Transfer Learning). Under Assumptions 1 and 3, as $n, m \to \infty$, the mixture strategy with a proper $\omega(\Theta_s)$ but an improper $\omega(\Theta_t | \Theta_s)$ for logarithmic loss yields*

$$\mathcal{R}_I \geq D_{\mathrm{KL}}(P_{\theta_t^*} \| P_{\tilde{\theta}_t}), \tag{59}$$

where $\tilde{\theta}_t = \min_{\theta_t \in \mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} D_{\mathrm{KL}}(P_{\theta_t^*} \| P_{\theta_t})$.

Compared to the result in Theorem 1, the excess risk, in this case, does not converge to zero even when n goes to infinity because it can be shown that $\tilde{\theta}_t$ does not coincide with θ_t^*. In other words, if the wrong prior information about the parameters leads to a choice of an improper prior distribution, the posterior of θ_t will approach $\tilde{\theta}_t$ instead of the true parameter θ_s^* and the source data will hurt the performance instead even if we have abundant target data. It can be shown that a similar phenomenon occurs in online predictions, as characterized by the following proposition.

Proposition 2 *(Negative Transfer for Online Transfer Learning). Let* $\mathcal{R}_0^{\omega(\Theta_s,\Theta_t)}(n)$ *denote the regret induced by the mixture strategy* $Q(D_t^n|D_s^m)$ *with the source data and the prior* $\omega(\Theta_s, \Theta_t)$*. Under logarithmic loss, if* $\omega(\Theta_s)$ *is proper but* $\omega(\Theta_t|\Theta_s)$ *is improper, the following inequality holds when both n and m are sufficiently large.*

$$\mathcal{R}_0^{\omega(\Theta_s,\Theta_t)}(n) \geq n D_{\mathrm{KL}}(P_{\theta_t^*} \| P_{\tilde{\theta}_t}), \tag{60}$$

where $\tilde{\theta}_t = \mathit{argmin}_{\theta_t \in \mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} D_{\mathrm{KL}}(P_{\theta_t^*} \| P_{\theta_t})$*. Furthermore, for any proper* $\hat{\omega}(\Theta_t)$*, let* $\mathcal{R}_0^{\hat{\omega}(\Theta_t)}(n)$ *denote the regret induced by* $\hat{Q}(D_t^n)$ *with the prior* $\hat{\omega}(\Theta_t)$ *but without the source data. We have*

$$\mathcal{R}_0^{\omega(\Theta_s,\Theta_t)}(n) > \mathcal{R}_0^{\hat{\omega}(\Theta_t)}(n). \tag{61}$$

For online transfer learning, if we use the wrong prior information, the regret rate is $\Omega(n)$ while the regret is $O(\log n)$ without the source data introduced. Such improper prior leads to a higher regret, and the negative transfer is characterized by the parameters closest to the true parameters θ_t^* in terms of the KL divergence. It immediately follows that the averaged regret $\frac{1}{n}\mathcal{R}_0^{\omega(\Theta_s,\Theta_t)}(n)$ does not asymptotically go to zero. As such, we concretely demonstrate when the negative transfer will happen for both ITL and OTL scenarios and theoretically confirm that the regret is quantitatively captured by the KL divergence between the true distribution and the estimated posterior. It is also worth noting that the improper prior is only a sufficient condition for proving the negative transfer. It is still possible to encounter the negative transfer when the prior density around θ_t^* is relatively small; see Section 4.2 for example.

Positive transfer In contrast, if $\omega(\Theta_s, \Theta_t)$ is chosen properly, we can always find a prior such that the knowledge transfer from source data encourages lower regret, leading to the *positive transfer*. Take the Bernoulli case as an example again. Without the source, the prior $\hat{\omega}(\Theta_t)$ can be selected as a uniform distribution over the whole probability range $[0, 1]$. With the source, if the prior knowledge $\omega(\Theta_s|\Theta_t)$ could encourage a tighter support near θ_t^*, say $[\theta_t^* - c, \theta_t^* + c] \subset [0, 1]$, then there will exist some proper prior that leads to a lower regret. It can be interpreted that the source data narrow down the uncertainty on the choice of θ_t^*, and it is more likely to obtain a more accurate estimation. We first consider the OTL case under logarithmic loss to mathematically interpret the positive transfer and establish the following proposition.

Proposition 3 *(Positive Transfer for OTL). When both n and m are sufficiently large, for any proper* $\hat{\omega}(\Theta_t)$*, we can always find a proper prior* $\omega(\Theta_s, \Theta_t)$ *satisfying the following inequality if the support of* $\omega(\Theta_t|\Theta_s)$ *is a proper subset of* Λ *for any* $\|\theta_s - \theta_s^*\| \leq \delta_s$ *with some* $\delta_s > 0$*.*

$$\mathcal{R}_0^{\omega(\Theta_s,\Theta_t)}(n) < \mathcal{R}_0^{\hat{\omega}(\Theta_t)}(n) \tag{62}$$

As aforementioned, not all proper priors will lead to positive transfer. In our claim, we can find a proper prior that if $\omega(\Theta_t|\Theta_s)$ encourages a tighter support over Θ_t, making use of source data appropriately can narrow down the uncertainty range over Θ_t. It then follows that there will always exists a prior that assigns a more concentrated mass around θ_t^*, which reduces the expected regret.

For the ITL scenario, from Theorem 9 one can see that the prior knowledge is not revealed in characterizing the excess risk since the data overwhelms the prior as both n and m goes sufficiently large. However, the prior knowledge still plays an important role when the sample sizes are limited. To see this, by generalizing Theorem 1 from [22], for any n and m, the excess risk of ITL in (2) can be upper bounded by

$$\mathcal{R}_I \leq -\log \int P(\theta_t|D_t^n) \frac{\omega(\theta_t|\theta_s)}{\hat{\omega}(\theta_t)} e^{-nD_{KL}(P_{\theta_t^*} \| P_{\theta_t})} d\theta_t P(\theta_s|D_s^m) d\theta_s, \tag{63}$$

where $P(\theta_t|D_t^n) = \frac{P_{\theta_t}(D_t^n)\hat{\omega}(\theta_t)}{P(D_t^n)}$ denotes the posterior induced by $\hat{\omega}(\theta_t)$. Here we allow m goes to infinity (abundant source data) but let n be a constant (limited target data). With the proper $\omega(\Theta_s)$, the posterior is approximated as $P(\theta_s|D_s^m) = \delta(\theta_s^*)$, then we can rewrite the inequality for any $n \in \mathbb{N}$,

$$\mathcal{R}_I \leq -\log \int P(\theta_t|D_t^n)\frac{\omega(\theta_t|\theta_s^*)}{\hat{\omega}(\theta_t)}e^{-nD_{KL}(P_{\theta_t^*}\|P_{\theta_t})}d\theta_t. \tag{64}$$

Compared to the results without the source, we have the bound

$$\mathcal{R}_I \leq -\log \int P(\theta_t|D_t^n)e^{-nD_{KL}(P_{\theta_t^*}\|P_{\theta_t})}d\theta_t. \tag{65}$$

It is observed that the ratio $\frac{\omega(\theta_t|\theta_s^*)}{\hat{\omega}(\theta_t)}$ still plays a role in this non-asymptotic bound. Informally speaking, to achieve positive transfer in the single prediction, one may select $\omega(\theta_t|\theta_s^*)$ to be large for those θ_t that leads to low divergence $D_{KL}(P_{\theta_t^*}\|P_{\theta_t})$ and vice versa. As a consequence, the R.H.S. in (64) will be smaller than the R.H.S. in (65). However, due to the fact that the posterior distribution $P(\theta_t|D_s^n)$ cannot be well estimated with finite n, it is relatively difficult to tell whether the positive transfer will happen given a specific prior. Rather, we intuitively explain the role of prior knowledge from its non-asymptotic upper bound.

3.4. Bernoulli example

In this section, we illustrate the results presented for OTL case using a simple Bernoulli example, the results can be calculated using the same procedures for ITL and TVTL cases. We assume the parametric distributions are $P_{\theta_s^*} \sim \text{Ber}(\theta_s^*)$ and $P_{\theta_t^*} \sim \text{Ber}(\theta_t^*)$ for $\theta_s^*, \theta_t^* \in [0,1]$, that is, $P(Z_t^{(k)}=1)=\theta_t^*$ and $P(Z_s^{(k)}=1)=\theta_s^*$ and we also assume $\theta_s^* \neq \theta_t^*$. Given a batch of source data D_s^m with m samples i.i.d. drawn from $P_{\theta_s^*}$ and n target samples D_t^n i.i.d. drawn from $P_{\theta_t^*}$ sequentially, we will make predictions for each target sample $Z_t^{(k)}$ at time k based upon the received target data D_t^{k-1} and source data D_s^m. Under the logarithmic loss ℓ, let $\ell(b_k, Z_t^{(k)}) = Q(Z_t^{(k)}|D_t^{k-1}, D_s^m)$ and $\ell(b_k^*, Z_t^{(k)}) = P_{\theta_t^*}(Z_t^{(k)})$, we are interested in the expected regret $\mathcal{R}_O$ as:

$$\mathcal{R}_O = \mathbb{E}_{\theta_s^*,\theta_t^*}\left[\sum_{k=1}^n \ell(b_k, Z_t^{(k)}) - \sum_{k=1}^n \ell(b_k^*, Z_t^{(k)})\right] \tag{66}$$

$$= \sum_{D_s^m}\sum_{D_t^n} P_{\theta_s^*}\left(D_s^m\right) P_{\theta_t^*}\left(D_t^n\right) \log\left(\frac{P_{\theta_t^*}\left(D_t^n\right)}{Q\left(D_t^n|D_s^m\right)}\right). \tag{67}$$

Without prior knowledge In this section, we analytically show that without any prior knowledge on the relationships between the source and target data (apart from the assumption $\theta_s^* \neq \theta_t^*$), the transfer learning algorithms lead to negative transfer in the worst case. To see this, by assigning a prior distribution $\omega(\Theta_t, \Theta_s)$ over $[0,1]^2$, we firstly define the distribution Q with the mixture strategy as

$$Q\left(D_t^n|D_s^m\right) = \frac{\int_0^1\int_0^1 \omega(\theta_s,\theta_t)P_{\theta_t}(D_t^n)P_{\theta_s}(D_s^m)d\theta_s d\theta_t}{\int_0^1 \omega(\theta_s)P_{\theta_s}(D_s^m)d\theta_s}, \tag{68}$$

on account of the assumption $P_{\theta_s,\theta_t}(D_t^k, D_s^m) = P_{\theta_t}(D_t^k)P_{\theta_s}(D_s^m)$. Without any prior knowledge, we will arbitrarily choose two types of priors $\omega(\Theta_s,\Theta_t)$ and analytically examine their regrets. We start by looking at the case where the joint prior distribution is assumed to be $\omega(\Theta_s,\Theta_t) = \Theta_s + \Theta_t$, which gives the marginal distributions $\omega(\Theta_s) = \Theta_s + \frac{1}{2}$ and $\omega(\Theta_t) = \Theta_t + \frac{1}{2}$. By knowing θ_s, there is still some uncertainty over Θ_t. It can be seen from the conditional distribution

$$\omega(\Theta_t|\theta_s) = \frac{\theta_s}{\theta_s+\frac{1}{2}} + \frac{\Theta_t}{\theta_s+\frac{1}{2}}. \tag{69}$$

The predictor distribution $Q(D_t^n|D_s^m)$ can be calculated explicitly as

$$Q\left(D_t^n|D_s^m\right) = \frac{1}{(n+1)}\frac{1}{\binom{n}{k_t}}\frac{2k_s+2+(k_t+1)\frac{2(m+2)}{n+2}}{m+2k_s+4}, \tag{70}$$

where we denote number of 1's received from the source and target by k_t and k_s, respectively. If m, n are sufficiently large and $\mathbb{E}_{\theta_s^*}[k_s] = \theta_s^* m$ and $\mathbb{E}_{\theta_t^* s}[k_t] = \theta_t^* n$, we expect over a long sequence that the conditional mutual information can be calculated as

$$\mathcal{R}_O^{\omega(\Theta_t,\Theta_s)}(n) = \frac{1}{2}\log(n+1) + \frac{1}{2}\log\frac{1}{\pi\theta_t^*(1-\theta_t^*)} + \log\frac{\theta_s^*+\frac{1}{2}}{\theta_t^*+\theta_s^*}. \tag{71}$$

Compared to the mixture distribution Q by the marginal $\hat{\omega}(\theta_t) = \theta_t + \frac{1}{2}$ without introducing the source

$$Q\left(D_t^n\right) = \int_0^1 (\theta_t + \frac{1}{2})(\theta_t)^{k_t}(1-\theta_t)^{n-k_t} d\theta_t = \frac{1}{(n+1)} \frac{1}{\binom{n}{k_t}} (\frac{1}{2} + \frac{k_t+1}{n+2}), \tag{72}$$

and the expected regret induced by this predictor can be calculated as

$$\mathcal{R}_O^{\hat{\omega}(\Theta_t)}(n) = \frac{1}{2}\log(n+1) + \frac{1}{2}\log\frac{1}{\pi\theta_t^*(1-\theta_t^*)} + \log\frac{1}{\theta_t^* + \frac{1}{2}}. \tag{73}$$

The difference of the regrets in Equation (71) and (73) is $\log\frac{\omega(\theta_t)}{\omega(\theta_t|\theta_s)} = \log\frac{(\theta_s^*+\frac{1}{2})(\theta_s^*+\frac{1}{2})}{\theta_t^*+\theta_s^*}$. From this example, due to the fact that we do not have any specific prior knowledge over θ_t^* and θ_s^*, the prior distribution $\omega(\theta_t|\theta_s)$ may not lead to a better estimation compared to the case without the source data, and the expected regret with the source can larger (e.g., if $(\theta_s^* - \frac{1}{2})(\theta_t^* - \frac{1}{2}) > 0$) in the worst case and the negative transfer will happen. Additionally, even when m goes to infinity, and the source data does not change the convergence rate w.r.t. n, this result confirms the Theorem 11 numerically.

If we consider another extreme case where the prior distribution $\omega(\Theta_t|\Theta_s) = \delta(\Theta_s)$ and $\omega(\Theta_s) = 1$, that is, Θ_s is uniformly distributed and knowing Θ_s is equivalent to knowing Θ_t. Then we obtain the mixture as

$$Q\left(D_t^n|D_s^m\right) = \frac{\frac{1}{n+m+1}\frac{1}{\binom{m+n}{k_s+k_t}}}{\frac{1}{m+1}\frac{1}{\binom{m}{k_s}}}. \tag{74}$$

Analogously we expect over a large m, where $k_s = \theta_s^* m$ and $k_t = \theta_t^* n$ and the source samples are abundant, e.g., $m \gg n$. We have

$$\mathcal{R}_O^{\omega(\Theta_t,\Theta_s)}(n) \le \log(1 + \frac{n}{m+1}) + n\left(\theta_t^*\log\frac{\theta_t^*}{\theta_s^*} + (1-\theta_t^*)\log\frac{1-\theta_t^*}{1-\theta_s^*}\right) + C, \tag{75}$$

where C is some constant that depends on θ_t^* and θ_s^*. By introducing abundant source data, $\log(1+\frac{n}{m+1})$ term will vanish with the rate $O(\frac{n}{m})$ but it will also introduce the term $n\left(\theta_t^*\log\frac{\theta_t^*}{\theta_s^*} + (1-\theta_t^*)\log\frac{1-\theta_t^*}{1-\theta_s^*}\right)$ that grows linearly in n. This specific choice of prior is *improper* since $\omega(\theta_t^*|\theta_s^*) = 0$ whereas $\theta_t^* \neq \theta_s^*$, which will finally lead to the inaccurate estimation of θ_t^* while both m and n are sufficiently large. This result is unsurprising since we can estimate θ_s^* accurately and $\omega(\theta_t|\theta_s^*)$ enforces $\theta_t = \theta_s^*$ and the regret for each sample is at least $D(P_{\theta_t^*}\|P_{\theta_s^*})$. We can confirm it by calculating the true regrets precisely as m is sufficiently large,

$$\mathcal{R}_O^{\omega(\Theta_t,\Theta_s)}(n) = n\left(\theta_t^*\log\frac{\theta_t^*}{\theta_s^*} + (1-\theta_t^*)\log\frac{1-\theta_t^*}{1-\theta_s^*}\right) = nD_{\mathrm{KL}}(P_{\theta_t^*}\|P_{\theta_s^*}). \tag{76}$$

Note that the chosen prior will lead to the negative transfer, and the rate of the expected regret in this case is $O(n)$ as we proved in Proposition 2.

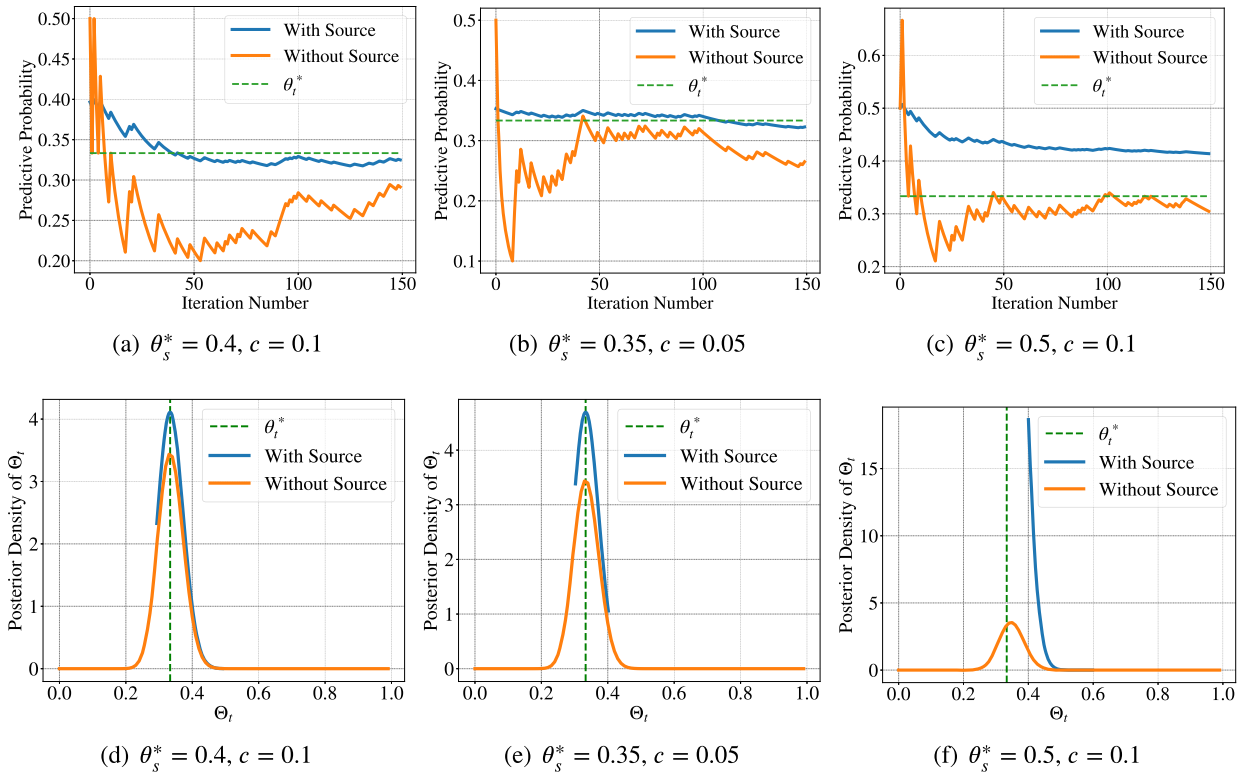
With prior knowledge In the previous section, we demonstrated two choices of prior distributions for the mixture strategy, both leading to negative transfer in the worst case. Now we show that we can always find an appropriate ω for better predictions with some correct prior knowledge. Noticing that the expected regret is captured by the conditional prior $\omega(\Theta_t|\Theta_s)$, even though we do not have access to the true parameters, we may know some relationship between the source and target parameters. In particular, for $\Lambda = [0,1] \subset \mathbb{R}$, we make the following assumption.

Assumption 4 *(Prior knowledge with ℓ_1 norm).* For $\theta_s^*, \theta_t^* \in [0,1]$ and $c > 0$,

$$|\theta_s^* - \theta_t^*| \le c. \tag{77}$$

This assumption implies that θ_t^* is not far away from θ_s^*, and if θ_s^* can be approximated accurately, θ_t^* can be estimated more precisely with a tighter support. We encode this particular relationship in terms of the conditional prior distribution $\omega(\Theta_t|\theta_s)$, say, given any θ_s, Θ_t is uniformly distributed over $[\theta_s - c, \theta_s + c]$ with density $\frac{1}{2c}$ and the hyperparameter c can be interpreted as our knowledge level. Larger c indicates that we are more uncertain about θ_t^* given θ_s^* and vice versa. Additionally, we assume the marginal $\omega(\theta_s)$ is proper, e.g., θ_s^* can be estimated accurately with sufficient source samples. As a result, we can give the explicit expression of the mixture distribution $Q(D_t^n|D_s^m)$,

Lemma 1. *Under Assumption 4, given any θ_s, we assume Θ_t is uniformly distributed over $[\theta_s - c, \theta_s + c]$ with density $\frac{1}{2c}$. Then $\hat{\theta}_s$ is estimated via the source samples D_s^m where its posterior distribution can be written as $P(\hat{\theta}_s|D_s^m) = \frac{\omega(\hat{\theta}_s)P_{\hat{\theta}_s}(D_s^m)}{Q(D_s^m)}$ for any proper prior $\omega(\hat{\theta}_s)$. Then we have,*



(a) $\theta_s^* = 0.4, c = 0.1$ (b) $\theta_s^* = 0.35, c = 0.05$ (c) $\theta_s^* = 0.5, c = 0.1$

(d) $\theta_s^* = 0.4, c = 0.1$ (e) $\theta_s^* = 0.35, c = 0.05$ (f) $\theta_s^* = 0.5, c = 0.1$

Fig. 2. With different c and θ_s^*, subfigures (a), (b) and (c) show the predictive probability $P(z_t^{(k)} = 1 | D_t^{k-1}, D_s^m)$ with single trial. The prediction the curves that are closer to $\theta_t^* = \frac{1}{3}$ entails more accurate estimation. Subfigures (d), (e) and (f) show the probability density of the posterior $P(\Theta_t | D_t^n, D_s^m)$ (Blue) and $P(\Theta_t | D_t^n)$ (Orange) after receiving 150 target samples. The posterior with the closer θ_s^* and smaller c leads to higher density around θ_t^*. The green dashed lines represent the true θ_t^* for reference. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$Q(D_t^n | D_s^m) = \mathbb{E}_{\hat{\theta}_s | D_s^m} \left[\frac{1}{2c} \binom{n}{k_t}^{-1} \left(\sum_{i=1}^{k_t} \binom{n}{i} \frac{(\hat{\theta}_s - c)^i (1 - \hat{\theta}_s + c)^{n-i+1}}{n-i+1} - \frac{(\hat{\theta}_s + c)^i (1 - \hat{\theta}_s - c)^{n-i+1}}{n-i+1} + \frac{(1 - \hat{\theta}_s + c)^{n+1} - (1 - \hat{\theta}_s - c)^{n+1}}{n+1} \right) \right]. \tag{78}$$

Remark 6. It is relatively hard to directly tell the effect of c from the above expression. Since $\hat{\theta}_s$ will be concentrating around θ_s^* with large m, we then can estimate $Q(D_t^n | D_s^m)$ empirically if all the parameters c, n, k_t are known. We shall verify this intuition by conducting experiments with different parameters to show the effect of the prior knowledge.

To examine the effect of hyperparameter c under Assumption 4, numerical experiments are conducted and results are presented. Consider the following specific settings, let the true parameter $\theta_t^* = \frac{1}{3}$. Under Assumption 4, we assume $\omega(\Theta_s) = 1$ over $[0, 1]$ and given θ_s, Θ_t is distributed uniformly over $[\theta_s - c, \theta_s + c]$. As the target arrives sequentially and we have sufficient source data with $m = 10000$, we plot the predictive probability $P(Z_t^{(k)} | D_t^{k-1}, D_s^m)$ in Fig. 2(a), 2(b) and 2(c) for a single trial with different θ_s^* and c.

From Figs. 2(a) and 2(b), we observe that when the target sample size k is relatively small, the prior knowledge about the source and the target distribution can help estimate the true value of θ_t better than without introducing the source data. It can also be seen that when k is relatively large, both estimates of θ_t^* with or without the source are fairly close to the true value. However, in Fig. 2(c) the prior knowledge is improper as we set $\theta_s^* = 0.5$ and $c = 0.1$, whereas the true target parameter is not contained in the effective support, e.g., $[0.4, 0.6]$. Even with the iteration number increasing, the predictive probability approaches 0.4, which is the best predictor within the support, provably verifying the Proposition 2. Since the single trial does not fully reflect the usefulness of the source data, we next examine the posterior distribution $P(\Theta_t | D_t^n, D_s^m)$ with source and $P(\Theta_t | D_t^n)$ without source after receiving 150 target data with different θ_s^* and c to see the effect of prior knowledge as shown in Fig. 2(d), 2(e) and 2(f). From the comparisons, the posterior distribution of Θ_t with the source data is more concentrated under the correct prior knowledge. Moreover, a closer θ_s^* and a smaller c will yield a

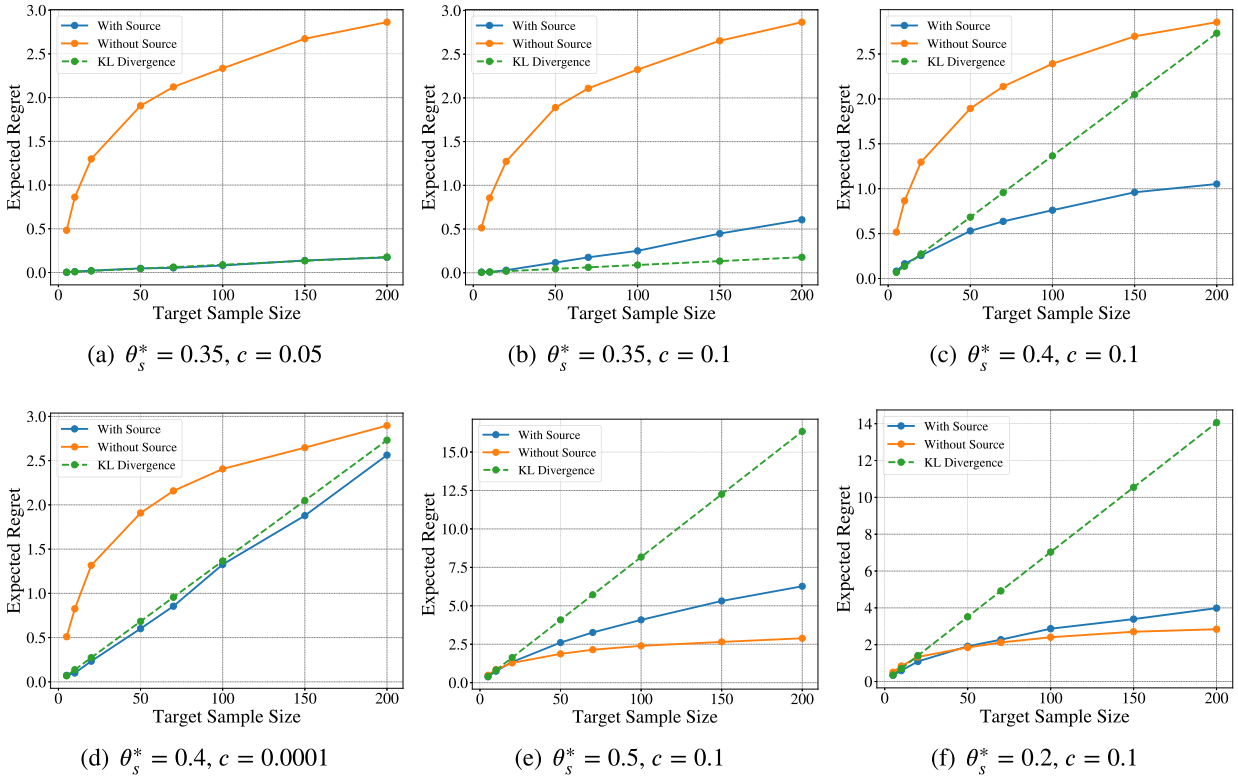


Fig. 3. Expected regrets by 2000 repeats with different c and θ_s^*. Orange, blue and green curves represents the expected regrets without the source, with the source, and $kD_{\mathrm{KL}}(P_{\theta_t^*} \| P_{\theta_s^*})$, respectively.

more concentrated density around θ_t^*. However, as shown in Fig. 2(f), the estimation of the θ_t^* does not converge to its true value with an improper prior, which fits in line with our intuition.

To evaluate the expected regrets, we repeat the experiments 2000 times and take the average for different numbers of target samples. The results are shown in Fig. 3. The regret curves reflect the influence of the knowledge level c and the gaps between the true parameter θ_s^* and θ_t^*. When $\omega(\theta_t|\theta_s^*)$ is proper, e.g., the density is concentrated around θ_t^*, then a smaller c and a smaller gap $|\theta_t^* - \theta_s^*|$ will yield a lower regret, which can be seen from Fig. 3(a), 3(b) and 3(c). However, if the conditional prior is improper (e.g., $[\theta_s^* - c, \theta_s^* + c]$ does not cover θ_t^*), the regrets are determined by both c and the distance $|\theta_s^* - \theta_c^*|$. For example, comparing the case $\theta_s^* = 0.4, c = 0.1$ (Fig. 3(c)) with $\theta_s^* = 0.4, c = 0.0001$ (Fig. 3(d)), the former case is the proper, while the latter does not cover the true θ_t^*, so the worse regrets. In addition, if c is small enough ($c = 0.001$), the estimation of θ_t will be centered at θ_s^* thus the regrets will coincide with the KL divergence $kD(P_{\theta_t^*} \| P_{\theta_s^*})$, which confirms the negative transfer case as discussed before. Comparing the case $\theta_s^* = 0.5, c = 0.1$ (Fig. 3(e)) with $\theta_s^* = 0.2, c = 0.1$ (Fig. 3(f)), even though both cases are under the improper priors, the latter yields a lower regret as the true source parameter is more closer to the true target parameter. Overall, once the prior information $\omega(\theta_t|\theta_s)$ is located around θ_t^*, and the target samples are inadequate to make an accurate prediction, the knowledge transfer is sensible, and indeed the regret can be further optimized.

4. Algorithms and experiments

The mixture strategy requires the calculation of the posterior of Θ_s and Θ_t given the source and target data. This computation generally has a high complexity, especially when the model parameter space Λ lies in a high dimensional space. Since our main aim is to estimate the true parameters θ_t^* and θ_s^*, we may not need the full posterior. To this end, we propose an efficient algorithm for approximating the underlying parameters, called the efficient mixture posterior updating (EMPU) algorithm, and several variants are proposed for real-world applications. Furthermore, we propose an algorithm that works without the parametric assumption on the statistical model. Based on the Dirichlet process, this algorithm can be seen as a nonparametric version of the mixture strategy. We exhibit the experimental results for synthetic transfer problems and real-world learning problems. We show that our methods achieve both computational efficiency and high performance.

4.1. Efficient posterior updating

In this subsection, we introduce an efficient algorithm for parametric models to approximate the mixture strategy. To illustrate, we consider the OTL case as an example, which can be extended to ITL and TVTL cases straightforwardly. From an algorithmic perspective, the mixture strategy introduced in section 3.1 can be rewritten as the following steps in Algorithm 1. However, estimating the posterior of Θ_t is computational expensive in practice (line 4 and 9), especially when the parameter dimension d is relatively high. To illustrate, we consider a simplified updating rule assuming the parameter lies in $\mathcal{R}^d$ and at each dimension, and each parameter can take K different values. Then there are d^K combinations for the parameter set, and the computational cost grows polynomially with d. For the continuous random variable (as K goes to infinity), updating the posterior is usually infeasible when d is large.

Algorithm 1: Mixture Strategy in OTL.

input : $\mathcal{D}_S^m$, loss function ℓ, prior knowledge over θ_t and θ_s
1 Encode the prior knowledge as $\omega(\Theta_t, \Theta_s)$;
2 Calculate the posterior θ_s from $\mathcal{D}_S^m$, i.e., $P(\theta_s|D_s^m)$;
3 Initialize target dataset $D_t^{k-1} = []$;
4 Initialize the prior $P(\Theta_t|D_s^m)$ in Equation (18) with $\omega(\Theta_t, \Theta_s)$ and $P(\Theta_s|D_s^m)$;
5 **for** $k = 1, \cdots, T$ **do**
6 Receive target sample $Z_t^{(k)}$;
7 Make prediction for $Z_t^{(k)}$ with the posterior $P(\Theta_t|D_t^{k-1}, D_s^m)$ under loss ℓ;
8 Add $Z_t^{(k)}$ to D_t^{k-1};
9 Update the posterior $P(\Theta_t|D_t^k, D_s^m)$;
10 **end**
output: Sequential predictions for $Z_t^{(k)}$

Concerning the computational infeasibility issues, we propose an algorithm for an efficient posterior updating algorithm to make the mixture strategy amenable to faster implementation. The Efficient Mixture Posterior Updating (EMPU) algorithm is illustrated in Algorithm 2. Since estimating the full posterior is challenging as the sample increases, we discretize the support of Θ_t and propose a gradient-based mixture strategy for efficiently estimating the posterior with the prior knowledge. Precisely, we first learn the distribution parameter $\hat{\theta}_s$ from the $\mathcal{D}_S^m$ by statistical methods such as maximum a posterior or maximum likelihood estimation (line 2). Then we quantize the support of the parameter space Λ into N points with the prior knowledge $\omega(\Theta_t|\Theta_s)$ for posterior approximation. For example, we will sample $\theta_{t,i}, i = 1, 2, \cdots, N$ from Λ according to the prior distribution $\omega(\Theta_t|\hat{\theta}_s)$ given the learned $\hat{\theta}_s$ (line 3). Each $\theta_{t,i}$ will have a corresponding weight ω_i initialized by the conditional prior $\omega(\Theta_t|\hat{\theta}_s)$ (line 4). When we receive a new target sample Z_t, we will update θ_i based on the gradient descent with the step size η and loss function ℓ and also update ω_i using the exponential weighting strategy (line 8 and 9). After receiving n target samples, the updated $\theta_{t,i}$ and ω_i will be used for future predictions (depending on the tasks). In particular, when dealing with supervised learning problems such as classification and regression, we use a similar setup as in [81] and assume that each time t we will receive a feature-label pair $Z_t^k = (X_t^k, Y_t^k)$.

Algorithm 2: Efficient Mixture Posterior Updating in OTL.

input : $\mathcal{D}_S^m$, quantization number N, loss function ℓ, prior knowledge over θ_t and θ_s
1 Encode prior knowledge as distribution $\omega(\Theta_t, \Theta_s)$;
2 Estimate $\hat{\theta}_s$ from $\mathcal{D}_S^m$;
3 Randomly sample $\theta_{t,i}$ from $\omega(\theta_t|\hat{\theta}_s)$ for $i = 1, 2, \cdots, N$;
4 Initialize distribution ω_i with the prior knowledge ;
5 **for** $k = 1, \cdots, T$ **do**
6 Receive target sample $Z_t^{(k)}$;
7 **for** $i = 1, \cdots, N$ **do**
8 Update $\theta_{t,i}$ by $\theta_{t,i}(k+1) = \text{Proj}(\theta_{t,i}(k) - \eta \nabla_{\theta_t} \ell(\theta_{t,i}(k), Z_t^{(k)}))$;
9 Update ω_i by $\omega_i(k+1) = \omega_i(k) e^{-\ell(\theta_{t,i}(k+1), Z_t^{(k)})}$;
10 Normalize $\omega_i(k+1) = \frac{\omega_i(k+1)}{\sum_{i=1}^N \omega_i(k+1)}$
11 **end**
12 **end**
output: ω_i and $\theta_{t,i}$

Remark 7. Instead of applying the Bayes rule, we use the gradient descent-based method to update $\theta_{t,i}$ and apply the exponential weighting strategy for the weights ω_i. It should be noted that EMPU is an approximation of the posterior distributions, and hence the theoretical analysis in Theorem 11 and 12 does not apply to this algorithm. However, numerical results show that the learning performance by EMPU is similar to the mixture strategy suggested in Algorithm 1. In contrast

to evaluating the full posterior, the learning performance will depend on the hyperparameters, e.g., the quantization number N and the step size η.

4.2. Logistic regression example

We consider a logistic regression problem in a 2-dimensional space to compare Algorithm 1 the EMPU algorithm (Algorithm 2) in the OTL scenario. For the given parameter $\theta \in [0,1]^2$ and $Z_i = (X_i, Y_i) \in \mathbb{R}^2 \times \{0,1\}$, each label $Y_i \in \{0,1\}$ is generated from the Bernoulli distribution with probability $P(Y_i = 1|X_i) = \frac{1}{1+e^{-\theta^T X_i}}$. Suppose that the source and target input features $X_s^{(k)}$ and $X_t^{(k)}$ are drawn from the same normal distribution $\mathcal{N}(\begin{bmatrix} 5 \\ -5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$. The loss function is then given by

$$\ell(\theta, Z_i) := -(Y_i \log(\sigma(\theta^T X_i)) + (1 - Y_i)\log(1 - \sigma(\theta^T X_i))), \tag{79}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$. Let $\theta_t^* = (0.3, 0.5)$ and $\theta_s^* = (0.2, 0.4)$ denote the true parameters for the target and source domains. Given $m = 5000$, let the marginal prior $\omega(\Theta_s)$ be uniformly distributed over $[0,1]^2$ and our prior knowledge $\omega(\Theta_t|\Theta_s)$ assumes that Θ_t is normally distributed with the mean of Θ_s and covariance of $\begin{bmatrix} c^2 & 0 \\ 0 & c^2 \end{bmatrix}$, here c represents the prior belief on Θ_t such that smaller c implies Θ_t is closer to Θ_s and vice versa. To show the usefulness of the source data, we compare with the target only case ($m = 0$) where we assume the prior $\hat{\omega}(\Theta_t)$ is uniformly distributed over $[0,1]^2$.

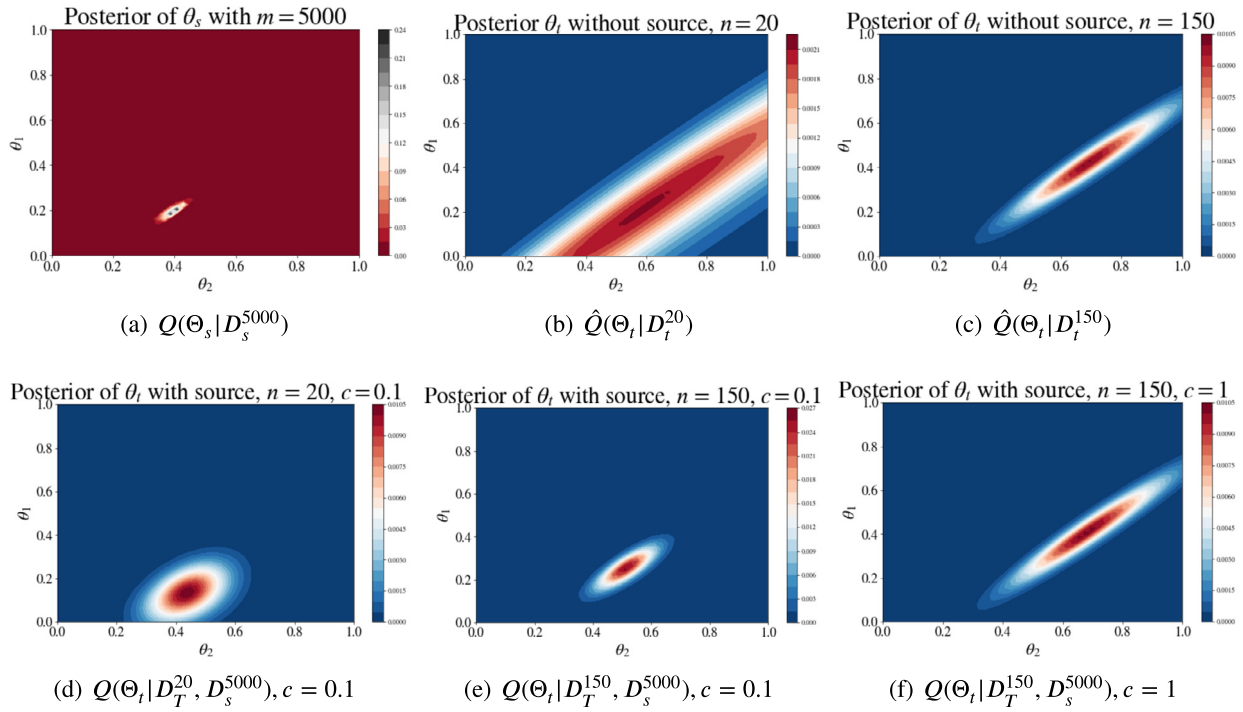
Full posterior Let $Q(\Theta_s|D_s^m)$ denote the posterior of Θ_s induced by the (marginal) prior $\omega(\Theta_s)$ and $Q(\Theta_t|D_s^m, D_t^n)$ denote the posterior of Θ_t induced by the prior $\omega(\Theta_t, \Theta_s)$ for transfer scenarios, let $\hat{Q}(\Theta_t|D_t^n)$ denote the posterior induced by the prior $\hat{\omega}(\Theta_t)$ without the source data. After receiving n target samples, we plot different posteriors to see the effect of the mixture strategy induced by the chosen prior in Fig. 4. From 4(a), given sufficient source data, the posterior of Θ_s will give a precise estimation of θ_s^* and the density will mostly stick around $[0.2, 0.4]$. While there is a lack of target samples ($n = 20$), the posterior $\hat{Q}(\Theta_t|D_t^n)$ (Fig. 4(b)) without the source is relatively scattered and the density around θ_t^* is quite low. When n increases to 150 (Fig. 4(c)), the posterior is more concentrated but still not centered at θ_t^*, implying that more target data are needed for accurate estimation. On the contrary, with the prior knowledge $\omega(\Theta_t|\Theta_s)$ and small $c = 0.1$, the posterior $Q(\Theta_t|D_s^m, D_t^n)$ will be concentrated more around θ_t^* as source and target parameters are particularly close, which can be seen in Fig. 4(d) and 4(e). However, when c increases to 1, the source data is no longer helpful as Θ_t is roughly distributed uniformly on $[0,1]^2$ and the posterior behaves similarly to the target only case as shown in Fig. 4(f).

To further demonstrate our theoretical results, we plot the expected regrets in Fig. 5 for positive and negative transfer cases, and we also plot the asymptotic estimation of CMI in dashed lines from Theorem 4 and 12 to numerically evaluate the difference. From the left figure, it is observed that introducing the source indeed yields lower regret, which fits our intuition from the posteriors. Even for small n (≈ 40), CMI captures the regret quite well and the gap is roughly $\log \frac{\omega(\theta_t^*|\theta_s^*)}{\omega(\theta_t^*)}$ as noted in Remark 4. In contrast, we also examine the negative transfer case with $\theta_s^* = [0.8, 0.15]$ where the results are shown in the right figure. In this case, the prior distribution $\omega(\theta_c^*|\theta_s^*)$ has an extremely low density, and the estimation will hardly approach the true parameters. As a result, the negative transfer happens, and source samples will hurt the performance instead. The expected regret with source data is far superior to the target-only case, and CMI captures this trend well when n goes reasonably large (≈ 80).

Overall, from both positive and negative transfer cases, the gaps between the regrets are mainly reflected on the prior knowledge $\omega(\theta_t^*|\theta_s^*)$ when n is reasonably large as mentioned in Remark 4 and 5, which experimentally confirms Theorem 12. Moreover, even though our asymptotic results are derived under large sample sizes, Fig. 5 shows that the asymptotic bounds can also be applied to the case when the sample sizes n and m are limited and may provide some practical insights on avoiding the negative transfer.

Efficient mixture posterior updating When estimating the full posterior for θ_t^* and θ_s^*, it is time-consuming when data sizes are large. To examine the efficiency and usefulness of the proposed EMPU algorithms, we set the quantization number $N = 100$ and step size $\eta = 0.01$ to conduct the iterations on the identical logistic regression transfer problem settings for comparisons. The results are shown in Fig. 6.

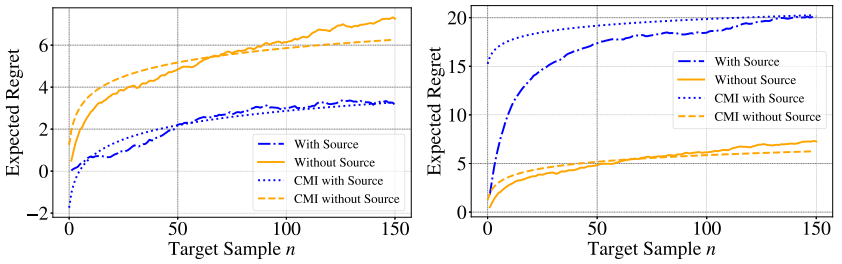
From the figure, it is easy to see that EMPU achieves similar regrets induced by the full posterior estimation and our derived bounds under both positive and negative transfer cases. As noted, the performance of EMPU depends on the hyperparameters of the step size η and the quantization number N. We investigate the effects of these hyperparameters and plot the expected regret by varying η and N after receiving 150 target samples under the negative transfer case, the non-transfer case (without the source data), and the positive transfer case. From Fig. 7(a), it is observed that in non-transfer and positive transfer cases, the lowest expected regrets are achieved at $\eta = 0.01$ while in the negative transfer case, the best performance is achieved at $\eta = 0.001$. It is speculated that under the negative transfer case, the estimation of θ_t will never approach θ_t^* with the improper prior, then it is more likely to achieve the optima $\tilde{\theta}_t$ defined in Proposition 2 faster within its support and plausibly jump over it with a larger step size, which incurs a higher regret. The effects of quantization number N are illustrated in Fig. 7(b). In all three cases, increasing the quantization number will decrease the regret when



(a) $Q(\Theta_s|D_s^{5000})$ (b) $\hat{Q}(\Theta_t|D_t^{20})$ (c) $\hat{Q}(\Theta_t|D_t^{150})$

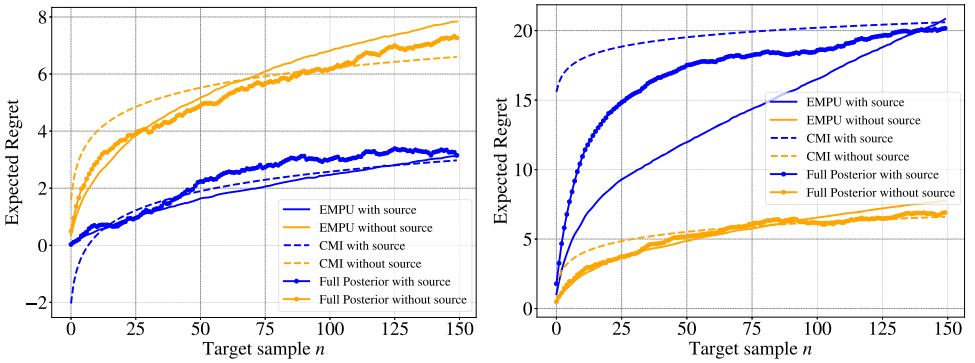
(d) $Q(\Theta_t|D_T^{20}, D_s^{5000}), c = 0.1$ (e) $Q(\Theta_t|D_T^{150}, D_s^{5000}), c = 0.1$ (f) $Q(\Theta_t|D_T^{150}, D_s^{5000}), c = 1$

Fig. 4. The posterior of θ_s and θ_t given D_s^m and D_t^n under different prior belief c and target sample size n. The posterior of Θ_t is more concentrated around θ_t^* with the source data introduced.



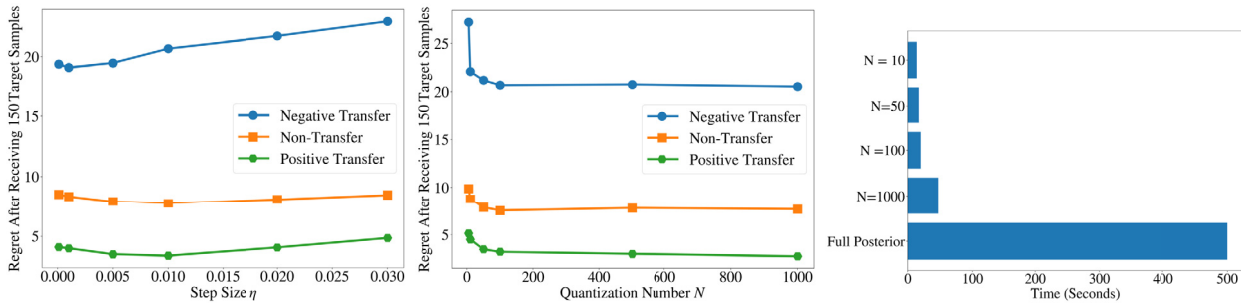
(a) Positive Transfer (b) Negative Transfer

Fig. 5. The comparisons of the expected regret $\mathcal{R}_O$ of positive transfer with $\theta_s^* = [0.2, 0.4]$ (left) and the negative transfer (right) with $\theta_s^* = [0.8, 0.15]$ under the same settings where $\theta_t^* = [0.3, 0.5]$ and $c = 0.1$. The results are approximated by 200 experimental repeats. The regrets without the source data are sketched in orange and those with the source data are sketched in blue.



(a) Positive Transfer (b) Negative Transfer

Fig. 6. Comparisons of full posterior and EMPU algorithms under the positive and negative transfer scenarios. The results are approximated by 200 experimental repeats. The regrets without the source data are sketched in orange and those with the source data are sketched in blue.



(a) Effect of η (b) Effect of N (c) Time Comparisons

Fig. 7. After receiving 150 target samples, we plot the expected regret by varying different η and N in (a) and (b). We also compare the running time under different quantization numbers N and the full posterior algorithm. The results are approximated by 200 experimental repeats.

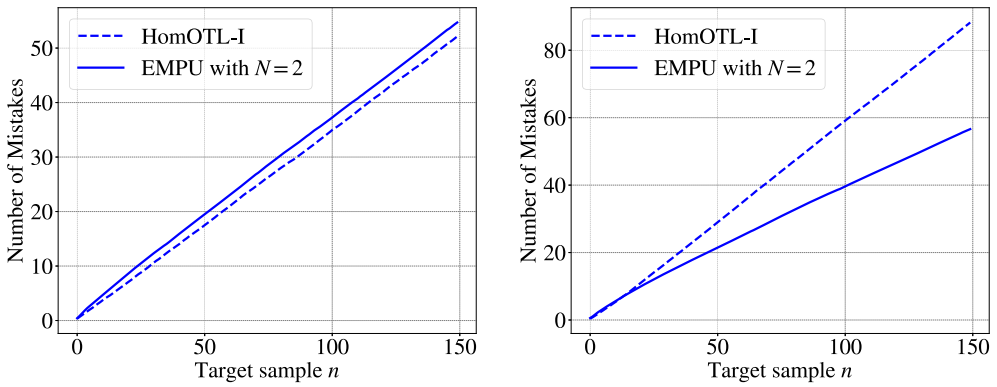
N is small (e.g., $N < 100$). Nevertheless, no significant changes are spotted when N increases to more than 100. Since larger N leads to higher computational complexity, the best choice for this quantity is $N = 100$. Furthermore, we plot the running time by varying N from 10 to 1000. Compared to the mixture strategy algorithm with the full posterior, EMPU is 20 times faster under $N = 100$ but achieves similar regret, demonstrating its efficiency.

Comparisons to other OTL algorithms We compare our proposed EMPU with the online transfer learning algorithm proposed in [81], the HomOTL-I algorithm. In their scheme, the authors assign the weights ω_s and ω_t for source and target domains and each domain is endowed with the model parameters θ_s and θ_t. At each time k, the weights and the model parameters are updated with the following rules:

$$\begin{cases} \omega_s^k = \dfrac{\omega_s^{k-1} g\left(\ell^*\left(\theta_s, Z_t^k\right)\right)}{\omega_t^{k-1} g\left(\ell^*(\theta_t^{k-1}, Z_t^k\right) + \omega_t^{k-1} g\left(\ell(\theta_s, Z_t^k)\right)}, \omega_s^0 = 0 \\ \omega_t^k = \dfrac{\omega_t^{k-1} g\left(\ell^*\left(\theta_t, Z_t^k\right)\right)}{\omega_t^{k-1} g\left(\ell^*(\theta_t^{k-1}, Z_t^k\right) + \omega_t^{k-1} g\left(\ell(\theta_s, Z_t^k)\right)}, \omega_t^0 = 0 \\ \theta_t^k = \theta_t^{k-1} - \eta \nabla_{\theta_t} \ell(\theta_t^{k-1}, Z_t^k) \end{cases} \tag{80}$$

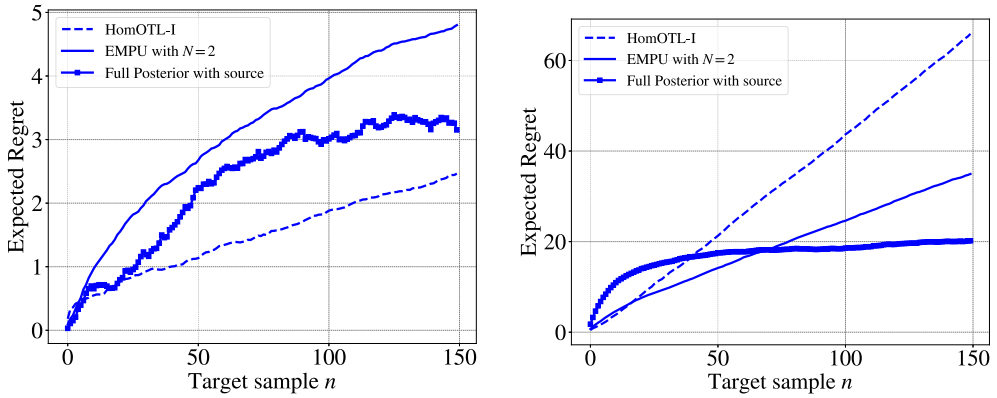
where g is some decaying function, ℓ is the hinge loss, and ℓ^* is the squared loss. The predictions are made from the learned model parameters θ_s and θ_t along with the weights ω_s and ω_t by a truncated linear model. Similar updating rules are proposed in [65,71] with different choices of g. The EMPU algorithm is closely related to this framework. To illustrate, if we regard the distribution parameters Θ_s and Θ_t as the model parameters and choose the decaying function g as the exponential weighting function, e.g., $g(x) = e^{-x}$, and let both ℓ and ℓ^* be the cross-entropy loss, then the above scheme will practically coincide with our proposed method if we choose $N = 2$. The main differences are that θ_t will be updated and θ_s is kept fixed at each iteration in their scheme, and our model is not limited to linear models, and the loss function is not limited to hinge loss. As a consequence, there is no prior knowledge of the target domain, and the performance at the beginning will rely heavily on the source domain since θ_s remains unchanged all the time; thus may perform badly if two domains are distinct. While in our scheme, the source parameter Θ_s behaves as the prior knowledge for the target parameter, which is not explicitly engaged in the prediction. Thus we can choose a proper prior at the beginning to either avoid the negative transfer if the source domain varies differently from the target domain or improve the prediction if the two domains are close.

Using the same settings of the logistic regression problems, we study the *small discrepancy* case where the target and source data are generated with $\theta_t^* = [0.3, 0.5]$ and $\theta_s^* = [0.2, 0.4]$ and the *large discrepancy* case where $\theta_t^* = [0.3, 0.5]$ and $\theta_s^* = [0.8, 0.15]$. In our method, we will set the quantization number $N = 2$ and the prior knowledge $c = 0.3$ to have proper priors under both cases. Then at each iteration k we predict the label Y_k by first sampling $\theta_{k,i}$ according to the distribution ω_k, the predicted $\hat{Y}_k$ will be 1 if the probability $\sigma(\theta_{k,i}^T X_i) > 0.5$ and 0 otherwise. In HomOTL-I, we firstly learn $\hat{\theta}_s$ by the linear regression method and initialize $\theta_{t,0}$ uniformly randomly in $[0, 1]^2$. Then at each iteration k we conduct the HomOTL-I to predict $\hat{Y}_k$. We will plot the accumulated number of mistakes $\sum_{i=1}^T |\hat{Y}_k - Y_k|$ by averaging 200 experimental repeats. We choose the step size $\eta = 0.01$ for both algorithms. The comparisons can be found in Fig. 8. From the small discrepancy case, HomOTL-I indeed performs slightly better than the EMPU algorithm since the initially learned $\hat{\theta}_s$ is very helpful for prediction in the target domain as $P_{\theta_s^*}(Y|X)$ and $P_{\theta_t^*}(Y|X)$ are relatively close in terms of the parameterization, which leads to a smaller number of mistakes. In the large discrepancy case, the domain divergence becomes relatively large, and the initially learned $\hat{\theta}_s$ is more of a hindrance which causes a larger number of mistakes in the target domain. The EMPU



(a) Small discrepancy (b) Large discrepancy

Fig. 8. Comparisons of EMPU with $N=2$ and HomOTL-I method [81] by the average number of mistakes made at each iteration. We set the step size η to be 0.01 for all experiments. The results are approximated by 200 experimental repeats.



(a) Small discrepancy (b) Large discrepancy

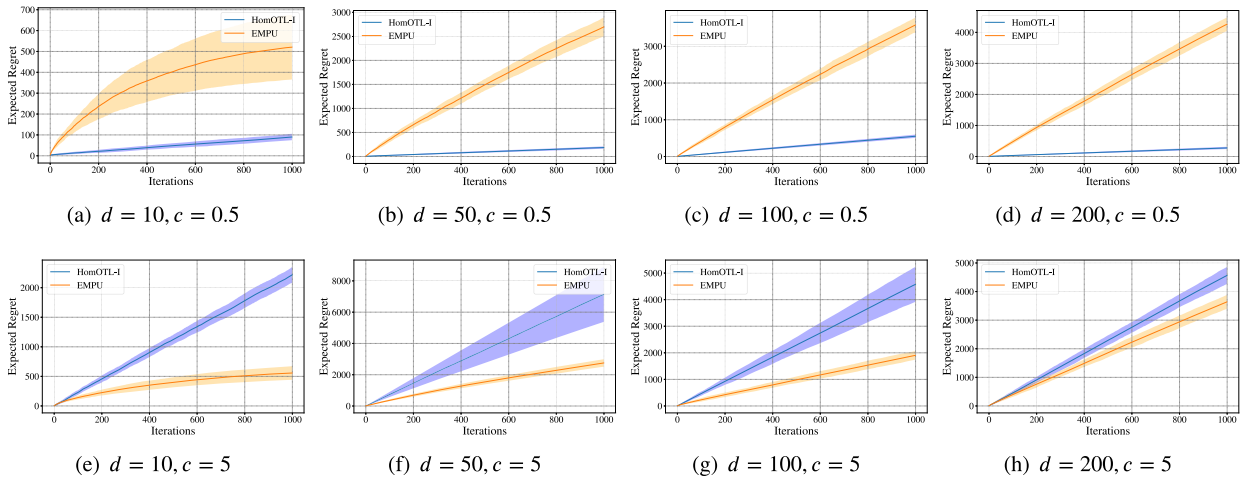
Fig. 9. Comparisons of EMPU with $N=2$ and HomOTL-I [81] by the expected regret. We set the step size η to be 0.01 for all experiments. The results are approximated by 200 experimental repeats.

algorithm performs reasonably well in both cases, showing that updating with prior knowledge is particularly beneficial for achieving a lower regret when two domains are far different.

Likewise, we adapt the HomOTL-I algorithm to our framework by choosing both ℓ and ℓ^* to be the cross-entropy loss defined in Equation (79), and we use the logistic regression as our predictive model. To elaborate, we first learn $\hat{\theta}_s$ by the logistic regression method and initialize $\theta_{t,0}$ uniform randomly in Λ. The comparisons can be found in Fig. 9 under the small and the large discrepancy scenarios by the expected regret. We observe similar patterns in both scenarios. HomOTL-I indeed performs better than EMPU and the mixture strategy with full posterior since the estimated $\hat{\theta}_s$ is already very close to the true target parameters $\theta_t^* = [0.3, 0.5]$ at the beginning of the prediction, thus a lower regret. However, in the negative transfer case, the initialized $\hat{\theta}_s$ is located fairly far away from θ_t^*, HomOTL-I will mainly rely on the bad-performing $\hat{\theta}_s$ and induce a much higher expected regret.

4.2.1. High dimensional example

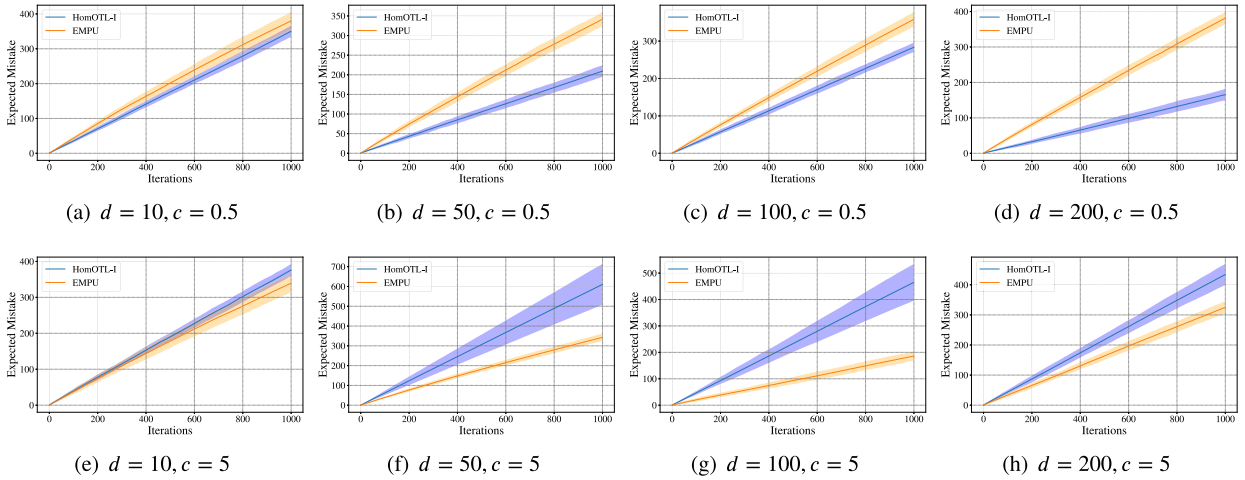
In this section, we evaluate our method when the parameter dimension d becomes large to show the effectiveness of the EMPU algorithm, whereas the full posterior updating is infeasible. For the experimental setup, we vary the parameter and input data dimension from $d = 10$ to 200 for generating the synthetic data under the logistic regression model described before. Specifically, we generate θ_t^* by randomly drawing a sample from a uniform distribution $U([0,1]^d)$. Then we generate $\theta_s^* = \theta_t^* + c * \epsilon$ where ϵ is drawn from a uniform distribution $U([-1,1]^d)$ and c controls the distance between the source and target parameters. For both source and target domains, we draw the features X from a normal distribution $\mathcal{N}(\mu, \sigma^2 I_d)$ where μ is randomly from a uniform distribution $U([-3,3]^d)$ and variance σ^2 is to be 4. Then the corresponding Y label is produced as per the Bernoulli distributions $\text{Ber}(\sigma(X^T\theta_s^*))$ and $\text{Ber}(\sigma(X^T\theta_t^*))$ for the source and target domains, respectively. Next we generate the source sample as described above with the sample size $m = 100000$, and we use the logistic regression method to estimate the true parameter $\hat{\theta}_s^*$ for further prediction. We will then sequentially receive $n = 1000$ target data for prediction. As calculating the full posterior is impractical when d is large, we only compare the OTL algorithm with the



(a) $d = 10, c = 0.5$ (b) $d = 50, c = 0.5$ (c) $d = 100, c = 0.5$ (d) $d = 200, c = 0.5$

(e) $d = 10, c = 5$ (f) $d = 50, c = 5$ (g) $d = 100, c = 5$ (h) $d = 200, c = 5$

Fig. 10. The results on the expected regret of the sequential target under different d and c generated by 50 experimental repeats. The solid line shows the expected regret of the HomOTL-I (blue) and EMPU (orange) algorithms. The shaded areas represent the standard deviation from their mean value.



(a) $d = 10, c = 0.5$ (b) $d = 50, c = 0.5$ (c) $d = 100, c = 0.5$ (d) $d = 200, c = 0.5$

(e) $d = 10, c = 5$ (f) $d = 50, c = 5$ (g) $d = 100, c = 5$ (h) $d = 200, c = 5$

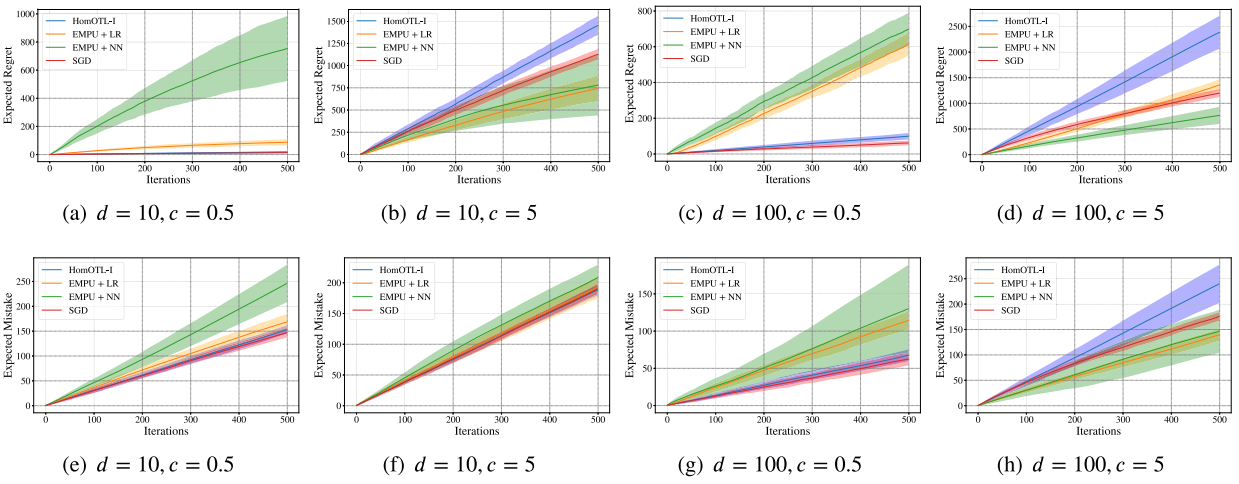
Fig. 11. The results on the expected mistakes of the sequential target under different d and c generated by 50 experimental repeats. The solid line shows the expected mistakes of the HomOTL-I (blue) and EMPU (orange) algorithms. The shaded area represents the standard deviation from the mean value.

proposed EMPU algorithm in terms of the expected regret and the expected loss. Specifically, for the OTL algorithm θ_t is initially drawn from the normal distribution $\mathcal{N}(\hat{\theta}_s^*, c\mathbf{I}_d)$. The EMPU algorithm is conducted with the quantization $N = 100$ and each $\theta_{t,i}$ is generated from the normal distribution $\mathcal{N}(\hat{\theta}_s^*, c\mathbf{I}_d)$. By setting the learning rate $\eta = 0.005$ for both methods, we examine different dimensions d and gap coefficients c and sketch the results in Fig. 10 and 11.

As shown in Figs. 10 and 11, we observe similar patterns on the expected regrets and mistakes for different dimensions d and the gap coefficients c. When the gap is small ($c = 0.5$), HomoOTL-I will outperform EMPU as the initial source weights already give a near-optimal estimation. However, for a large gap ($c = 5$), EMPU will outperform OTL in this case when the source parameter lies far away from the target ones. We could also notice that as the dimension d increases, the estimation of the target parameter becomes more computationally expensive, and the expected regret and mistakes become larger.

4.2.2. Neural network example

The EMPU algorithm can be applied to more sophisticated models such as neural networks, which are among the most popular and widely used models in machine learning. Additionally, the assumption of parametric distribution could be weakened by the neural network, i.e., the underlying data-generating distribution may not necessarily lie in the hypothesis space or be in a parametric family. That is, we can use the neural network for approximating data underlying distributions due to its strong representation capability, allowing a broader application in the real world. Due to the **model misspecification** that the true data distribution does not lie in the parametric family used in prediction, the KL divergence gap, however, naturally exists during the estimation process. To illustrate, we take the OTL (similar to ITL) as an example. Assume that the target data are drawn from an arbitrary distribution p_t, but we still restrict our mixture strategy within a



(a) $d = 10, c = 0.5$ (b) $d = 10, c = 5$ (c) $d = 100, c = 0.5$ (d) $d = 100, c = 5$

(e) $d = 10, c = 0.5$ (f) $d = 10, c = 5$ (g) $d = 100, c = 0.5$ (h) $d = 100, c = 5$

Fig. 12. The results on the expected mistakes and regrets of the sequential target data under different feature and parameter dimensions d and c are generated by 50 experimental repeats under the misspecified setting. The first row shows the expected mistake of the HomOTL-I (blue), EMPU with the logistic regression (orange), EMPU with the neural network (green), and the baseline stochastic gradient descent (red) algorithms. The second row shows the expected mistake as per the setting in the first row. The shaded area represents the standard deviation from the mean value.

certain parametric family p_θ, e.g., the neural network. Then by denoting the optimal distribution within the parametric family $p_{\theta^*} = \text{argmin}_{p_\theta} D_{\text{KL}}(p_t \| p_\theta)$, we can then decompose the expected regret by,

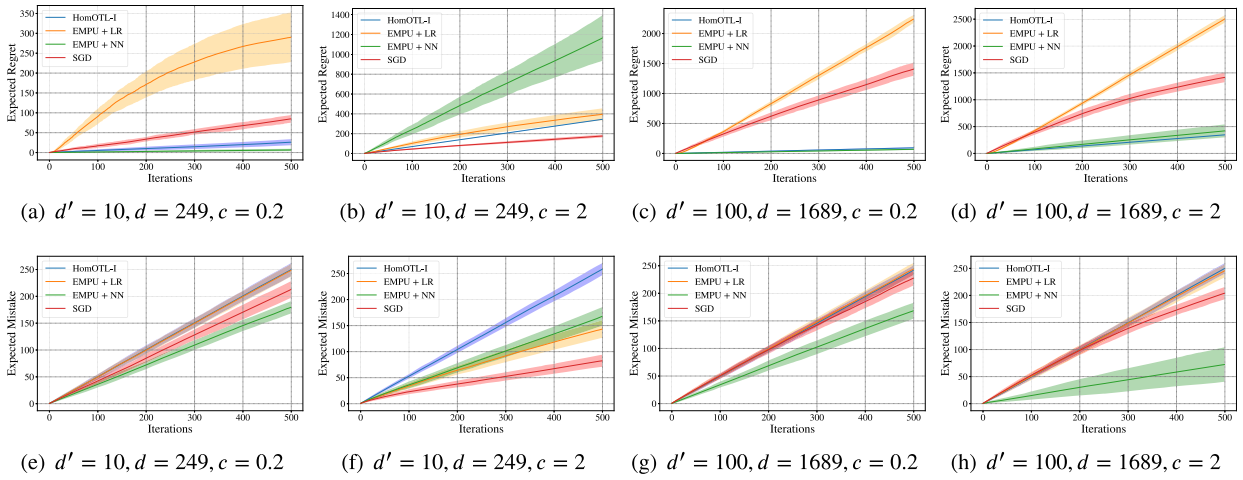
$$\mathcal{R}_O = \mathbb{E}_{p_t}\left[\log \frac{p_t(D_t^n)}{p_{\theta^*}(D_t^n)}\right] + \mathbb{E}_{p_t}\left[\log \frac{p_{\theta^*}(D_t^n)}{Q(D_t^n | D_s^m)}\right]. \tag{81}$$

From the R.H.S. of (81), the best predictor that the mixture strategy can achieve is p_{θ^*} and as the sample size increases, the second term will converge to zero under the proper prior. However, the first term is the KL divergence between p_t and p_{θ^*} that only depends on the hypothesis space and does not vanish even with sufficient data. Therefore, there is a trade-off between the model selection and the learning rate. A complex model might provide a small KL divergence term, but the second term will be large since convergence slows down as d becomes large. A simple model, however, results in a large KL divergence term and a small second term with a small d. This trade-off naturally brings up how to select an appropriate model for prediction. To illustrate the effect of the model selection, we examine the EMPU algorithm with the neural network under both the **misspecified** model and **correct** model.

Firstly we conduct the experiments under the misspecified setting. For experimental setups, we generate the source and target data in the same way as in Section 4.2.1 with different sets of input data dimensions and parameter dimensions d and gap coefficients c. The misspecified EMPU algorithm utilizes a two-layer perception model as the hypothesis space with 16 and 4 hidden units, the Relu activation function, and one dimension output with the sigmoid function. The conditional distribution $p_\theta(Y|X)$ is characterized by the neural network, and θ represents the weights for all layers. In terms of the model training, we first train a source network from the source data whose parameters are denoted by $\hat{\theta}_s^*$. Then we generate $N = 10$ target models where each model parameter is the noisy version of the estimated source parameter $\theta_{t,i} = \hat{\theta}_s^* + c' * \epsilon_i$ where $\epsilon \sim \mathcal{N}(0, 1)$. Our benchmarks are the correct EMPU model under the logistic regression model with $N = 100$ and OTL algorithm and the stochastic gradient descent algorithm using a logit model with the logarithmic loss. The learning rate for all algorithms is set to be 0.005, and we use the binary cross entropy loss function. The expected regret and mistakes results are shown in Fig. 12.

From the figure, we can observe that when the true parameter dimension d and distance c are small, the EMPU with the logistic regression model outperforms that with the neural network, and this is possibly due to a large number of parameters and the estimation of the weights in the neural networks is usually biased due to the nonlinearity from the activation layers. When c becomes large, the EMPU algorithm performs similarly well under both logistic regression and neural networks because the parameters in the source and target domains differ significantly, and the estimation does not heavily lie on the source parameter estimation. Despite this, the correct model performs better than the misspecified one. The performance of OTL and SGD depends heavily on the source domain parameters: a closer match between source and target domain parameters results in better performance and vice versa. As d increases to 100, there is a smaller performance gap between the correct and misspecified models as parameter estimation becomes more challenging in this case. With a large $c = 5$, the EMPU with the neural network can even perform better than that with the logistic regression in terms of the expected regret. As a result of over-parameterization, the neural network becomes easier to converge to a sub-optimal solution for the high-dimensional data, thus making it more suitable for real-world problems.

We also examine the case where the data are generated from the neural network, i.e., the correct model for the EMPU with the neural network and the misspecified model for the EMPU with the logistic regression. In this case, the dimensions



(a) $d' = 10, d = 249, c = 0.2$ (b) $d' = 10, d = 249, c = 2$ (c) $d' = 100, d = 1689, c = 0.2$ (d) $d' = 100, d = 1689, c = 2$

(e) $d' = 10, d = 249, c = 0.2$ (f) $d' = 10, d = 249, c = 2$ (g) $d' = 100, d = 1689, c = 0.2$ (h) $d' = 100, d = 1689, c = 2$

Fig. 13. The results on the expected mistakes and regrets of the sequential target data under different feature dimensions d', parameter dimension d and c are generated by 50 experimental repeats under the in-specified setting. The first row shows the expected mistake of the HomOTL-I (blue), EMPU with the logistic regression (orange), EMPU with the neural network (green) and the baseline stochastic gradient descent (red) algorithms. The second row shows the expected mistake as per the setting in the first row. The shaded area represents the standard deviation from the mean value.

of the input feature and parameters are no longer identical, and we denote the latter as d'. Now we generated the features X from a normal distribution $\mathcal{N}(\mu, \sigma^2 I_{d'})$ where μ is randomly from a uniform distribution $U([-3,3]^{d'})$ and variance σ^2 is to be 4. Denote the neural network by f_θ characterized by the model parameter θ, the features are fed into a two-layer perception model with 16 and 4 hidden units to generate the corresponding labels, e.g., $Y_i \sim \text{Ber}(f_\theta(X_i))$. We randomly sample the parameter θ_s^* from Kaiming initialization [24] and the true target parameter is generated by $\theta_t^* = \theta_s^* + c * \epsilon$ where ϵ is drawn from a standard normal distribution. With $d' = 10, 100$ (corresponding to $d = 249$ and 1689) and $c = 0.2, 2$, we then conduct the experiments using the same setting as in the misspecified case and the results are shown in Fig. 13.

As opposed to the previous situation, we can observe that with the correct model, the EMPU with the neural networks outperforms the other benchmarks when d' and c are both small. However, considering the case when $c = 2$ and $d' = 10$, the neural network has a large expected regret, and the expected mistake is even worse than the EMPU with the logistic regression model. This is possibly due to that a large gap c may cause the neural network to converge slowly with a relatively large number of parameters, resulting in poor predictions. The other three benchmarks, have fewer parameters (d' = 10) and are therefore able to reach their optimal solutions more quickly. Under the case of the high dimensional data (d' = 100), the advantage of the neural network becomes apparent, and with the correct model, it outperforms all the other three benchmarks.

By comparing these two scenarios, we can gain a better understanding of the model selection perspective. To highlight the impact of model selection in real-world problems, we validate the proposed algorithms against real datasets for comparisons in Section 4.4.

4.3. Gibbs EMPU

Previous theoretical results are derived under the assumption that the data distributions are parametric, say, $Z \sim P_\theta(Z)$. Such an assumption of the parametric model may not always hold in practical machine learning problems such as image processing and sentiment analysis. In the previous section, we saw that EMPU could be applied to different models, but we may need to assume that the source parameter is within a certain distance from the target parameters. Alternatively, we propose a general framework inspired by the Gibbs algorithm [78,69] and the mixture strategy, allowing for the inclusion of more general models and prior information, namely, the Gibbs EMPU. We present the detailed procedures in Algorithm 3.

The algorithm is conducted in the following way. Given a general learning model characterized by some parameters $\theta \in \Theta$, we first learn the source parameter $\hat{\theta}_s$ from the source data D_s^m, and we have the prior knowledge $\omega(\theta_t|\hat{\theta}_s)$ of the target model parameters given the estimated source one, which may not be limited to the constraints on the parameter distances. Then for each time k, for each incoming target instance, we will sample N different target parameters based on the prior knowledge and the Gibbs posterior distribution, which is given by

$$P(\theta_t|D_t^k, \hat{\theta}_s) = \frac{\omega(\theta_t|\hat{\theta}_s)e^{-\gamma\sum_{j=1}^k \ell(b(\theta_t), Z_t^{(j)})}}{\int \omega(\theta_t|\hat{\theta}_s)e^{-\gamma\sum_{j=1}^k \ell(b(\theta_t), Z_t^{(j)})}d\theta_t}$$

where $\gamma > 0$ is the hyperparameter that controls the belief over the data, and a larger γ leads to a smaller effect of prior knowledge. Note that different from the original mixture strategy, we replace the underlying data distribution $P_\theta(D_t^k)$ with

Algorithm 3: Gibbs EMPU Framework in OTL.

input : $\mathcal{D}_S^m$, quantization number N, loss function ℓ, prior knowledge over θ_t and θ_s, Gibbs parameter γ

1 Encode prior knowledge as distribution $\omega(\Theta_t, \Theta_s)$;
2 Estimate $\hat{\theta}_s$ from $\mathcal{D}_S^m$;
3 **for** $k = 1, \cdots, T$ **do**
4 **for** $i = 1, \cdots, N$ **do**
5 Sample ϵ_i from perturbation distribution and then calculate

$$\theta_{t,i} = \text{argmax}_{\theta_t} \log \omega(\theta_t|\hat{\theta}_s) e^{-\gamma \sum_{j=1}^{k-1} \ell(b(\theta_t+\epsilon_i), Z_t^{(j)})} \tag{82}$$

6 **end**
7 Predict $\hat{Z}_t^{(k)}$ with $\frac{1}{N}\sum_{i=1}^{N} b(\theta_{t,i})$;
8 Receive target sample $Z_t^{(k)}$;
9 Record history D_t^k;
10 **end**

output: $\theta_{t,i}$

the loss function dependent term $e^{-\gamma \sum_{j=1}^{k} \ell(b(\theta_t), Z_t^{(j)})}$ to ease the constraint on the parametric condition. For a continuous parameter space Λ, directly sampling $\theta_{t,i}$ from the Gibbs distribution $P(\theta_t|D_t^k, \hat{\theta}_s)$ is non-trivial. Instead, we draw a set of target parameters $\theta_{t,i}$ using the maximum a posterior (MAP) estimation as shown in Line 5 in Algorithm 3. Here we draw a set of perturbations $\epsilon_i \sim \mathcal{N}(0, \sigma_p^2)$ to heuristically model the randomness of the sampling process, where σ_p is chosen appropriately according to the parameter space Λ. Then the sampled model parameters $\theta_{t,i}$ are aggregated to predict the next instance, e.g., $\hat{Z}_t^{(k)} = \frac{1}{N}\sum_{i=1}^{N} b(\theta_{t,i})$. As k increases, it becomes increasingly difficult to sample the parameters and compute the summation of the loss in the posterior distribution each time. To resolve this issue, we could rewrite the equation (82) as

$$\theta_{t,i} = \text{argmin}_{\theta_t} \sum_{i=1}^{k-1} \ell(b(\theta_t + \epsilon_i), Z_t^{(i)}) - \frac{1}{\gamma} \log \omega(\theta_t|\hat{\theta}_s). \tag{83}$$

The minimization can be done adaptively in an online fashion via the optimization tools such as stochastic gradient descent if $\omega(\theta_t|\hat{\theta}_s)$ is concave and differentiable w.r.t. θ_t. Therefore, such an extension enables us to work with more general models without encoding the prior information regarding the distance between source and target parameters. In particular, if we set the prior knowledge to be the normal distribution:

$$\omega(\theta_t|\theta_s) \propto e^{-\|\theta_s - \theta_t\|_2^2}.$$

Then the equation (82) becomes,

$$\theta_{t,i} = \text{argmin}_{\theta_t} \sum_{i=1}^{k-1} \ell(b(\theta_t + \epsilon_i), Z_t^{(i)}) + \frac{1}{\gamma} \|\theta_s - \theta_t\|_2^2, \tag{84}$$

which will lead to the regularized ERM algorithm with the perturbation. To demonstrate the effectiveness of the Gibbs EMPU algorithm, we will conduct the Gibbs EMPU using (84) to several real-world transfer learning problems in Section 4.4.

4.4. Real dataset experiments

The proposed algorithms were evaluated only using the synthetic data in previous results. In this section, we will conduct the EMPU algorithms on several transfer learning tasks to show their effectiveness in real-world problems. Below is a list of the transfer learning datasets we have trained and validated.

- Office-Caltech-10 [18]: This dataset contains four subsets, and each domain has a set of office photos with the same 10 classes. In particular, the four subsets are **W**ebcam (W for short), **D**SLR (D for short), **A**mazon (A for short) and **C**altech dataset (C for short). We use each subset as a domain, and consequently, we can construct 12 transfer learning problems by choosing one domain as the source and another as the target. We use the SURF features as described in [18] encoded with a visual dictionary of 800 dimensions.
- 20 Newsgroups[2]: the dataset contains approximately 20000 reviews from 7 major categories that can be split into 20 subcategories. The source and target domains were picked from the same two major categories but different subcategories in each transfer learning task, in the same way as in [57]. The label for each instance will be its corresponding major categories, leading to a binary classification transfer problem.

[2] http://qwone.com/~jason/20Newsgroups/.

- MNIST and USPS: These two datasets contain black and white hand-written digits from 0 to 9, where MNIST[3] has approximately 70000 images, and USPS[4] has approximately 10000 images in total. Since each domain shares the same digits from 0 to 9 (but different writing styles), we, therefore, constructed two transfer learning tasks: 'MNIST to USPS' and 'USPS to MNIST' in the same way in [60] and report classification accuracy for each task. We use 2000 instances from the MNIST datasets and 1800 instances from the USPS datasets.
- GIST datasets [54]: The dataset is associated with the traffic scenario identification problem: an intersection was observed by a fixed traffic camera where the scenario frames were updated every 3 minutes, and the whole dataset includes over 5000 images captured over two weeks. There is a continuous domain shift problem with this data since the environment may change during the whole day due to light, weather, illumination, and traffic status. Six tasks are created by naming them with 'm/n' where m refers to the source sample size with a particular start index and n refers to the size of the target sample subsequently after the source batch. For each 'm/n' task, we chose six different start indices - 350, 650, 950, 1250, 1550, and 1850 - and reported their average accuracy.
- SIFT-SPM datasets [33]: This is the scenario identification problem identical to the GIST datasets, but the features are extracted using three pyramid layers, and the input dimension will be relatively large (e.g., > 4000). We created the tasks the same way as we did for the GIST datasets and reported the average accuracy for each.

For the first three transfer cases (OC-10, 20 Newsgroups, and handwritten digits), we use the whole batch of the source data, and the whole target instances arrive sequentially in an i.i.d. fashion. For the scenario identification problems (GIST and SIFT-SPM datasets), the target data arrive sequentially in a real-time manner. The benchmarks we compare with are listed as follows, and the detailed experimental setups can be found in Appendix A.18.

- Baseline: We only use the source data to predict the whole batch target with the linear SVM algorithm.
- Stochastic gradient descent (SGD): We conduct the prediction sequentially with the logistic regression model; that is, the model parameter will be updated in an online fashion every time we receive a target instance.
- Recurrent Neural Networks [19]: The method is based on the recurrent neural network with long-short term memory (LSTM) for the sequential prediction. The hidden layer size is set to be different for different tasks. We use the Adam optimizer with different learning rates η for different tasks.
- OTL [81]: We use the HomoOTL-I online transfer learning algorithm under the linear model with the Sigmoid smoothing output for binary classification problems and Softmax smoothing output for multi-class classification problems.
- CMA [25]: We use the continuous manifold-based adaptation method with the GFK feature mapping. We first transform the original target data with the GFK mapping under the CMA framework. Then we incorporate the label information from the target domain and iteratively conduct the 1-nearest-neighbor algorithm by combining the source and received target data with the GFK representation to predict the upcoming target data.
- EMPU: the efficient mixture posterior updating algorithm under the same linear model as the OTL algorithm. Here we assume the prior knowledge on the target parameter has a normal distribution where its variance is controlled by the gap coefficient c, i.e., $\theta_t \sim \mathcal{N}(\hat{\theta}_s, c\mathbf{I}_d)$ given the estimated source parameter $\hat{\theta}_s$. We set the quantization number $N = 20$ for all tasks but set the learning rate η and gap coefficient c differently for different tasks.
- EMPU with Neural Network: The efficient mixture posterior updating algorithm under the neural network model is also used as a benchmark. We also assume that the prior knowledge is normally distributed, i.e., $\theta_t \sim \mathcal{N}(\hat{\theta}_s, c\mathbf{I}_d)$ given the estimated source model parameters $\hat{\theta}_s$ and gap coefficient c. We set the quantization number $N = 20$ for all tasks but set the learning rate η, the gap coefficient c, and the layer architectures differently for different tasks.
- Gibbs EMPU: We apply the efficient mixture posterior updating algorithm under the Gibbs framework by iteratively optimizing (84) using the gradient descent algorithm. Here we set the number of prediction parameters $N = 20$ and the learning rate η and regularized coefficient γ differently for different tasks. The prior knowledge is characterized by γ, and the model is implemented by the linear model with the Sigmoid or Softmax smoothing output depending on the class number of the labels.

In particular, we choose the hyper-parameter c on the prior knowledge for the EMPU and EMPU with neural network in the following way for all experiments. We will first use the first 100 instances from the target domain to train a classifier, and we will denote it as $\hat{\theta}_t$. The principle of choosing $c = 10^k$ for some integer k is characterized in the following:

$$c = \operatorname{argmin}_{10^k} \left| 10^k - \frac{1}{d} \|\hat{\theta}_t - \hat{\theta}_s\|_1 \right|.$$

Here $\|\cdot\|_1$ denotes the l_1 norm and d denotes the model parameters. Such prior knowledge indicates that even though we do not know the exact target parameters, we know that the distance between the source and target is in the order of 10^k. For the Gibbs EMPU, $1/\gamma$ is empirically chosen to be from 10^{-6} to 10^{-4} for different tasks.

[3] http://yann.lecun.com/exdb/mnist/.
[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps.

Table 2
Accuracy in % after receiving whole target data for SURF Office-Caltech Dataset.

Tasks	Baseline	SGD	OTL	CMA	RNN	EMPU	EMPU-NN	G-EMPU
$A \to W$	31.53	58.31	41.36	64.75	68.81	68.81	69.83	68.47
$A \to D$	36.31	49.04	41.10	59.87	55.41	58.60	56.05	57.32
$A \to C$	35.44	50.13	44.00	40.87	53.87	56.37	54.40	54.59
$D \to A$	34.34	58.35	35.59	49.89	65.87	68.27	66.81	65.34
$D \to W$	77.97	83.05	84.07	67.46	84.75	87.12	87.46	88.81
$D \to C$	32.14	43.37	32.86	38.82	50.49	54.14	51.47	51.11
$W \to A$	37.79	61.59	37.58	53.97	67.01	69.52	67.12	66.91
$W \to D$	80.89	80.25	85.99	63.69	81.53	84.08	82.80	85.35
$W \to C$	33.93	45.41	33.75	34.82	48.17	52.45	50.49	51.13
$C \to A$	42.48	62.32	53.97	58.56	68.79	67.12	69.31	67.64
$C \to D$	33.76	45.22	41.40	66.24	58.60	50.32	58.60	50.32
$C \to W$	34.58	56.61	49.83	72.88	68.14	63.73	73.56	62.71
Average	42.60	57.80	47.66	55.99	63.91	65.04	65.66	64.14

Table 3
Accuracy in % after receiving whole target data for 20 Newsgroup Dataset.

Tasks	Baseline	SGD	OTL	CMA	RNN	EMPU	EMPU-NN	G-EMPU
rec vs talk	57.03	81.95	86.25	77.77	81.59	91.16	92.24	92.55
comp vs sci	57.42	86.59	89.05	71.03	87.72	91.45	91.91	93.19
rec vs sci	56.63	85.98	85.53	72.01	83.16	89.96	92.18	91.63
talk vs sci	67.20	70.88	85.95	72.97	85.33	87.48	90.99	88.90
comp vs rec	64.83	88.57	90.45	70.60	89.54	92.80	92.39	94.18
comp vs talk	65.06	74.83	86.25	72.95	87.85	92.37	92.33	94.14
Average	61.36	81.47	87.35	72.89	85.86	90.87	92.00	92.43

Table 4
Accuracy in % after receiving whole target data for Handwritten Digits Dataset.

Tasks	Baseline	SGD	OTL	CMA	RNN	EMPU	EMPU-NN	G-EMPU
U to M	28.40	71.72	75.06	85.10	81.17	80.28	80.44	81.94
M to U	45.33	63.50	70.10	83.20	79.45	75.00	80.70	78.85
Average	36.87	67.61	72.58	84.15	80.31	77.64	80.57	80.35

Table 5
Accuracy in % after receiving whole target data for GIST Dataset.

Tasks	Baseline	SGD	OTL	CMA	RNN	EMPU	EMPU-NN	G-EMPU
50/480	66.22	83.44	80.59	83.71	83.89	84.20	84.30	83.54
50/1200	66.35	84.76	83.07	84.60	84.78	85.63	85.76	84.82
50/2400	69.04	86.60	84.76	86.86	86.13	86.18	86.49	86.57
100/480	72.05	85.14	80.87	83.51	83.37	84.86	84.48	85.10
100/1200	67.22	86.46	83.36	85.00	84.99	86.84	85.57	86.28
100/2400	72.35	87.47	84.88	87.21	86.38	86.81	86.60	87.18
Average	68.87	85.64	82.92	85.15	84.92	85.75	85.43	85.58

Performance Comparisons For each algorithm, we will make prediction $\hat{Y}_i$ at each iteration and report the accuracy till time T:

$$\mathcal{R}_{acc} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{1}_{Y_i = \hat{Y}_i}. \tag{85}$$

The results are reported from Table 2 to 6. In the first two transfer learning tasks (OC-10 and 20 Newsgroup), we observe that all three EMPU-based algorithms outperform the other benchmarks in different sub-tasks. This also shows that by using prior knowledge properly, we can improve the efficiency of online transfer learning. However, for the third learning task (MNIST vs USPS), CMA outperforms the rest parameter-based algorithms because it utilizes the geometric property of the data with the nearest neighbor algorithm and uses all the source and target domain data when making the prediction. Despite this, EMPU is still more efficient than other parameter-based algorithms such as OTL and SGD. For the time-varying transfer learning tasks (GIST and SIFT-SPM), it is evident that the EMPU-based algorithms still maintain relatively high accuracy, making them the best among the others. The EMPU-based algorithms do not appear to have obvious advantages over other benchmarks for the GIST data as the data dimension is relatively small. When the data dimension becomes larger,

Table 6
Accuracy in % after receiving whole target data for SIFT-SPM Dataset.

Tasks	Baseline	SGD	OTL	CMA	RNN	EMPU	EMPU-NN	G-EMPU
50/480	71.56	75.00	72.71	74.29	75.52	76.00	81.25	77.40
50/1200	71.21	78.72	77.32	75.00	79.74	78.71	82.97	80.25
50/2400	74.15	81.79	81.44	73.18	82.76	82.22	85.58	82.92
100/480	73.47	76.08	73.51	75.20	77.74	77.01	81.81	77.88
100/1200	71.49	79.28	77.99	76.45	81.29	80.04	83.90	81.48
100/2400	75.20	82.22	81.46	76.45	82.85	82.79	85.85	83.69
Average	72.85	78.85	77.40	75.07	79.98	79.46	83.56	80.60

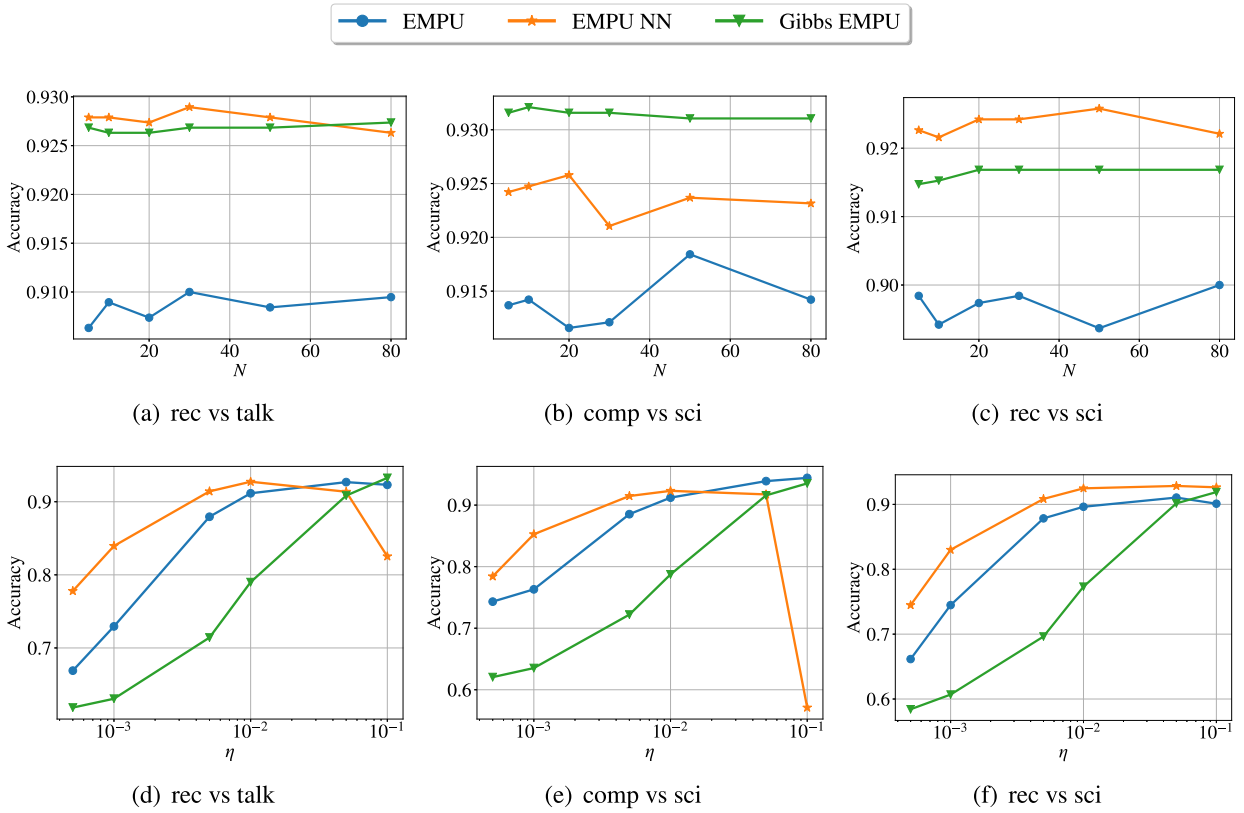
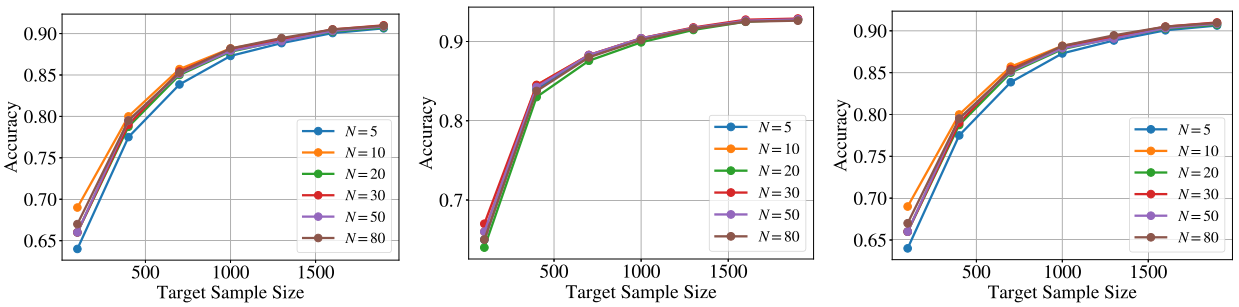


Fig. 14. Sensitivity on N and η on three transfer tasks in 20 Newsgroup datasets. The first row shows the accuracy of different EMPU algorithms after receiving the whole target batch with different N. Under the same setting, the second row shows the accuracy with different η.

for example, in the SIFT-SPM dataset, the dimension of the data increases to 4200, it is more difficult for the CMA-based algorithm to identify the accurate labels, and the simple linear EMPU model is less effective due to that the target sample size is relatively small to learn a promising model. The EMPU with the neural network still provides promising prediction results in this case. The reason is that a more complex model can capture and exploit the differences between target and source domains for the high-dimensional data, thus achieving better prediction results.

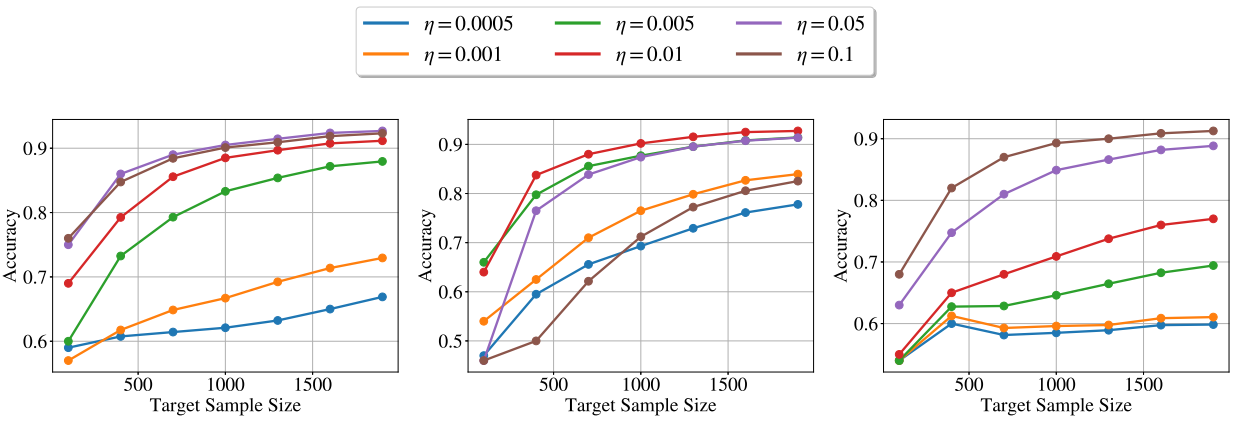
To examine the effects of quantization number N, the learning rate η and prior knowledge coefficient c, γ by fixing one variable and varying another. As an example, we conduct a series of experiments on hyperparameter sensitivity in the 20 Newsgroup dataset.

Effect of N By varying N from 5 to 80, we examine the effects of the quantization number in two ways. First, we plot the accuracy after receiving the whole target data with different quantization numbers N for different tasks and algorithms in Fig. 14(a)-14(c). We also plot the (sequential) accuracy versus the number of the target we received to show the learning procedures for different algorithms under the same task, and the result is sketched in Fig. 15. Considering the space limitation and the conciseness of the presentation, we only present the accuracy on a few sub-tasks while the results of other sub-tasks are similar. From the Figure, we can observe a similar pattern under different algorithms, i.e., the choice of N does not significantly impact the final or the sequential accuracy. The degradation only occurs when $N = 5$ and the accuracy is



(a) EMPU (b) EMPU NN (c) Gibbs EMPU

Fig. 15. Sensitivity on N for the task `rec vs talk` under the different EMPU algorithms.



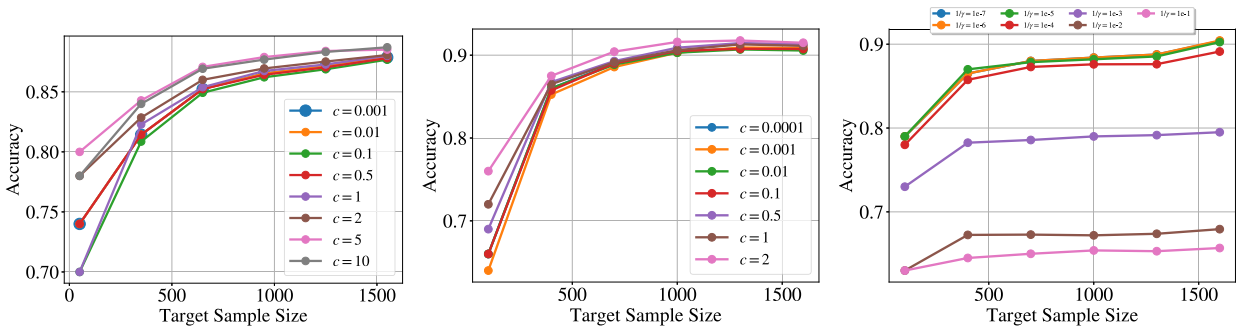
(a) EMPU (b) EMPU NN (c) Gibbs EMPU

Fig. 16. Sensitivity on η for the task `rec vs talk` under the different EMPU algorithms.

slightly worse than the other choices of N. Therefore, to save the computation cost while maintaining the performance, we choose $N = 20$ for all tasks.

Effect of η Similarly, we evaluate the effect of the learning rate η in the same way as we did with the quantization number N. The effects of the learning rate η are illustrated in 14(d)-14(f) for the accuracy of the whole target data, and Fig. 16 for the sequential accuracy. From this, we can see that the learning rate significantly impacts sequential predictions. Regarding the accuracy of the whole target data, the performance of the linear EMPU algorithm and the EMPU with the neural network will be degraded when the learning rate is too large or too small, as the extreme learning rate makes the model difficult to converge. In the case of Gibbs EMPU, the accuracy will increase as η increases, but the result will level off after 0.05. From Fig. 16, we can observe a similar pattern for sequential accuracy: as we gather more and more target data, the accuracy will increase, and the improvement closely hinges on the magnitude of the learning rate. Consequently, for other experiments, we experimentally choose a learning rate of either 0.001 or 0.01, depending on the order of magnitude of the model parameters.

Effect of the prior knowledge The effect of the prior knowledge is illustrated in Fig. 17 evaluated under the task `talk vs sci`. Since the optimal parameters will be for different models: for the linear EMPU algorithm, we vary the distance parameter c from 0.001 to 10 for a thorough examination; For EMPU with the neural network, we vary c from 0.0001 to 2; For Gibbs algorithm, we vary $1/\gamma$ from 1e-7 to 0.1. For the linear EMPU and EMPU with the neural network, the parameter c is directly associated with the distance between the source and target parameters. Hence it will impact the prediction for those early upcoming target data, e.g., when the target sample size is smaller than 500. Specifically, in EMPU, the selection of $c = 5$ will achieve the best performance, which matches the true distance $\frac{1}{d}\|\hat{\theta}_s - \hat{\theta}_t\| \approx 6$ estimated from all batch target data and batch source data. Therefore, a good c should lead to a prior distribution that better fits the true distance. As the target sample size increases, the prior information is overwhelmed by sufficient data, and thus the impact of c becomes insignificant. For Gibbs EMPU, since the prior information for this case is characterized by the regularized term in (84), we can vary γ for controlling our belief on $\hat{\theta}_s$. If γ is set to be small, we will be less focused on empirical minimization, resulting in an under-fitted model. On the contrary, if γ is set to be large, the role of $\hat{theta}_s$ is diminished,



(a) EMPU (b) EMPU NN (c) Gibbs EMPU

Fig. 17. Sensitivity on c for the task `talk vs sci` under the EMPU and EMPU with neural networks and γ under the Gibbs EMPU.

and the source data becomes less efficacious in some positive transfer cases. As such, the choice of a suitable γ is crucial, and we empirically set $1/\gamma$ to be from 10^{-6} to 10^{-4} depending on the gradient and the magnitude of the parameters for different tasks.

We empirically verify the effectiveness of our proposed algorithm and particularly examine the sensitivity of different hyper-parameters. We find that the quantization number N has less impact on the performance of all the EMPU models, while an appropriate selection of the learning rate and parameter on the prior information is critical. The choice of the parameter c also matches the intuitions from both theoretical results and experiments: introducing the source data can effectively improve the performance in the target domain with the appropriate prior information. In the case of learning from high-dimensional data, the performance of the EMPU algorithm can be improved by choosing a more complex model such as neural networks. Due to its parameter-based nature, these algorithms lack the ability to exploit data geometric properties well, such as USPS and MNIST data with the SURF features.

4.5. Nonparametric modeling

Previous algorithms are derived under the assumption that the data distributions or the models are parameterized by θ. That is, the model parameters are fixed once we select a particular model architecture. To further relax the constraint, we devise a novel algorithm that works in a nonparametric setting for more general models. Roughly speaking, the nonparametric algorithm allows the number of parameters in the model to grow with the number of samples. Similar to the mixture strategy on a parametric model, we will define a distribution on the parameter of possibly infinite dimension. Specifically, we use the Dirichlet Process Mixture (DP) [16,2,42] to construct the prior and permit the posterior inference from the data. The Dirichlet Process is usually denoted by

$$G \sim DP(\alpha, G_0),$$

where G is a random discrete measure. There are two parameters in the Dirichlet process: the base distribution G_0, which represents a random guess of the data true distribution, and α, a positive scalar controlling the concentration. From the DP, we could model the data distributions by the DP mixture (DPM) where the distribution parameters $\theta_i, i \in \mathbb{N}$ are drawn from some random measure G sampled from $DP(\alpha, G_0)$, and each data Z_i is drawn from the parametric distribution $P_{\theta_i}(Z)$. If we integrate a parametric density $P_{\theta_i}(Z)$ against the random measure G, we obtain a mixture model from the stick-breaking process [43]. First, we revisit some basic properties of the DP. Then we will show how it is adapted to transfer learning algorithms and related to the parametric learning framework. Let us first consider the case where the received data are discrete, e.g. $Z \in \mathcal{Z}$ where $|\mathcal{Z}| < \infty$. Under the DP, data is generated in two stages by

$$\begin{aligned} G &\sim DP(\alpha, G_0), \\ Z_1, Z_2, \cdots | G &\sim G. \end{aligned} \tag{86}$$

We can write it as $G = \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i}$ almost surely. Then we can construct π_i via the stick-breaking process [43] and sample θ_i from G_0, which is known as the sticking-breaking representation for Dirichlet Process. Given data $Z_1, Z_2, \cdots, Z_n$, we can write out the posterior of G, which is also a DP as

$$G | Z_1, Z_2, \cdots, Z_n \sim DP(\alpha + n, \frac{\alpha G_0 + \sum_{i=1}^{n} \delta_{Z_i}}{n + \alpha}). \tag{87}$$

By integrating out the random measure G, the joint distribution $P(Z_1, Z_2, \cdots, Z_n)$ will follow the Pólya urn scheme or Chinese Restaurant Process (CRP) [17] such that we can calculate the posterior by

$$Z_{n+1} \mid Z_1, \ldots, Z_n \sim \sum_{i=1}^{n} \frac{1}{n+\alpha}\delta_{Z_i} + \frac{\alpha}{n+\alpha}G_0. \tag{88}$$

The beauty of this generative model is actually clustering the data automatically. The reason is that G itself is a discrete measure, and the samples Z_i drawn from G are more likely to lie in several different clusters. Within each cluster, Z_i has exactly the same value.

$$P(Z) = \sum_{i\in\mathbb{N}} \pi_i P_{\theta_i}(Z). \tag{89}$$

We sketch the following procedure to illustrate how the data are generated from the DPM:

$$\begin{aligned} G &\sim DP(\alpha, G_0), \\ \theta_1, \theta_2, \cdots | G &\sim G, \\ Z_i | \theta_i &\sim P_{\theta_i}(Z). \end{aligned} \tag{90}$$

Using this generative model, we view θ_i as the latent variables and the aim is to estimate the posterior $P(\theta_1, \theta_2, \cdots, \theta_n | Z_1, Z_2, \cdots, Z_n)$ from the dataset as each sample corresponds to an atom θ_i that is drawn from G. More specifically, we can find conditional distributions of the posterior distribution of model parameters by combining the conditional probability as [42]

$$\theta_i | \theta^{-i}, Z_i \sim \sum_{j\neq i} q_{ij}\delta_{\theta_j} + r_i G_i, \tag{91}$$

where $\theta^{-i} = \{\theta_1, \cdots, \theta_{i-1}, \theta_{i+1}, \cdots, \theta_n\}$ denotes the set of θ_j for all $j \neq i$, G_i is the posterior distribution of θ based on the base distribution G_0 and data Z_i, and the coefficients q_{ij} and r_i are defined as

$$q_{ij} = bP_{\theta_j}(Z_i), \tag{92}$$

$$r_i = b\alpha \int P_\theta(Z_i) dG_0(\theta), \tag{93}$$

where b is such that $\sum_{j\neq i} q_{ij} + r_i = 1$. At this stage, we can use the MCMC sampling method to approximate the posterior. Inspired by this non-parametric Bayesian framework, we now describe how the idea of mixture strategy in the parametric model carries over to the nonparametric model. Assume that the distributions of each source data $Z_s^{(i)}$ and target data $Z_t^{(i)}$ are parameterized by $\theta_{s,i}$ and $\theta_{t,i}$, respectively. And we assume that $\theta_{s,i}$ and $\theta_{t,i}$ are drawn from the distributions G_s and G_t generated by the prior $DP(\alpha, G_0)$. That is:

$$\begin{array}{cc} G_s \sim DP(\alpha, G_0) & G_t \sim DP(\alpha, G_0) \\ \theta_{s,1}, \theta_{s,2}, \cdots | G_s \sim G_s & \theta_{t,1}, \theta_{t,2}, \cdots | G_t \sim G_t \\ Z_t^{(i)} | \theta_{t,i} \sim P_{\theta_{t,i}}(Z) & Z_s^{(j)} | \theta_{s,j} \sim P_{\theta_{s,j}}(Z) \end{array}. \tag{94}$$

Consider the case that both the source and target data are discrete and they are assumed to be drawn from two different distributions G_s and G_t where both G_s and G_t are sampled from some DP prior, i.e.,

$$\begin{array}{cc} G_s \sim DP(\alpha, G_0) & G_t \sim DP(\alpha, G_0) \\ Z_s^{(1)}, Z_s^{(2)}, \cdots | G_s \sim G_s & Z_t^{(1)}, Z_t^{(2)}, \cdots | G_t \sim G_t \end{array} \tag{95}$$

Our aim is to estimate the posterior of G_t given all source and target samples by

$$P(G_t | D_s^m, D_t^n) \propto \int P(G_t | D_t^n) \omega(G_t | G_s) dP(G_s | D_s^m). \tag{96}$$

From the prediction side, we integrate out the distribution G and it yields the conditional probability,

$$P(Z_t^{(n+1)} | D_s^m, D_t^n) = \frac{\alpha G_0}{\alpha + n + m} + \frac{\sum_{i=1}^{n} \delta_{Z_t^{(i)}}}{\alpha + n + m} + \frac{\sum_{i=1}^{m} \delta_{Z_s^{(i)}}}{\alpha + n + m}. \tag{97}$$

Following Equation (91), we can write out the posterior of $\theta_{t,i}$ given the target data $Z_t^{(i)}$, the set of $\theta_{t,j}$ for all $j \neq i$ (written as θ_t^{-i}) and the set of $\theta_{s,k}$ for $k = 1, 2, \cdots, m$ (written as θ_s^m):

$$\theta_{t,i} | \theta_t^{-i}, \theta_s^m, Z_i \sim \sum_{j\neq i} q_{ij}\delta_{\theta_{t,j}} + \sum_k q_{ik}\delta_{\theta_{s,k}} + r_i G_i. \tag{98}$$

Table 7
Comparisons between Parametric and Nonparametric Models.

	Parametric	Nonparametric
Prior	$\Theta_t \sim \omega(\Theta_t \mid \Theta_s)$	$G_t \sim DP(\alpha, \beta G_s + (1-\beta)G_0)$
Likelihood	$P_{\theta_s^*}(Z_s)$, $P_{\theta_t^*}(Z_t)$	$P_{\theta_{s,j}}(Z_s^{(j)})$, $P_{\theta_{t,i}}(Z_t^{(i)})$
Mixture	$\int P_{\theta_s}(D_t^n)\omega(\theta_t \mid \theta_s)d\theta_t P(\theta_s \mid D_s^m)d\theta_s$	$\sum_{j\neq i} q_{ij}\delta_{\theta_{t,j}} + \sum_k q_{ik}\delta_{\theta_{s,k}} + r_i G_i$
Prediction	$\operatorname{argmin}_b \mathbb{E}_Q[\ell(b, Z_t)]$	$\operatorname{argmin}_b \sum_{i=1}^K \mathbb{E}_{\theta_{t,i}}[\ell(b, Z_t)]$

The coefficients q_{ij}, q_{ik} and r_i are defined as follows

$$q_{ij} = bP_{\theta_{t,j}}(Z_i), \tag{99}$$

$$q_{ik} = \beta b\alpha P_{\theta_{s,k}}(Z_i), \tag{100}$$

$$r_i = (1-\beta)b\alpha \int P_\theta(Z_i)dG_0(\theta), \tag{101}$$

where b is such that $\sum_{j\neq i} q_{ij} + \sum_k q_{ik} + r_i = 1$. Here we introduce the balancing coefficient $\beta \in [0,1]$ as our prior knowledge over the source and target domain. Larger β implies that we will rely more on the (empirical) source distribution as our prior and smaller β shows more beliefs on G_0. The next step is to estimate the posterior of $\theta_{t,i}$ from Equation (98) by the MCMC sampling algorithm. We then heuristically propose an efficient algorithm for online transfer learning under nonparametric Bayesian learning framework.

Algorithm 4: Nonparametric Posterior Updating and Prediction.

input : $\mathcal{D}_t^m$, Base distribution G_0, α, weighting coefficient β, parametric family P_θ
Estimate G_s from $\mathcal{D}_S^m$ from the Gibbs sampling by (91) ;
Initialize distribution $DP(\alpha, (1-\beta)G_0 + \beta G_s)$ as our prior knowledge for the target domain as indicated by (98);
for $k = 1, \cdots, T$ **do**
  Receive target sample $Z_t^{(k)}$;
  for $i = 1, \cdots, K$ **do**
    Estimate the posterior of the target distribution G_t using Equation (98) by Gibbs sampling, say, we sample $\theta_{t,k}^i$;
  end
  Predict $P_k(Z_t^{(k)}) = \sum_{i=1}^K P_{\theta_{t,k}^i}(Z_t^{(k)})$ using sampled G_t ;
end
output: Sequential prediction $P_k(Z_t^{(k)})$

Remark 8. We can see the analogy between the parametric and nonparametric models from the above algorithm. Firstly we estimate G_s from the source data with the Gibbs sampling, which behaves similarly to estimating θ_s in the parametric model. Then we define the coefficient weight β that controls whether G_t is similar to G_s as our prior knowledge, which corresponds to $\omega(\Theta_t \mid \Theta_s)$ in the parametric model. When we receive new target samples $Z_t^{(k)}$, the posterior is updated and $\theta_{t,k}$ are sampled K times from G_t. Then the probability distribution of $Z_t^{(k)}$ is approximated by the mixture of each θ_i. The effectiveness of this algorithm will depend on α, β, choice of G_0, the parametric family P_θ and the sampling number K. The parametric and nonparametric modeling analogy is summarized in Table 7. Even though the two methods have some similarities, there is no theoretical guarantee for the generalization performance of the nonparametric models. To show the effectiveness of the prior knowledge in the nonparametric algorithm, we conduct some simple experiments for empirical verification.

Experiments We illustrate the effect of the prior knowledge in the nonparametric model by firstly validating the algorithm in logistic regression problems for small and large discrepancy scenarios as described in 4.2. Here we use the same hyperparameters and parametric models given in [52] (See Simulation 1 for more details). We set $\alpha = 0.01$ and vary β The results are shown in Fig. 18. From the regret curves, we can see that if the domain divergence is small, one can achieve a better regret with a large β because the posterior will rely more on the source data, which is helpful for prediction. On the contrary, if the target distribution differs much from the source distribution, the regret will be higher with large β as relying more on the source data will hurt the performance of the target. If we decrease β to 0.01, the prediction counts more on the target data. Regardless of the source data, the regrets become close under small and large discrepancies.

We also validate the nonparametric algorithm for a real-world transfer problem using the MNIST [34] and USPS datasets [27], which are standard digit recognition datasets containing hand-written digits from 0-9. USPS contains 7291 training samples and 2007 testing samples with the resolution of 16 $\times$ 16, while MNIST consists of 60000 images for training and 10000 images for testing with the resolution of 28×28. We sampled 2000 images from the MNIST dataset and 1800 images

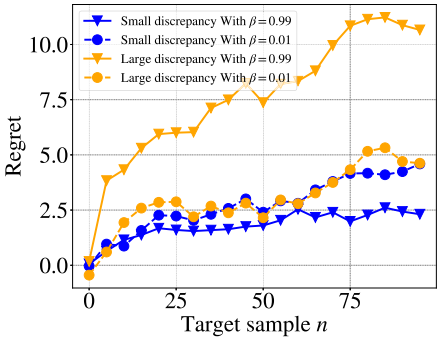


Fig. 18. Regret of the Logistic regression problems under the small and large discrepancy scenarios, we choose β to be 0.99 and 0.01 to show the usefulness of the source.

from the USPS dataset and adopted the 256 SURF features[5] following the existing literature [59]. We conduct the sequential knowledge transfer between these two datasets. For simplicity, we use U $\rightarrow$ M (M $\rightarrow$ U) to denote the transfer from the source domain USPS (MNIST) to the target domain MNIST (USPS). For each transfer case, we use the whole batch of the source data and randomly sample 80 samples from the target data as the initialization. Then the rest target will arrive sequentially. Our nonparametric algorithm will output the predictive probability for each class, and we will plot the regret by the multi-class cross-entropy loss according to the prediction:

$$\mathcal{R}_O = -\sum_{i=1}^{T}\sum_{k=1}^{C} \mathbf{1}_{Y_i=k} \log\left(P_{\theta_i}(Y=k|X_i)\right) \tag{102}$$

where $\mathbf{1}_{Y_i=k}$ denotes the indicator function that whether the output Y_i is equal to the kth-class, and $P_{\theta_i}(Y=k|X_i)$ denotes the predictive probability of kth-class given the parameters θ_i from the posterior of the Dirichlet process mixture and X_i. Moreover, we will make prediction $\hat{Y}_i$ at each iteration according to the class with the highest predictive probability and plot the number of mistakes:

$$\mathcal{R}_{mistake} = \sum_{i=1}^{T} \mathbf{1}_{Y_i \neq \hat{Y}_i}. \tag{103}$$

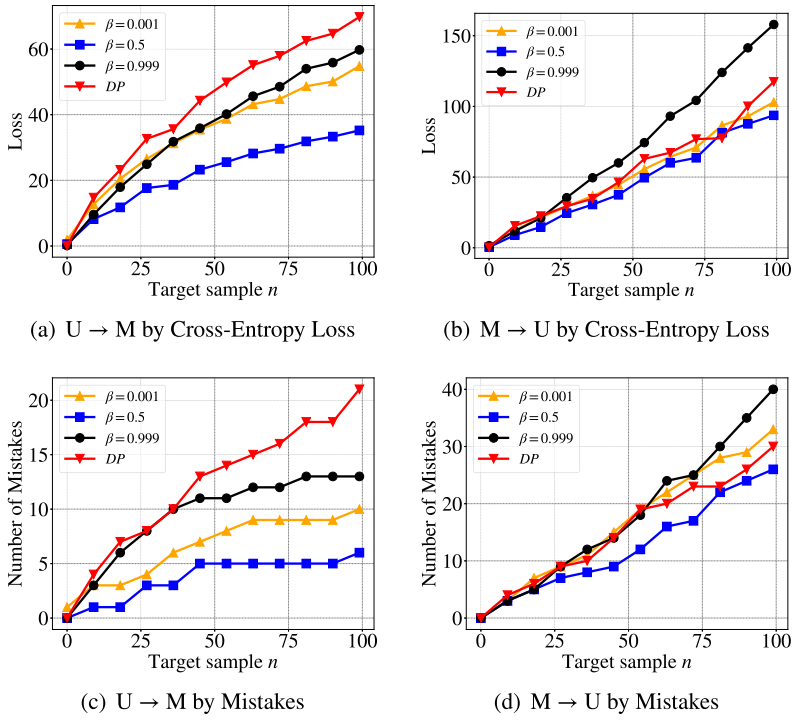
Owing to the fact that the computational cost for sampling the posterior with 256 input dimensions becomes an issue in real implementation, we first reduce the dimension from 256 to 16 by some projection methodologies in domain adaptation such as [37]. By this, we could alleviate the computational cost issues and reduce the domain divergence. Then we vary β from 0.001 to 0.999 to show our prior belief on the usefulness of the source and target data. We also compare our method to the baseline dpMNL method described in [52] by treating the source and target equally as one domain. After receiving 100 target samples, we plot the accumulated mistakes in Fig. 19.

From the experimental results, one can find that in both M $\rightarrow$ U and U $\rightarrow$ M settings, a very large β (0.999) or a very small β (0.001) may yield unsatisfactory predictive results compared to a moderate β value (0.5), which achieves better performance than two extreme cases. Heuristically the performance mainly depends on how significant the domain divergence is or how large the target sample size is. Explicitly, if β is too large, the prediction will rely more on the source domains and from which the negative transfer may occur if domain divergence is relatively large. When it comes to a small β, as the prediction does not benefit from the source domain, the performance mainly relies on the target samples with a comparatively small size, leading to undesirable loss and mistakes. With a moderate choice of β, the algorithm tries to strike a balance between the two extremes. Furthermore, with the baseline dpMNL method described in [52], the performance is not desirable if we treat the source and target data equally as one domain. This is possibly due to the fact that the source data is regarded as equally useful as the target data, while the domain discrepancy is not taken into account and thus cannot be alleviated in the learning and prediction, which will lead to a poor result. We may select a proper β at a moderate level to achieve better performance. A moderate β will properly extract the information from the limited target data and not overly trust the source data if the distribution differs. We empirically explore a trade-off between β and the performance, exposing the inherent nature of transfer learning.

5. Conclusion

We propose a general framework for transfer learning from a Bayesian approach. This learning framework extends traditional learning regimes to the case where the predictor is learned and deployed on samples drawn from different yet

[5] https://github.com/jindongwang/transferlearning/tree/master/data.



(a) U → M by Cross-Entropy Loss (b) M → U by Cross-Entropy Loss

(c) U → M by Mistakes (d) M → U by Mistakes

Fig. 19. The comparisons on the prediction performance for the knowledge transfer between MNIST and USPS datasets by the cross-entropy and the mistakes. We use U → M (M → U) to denote the transfer from the source domain USPS (MNIST) to the target domain MNIST (USPS).

related probability distributions in terms of parameterization. Specifically, the instantaneous, online, and time-variant transfer learning scenarios are examined, and the learning performance takes the shape of conditional mutual information. We give the asymptotic estimation of the conditional mutual information and identify the situations when the negative and positive transfer will happen. However, in our analysis, the i.i.d. properties of both source and target data are crucial. The natural follow-up challenge is formalizing non-i.i.d. settings for transfer learning and finding similar mixture strategies for efficient utilities. Another future work is to relax the assumptions on parametric conditions to general probability distributions and rigorously find the performance guarantee under the nonparametric framework with finite sample sizes, which will improve the generality and applicability of our methods.

Funding

This work was supported by the Melbourne Research Scholarship, University of Melbourne.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jingge Zhu reports a relationship with Australian Defence Science and Technology Group that includes: funding grants from project DE210101497.

Data availability

Data will be made available on request.

Acknowledgements

The preliminary version of this work is presented at the ISIT2021 conference, we greatly appreciate useful feedback and comments from all reviewers.

Appendix A

In this section, we give detailed proofs of the main theorems.

A.1. Proof of Theorem 1

Proof. We firstly show that given any prior over Θ_s and Θ_t,

$$I(Z_t'; \Theta_t, \Theta_s | D_t^n, D_s^m) = I(\Theta_t, \Theta_s; Z_t', D_t^n, D_s^m) - I(\Theta_t, \Theta_s; D_s^n, D_s^m) \tag{104}$$

$$= D(P_{\Theta_t,\Theta_s}(D_t^n, D_s^m, Z_t') \| Q(D_t^n, D_s^m, Z_t')) - D(P_{\Theta_s,\Theta_t}(D_s^m, D_t^n) \| Q(D_s^m, D_t^n)) \tag{105}$$

$$= \int \left(\mathbb{E}_{\theta_s,\theta_t} \left[\log \frac{P_{\theta_t,\theta_s}(D_t^n, D_s^m, Z_t')}{Q(D_t^n, D_s^m, Z_t')} \right] - \mathbb{E}_{\theta_s,\theta_t} \left[\log \frac{P_{\theta_s,\theta_t}(D_s^m, D_t^n)}{Q(D_s^m, D_t^n)} \right] \right) \omega(\theta_s, \theta_t) d\theta_s d\theta_t \tag{106}$$

$$= \int \left(\mathbb{E}_{\theta_s,\theta_t} \left[\log \frac{P_{\theta_t}(Z_t')}{Q(Z_t' | D_t^n, D_s^m)} \right] \right) \omega(\theta_s, \theta_t) d\theta_s d\theta_t, \tag{107}$$

where in the last equality we use the chain rule and the assumption that both source and target data are drawn in an i.i.d. fashion. The mutual information density at $\Theta_s = \theta_s^*$ and $\Theta_t = \theta_t^*$ is then given by

$$I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_t^n, D_s^m) = \mathbb{E}_{\theta_s^*,\theta_t^*} \left[\log \frac{P_{\theta_t^*}(Z_t')}{Q(Z_t' | D_t^n, D_s^m)} \right] = \mathcal{R}_I, \tag{108}$$

which completes the proof.

A.2. Proof of Theorem 2

Proof. We can show that the excess risk for instantaneous transfer learning scenario can be bounded as

$$\mathcal{R}_I = \mathbb{E}_{\theta_t^*,\theta_s^*} \left[\ell(b, Z_t') - \ell(b^*, Z_t') \right] \tag{109}$$

$$= \mathbb{E}_{D_s^m, D_t^n} \mathbb{E}_{Z_t'} \left[\ell(b, Z_t') - \ell(b^*, Z_t') | D_s^m, D_t^n \right] \tag{110}$$

$$= \mathbb{E}_{D_s^m, D_t^n} \int \left(\ell(b, z_t') - \ell(b^*, z_t') \right) P_{\theta_s^*,\theta_t^*}(z_t' | D_s^m, D_t^n) dz_t' \tag{111}$$

$$= \mathbb{E}_{D_s^m, D_t^n} \int \left(\ell(b, z_t') - \ell(b^*, z_t') \right) (P_{\theta_s^*,\theta_t^*}(z_t' | D_s^m, D_t^n) - Q(z_t' | D_s^m, D_t^n) + Q(z_t' | D_s^m, D_t^n)) dz_t' \tag{112}$$

$$\overset{(a)}{\leq} \mathbb{E}_{D_s^m, D_t^n} \int \left(\ell(b, z_t') - \ell(b^*, z_t') \right) (P_{\theta_s^*,\theta_t^*}(z_t' | D_s^m, D_t^n) - Q(z_t' | D_s^m, D_t^n)) dz_t' \tag{113}$$

$$\overset{(b)}{\leq} M \mathbb{E}_{D_s^m, D_t^n} \int (P_{\theta_s^*,\theta_t^*}(z_t' | D_s^m, D_t^n) - Q(z_t' | D_s^m, D_t^n)) dz_t' \tag{114}$$

$$\overset{(c)}{\leq} M \mathbb{E}_{D_s^m, D_t^n} \sqrt{2D \left(P_{\theta_s^*,\theta_t^*}(Z_t' | D_s^m, D_t^n) \| Q(Z_t' | D_s^m, D_t^n) \right)} \tag{115}$$

$$\overset{(d)}{\leq} M \sqrt{2 \mathbb{E}_{D_s^m, D_t^n} D \left(P_{\theta_s^*,\theta_t^*}(Z_t' | D_s^m, D_t^n) \| Q(Z_t' | D_s^m, D_t^n) \right)} \tag{116}$$

$$= M \sqrt{2D \left(P_{\theta_t^*} \| Q | D_s^m, D_t^n \right)} \tag{117}$$

$$= M \sqrt{2D(P_{\theta_t^*}(Z_t') \| Q(Z_t' | D_t^n, D_s^m))} \tag{118}$$

$$= M \sqrt{2I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_s^m, D_t^n)}, \tag{119}$$

where in (a) we use the definition of Q, then (b) holds since we assume the loss function is bounded, (c) follows from the Pinsker's inequality, (d) holds from the Jensen's inequality.

A.3. Proof of Theorem 3

Proof. Under the exponentially concave assumption, we can show that the excess risk for an instantaneous transfer learning scenario can be bounded as

$$\mathcal{R}_I = \mathbb{E}_{\theta_t^*,\theta_s^*}\left[\ell(b, Z_t') - \ell(b^*, Z_t')\right] \tag{120}$$

$$= \mathbb{E}_{D_s^m, D_t^n}\mathbb{E}_{Z_t'}\left[\ell(b, Z_t') - \ell(b^*, Z_t')|D_s^m, D_t^n\right] \tag{121}$$

$$= \mathbb{E}_{D_s^m, D_t^n}\int \left(\ell(b, z_t') - \ell(b^*, z_t')\right) P_{\theta_s^*,\theta_t^*}(z_t'|D_s^m, D_t^n)dz_t' \tag{122}$$

$$= \frac{1}{\beta}\mathbb{E}_{D_s^m, D_t^n}\int \beta\left(\ell(b, z_t') - \ell(b^*, z_t')\right)(P_{\theta_s^*,\theta_t^*}(z_t'|D_s^m, D_t^n))dz_t' \tag{123}$$

$$\overset{(a)}{\leq} \frac{1}{\beta}\mathbb{E}_{D_s^m, D_t^n}\int \left(\log\frac{e^{-\beta\ell(b^*,z_t')}Q(z_t'|D_s^m, D_t^n)}{e^{-\beta\ell(b,z_t')}P_{\theta_s^*,\theta_t^*}(z_t'|D_s^m, D_t^n)} + \log\frac{P_{\theta_s^*,\theta_t^*}(z_t'|D_s^m, D_t^n)}{Q(z_t'|D_s^m, D_t^n)}\right) P_{\theta_s^*,\theta_t^*}(z_t'|D_s^m, D_t^n)dz_t' \tag{124}$$

$$\overset{(b)}{\leq} \frac{1}{\beta}\mathbb{E}_{D_s^m, D_t^n}\log\int \frac{e^{-\beta\ell(b^*,z_t')}}{e^{-\beta\ell(b,z_t')}}Q(z_t'|D_s^m, D_t^n)dz_t' + \frac{1}{\beta}I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m, D_t^n) \tag{125}$$

$$\overset{(c)}{\leq} \frac{1}{\beta}I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m, D_t^n), \tag{126}$$

where in (a) we simply do the decomposition for the excess risk with the log trick, (b) follows due to the Jensen's inequality, (c) holds from the following inequality:

$$\mathbb{E}_{D_s^m, D_t^n}\log\int \frac{e^{-\beta\ell(b^*,z_t')}}{e^{-\beta\ell(b,z_t')}}Q(z_t'|D_s^m, D_t^n)dz_t' \leq 0, \tag{127}$$

which can be proved using the property of the exponential concavity of the loss function, e.g., see Lemma 3 in [82].

A.4. Proof of Theorem 4

Proof. We firstly show that given any prior over Θ_s and Θ_t,

$$I(D_t^n; \Theta_t, \Theta_s|D_s^m) = I(\Theta_t, \Theta_s; D_t^n, D_s^m) - I(\Theta_s; D_s^m) \tag{128}$$

$$= D(P_{\Theta_t,\Theta_s}(D_t^n, D_s^m)\|Q(D_t^n, D_s^m)) - D(P_{\Theta_s}(D_s^m)\|Q(D_s^m)) \tag{129}$$

$$= \int\left(\mathbb{E}_{\theta_s,\theta_t}\left[\log\frac{P_{\theta_t,\theta_s}(D_t^n, D_s^m)}{Q(D_t^n, D_s^m)}\right] - \mathbb{E}_{\theta_s,\theta_t}\left[\log\frac{P_{\theta_s}(D_s^m)}{Q(D_s^m)}\right]\right)\omega(\theta_s, \theta_t)d\theta_s d\theta_t \tag{130}$$

$$= \int\left(\mathbb{E}_{\theta_s,\theta_t}\left[\log\frac{P_{\theta_t}(D_t^n)}{Q(D_t^n|D_s^m)}\right]\right)\omega(\theta_s, \theta_t)d\theta_s d\theta_t, \tag{131}$$

where in the last equality we use the chain rule and the assumption that source data are independent of Θ_t. The mutual information density at $\Theta_s = \theta_s^*$ and $\Theta_t = \theta_t^*$ is then given by

$$I(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m) = \mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right] = \mathcal{R}_O. \tag{132}$$

A.5. Proof of Theorem 5

Proof. We can show that the expected regret for online transfer learning scenario can be bounded as

$$\mathcal{R}_O = \mathbb{E}_{\theta_t^*,\theta_s^*}\left[\sum_{k=1}^n \ell(b_k, Z_t^{(k)}) - \sum_{k=1}^n \ell(b_k^*, Z_t^{(k)})\right] \tag{133}$$

$$= \sum_{k=1}^n \mathbb{E}_{D_s^m, D_t^{k-1}}\mathbb{E}_{Z_t^{(k)}}\left[\ell(b_k, Z_t^{(k)}) - \ell(b_k^*, Z_t^{(k)})|D_s^m, D_t^{k-1}\right] \tag{134}$$

$$= \sum_{k=1}^n \mathbb{E}_{D_s^m, D_t^{k-1}}\int\left(\ell(b_k, Z_t^{(k)}) - \ell(b_k^*, Z_t^{(k)})\right)P_{\theta_s^*,\theta_t^*}(Z_t^{(k)}|D_s^m, D_t^{k-1})dx_t^{(k)} \tag{135}$$

$$= \sum_{k=1}^n \mathbb{E}_{D_s^m, D_t^{k-1}}\int\left(\ell(b_k, Z_t^{(k)}) - \ell(b_k^*, Z_t^{(k)})\right)(P_{\theta_s^*,\theta_t^*}(Z_t^{(k)}|D_s^m, D_t^{k-1}) - Q(Z_t^{(k)}|D_s^m, D_t^{k-1}) \tag{136}$$

$$+ Q(Z_t^{(k)}|D_s^m, D_t^{k-1}))dx_t^{(k)} \tag{137}$$

$$\overset{(a)}{\leq} \sum_{k=1}^{n} \mathbb{E}_{D_s^m, D_t^{k-1}} \int \left(\ell(b_k, Z_t^{(k)}) - \ell(b_k^*, Z_t^{(k)}) \right) (P_{\theta_s^*, \theta_t^*}(Z_t^{(k)} | D_s^m, D_t^{k-1}) - Q(Z_t^{(k)} | D_s^m, D_t^{k-1})) dx_t^{(k)} \tag{138}$$

$$\overset{(b)}{\leq} \sum_{k=1}^{n} \mathbb{E}_{D_s^m, D_t^{k-1}} M \int (P_{\theta_s^*, \theta_t^*}(Z_t^{(k)} | D_s^m, D_t^{k-1}) - Q(Z_t^{(k)} | D_s^m, D_t^{k-1})) dx_t^{(k)} \tag{139}$$

$$\overset{(c)}{\leq} \sum_{k=1}^{n} \mathbb{E}_{D_s^m, D_t^{k-1}} M \sqrt{2D\left(P_{\theta_s^*, \theta_t^*}(Z_t^{(k)} | D_s^m, D_t^{k-1}) \| Q(Z_t^{(k)} | D_s^m, D_t^{k-1}) \right)} \tag{140}$$

$$\overset{(d)}{\leq} M \sum_{k=1}^{n} \sqrt{2\mathbb{E}_{D_s^m, D_t^{k-1}} D\left(P_{\theta_s^*, \theta_t^*}(Z_t^{(k)} | D_s^m, D_t^{k-1}) \| Q(Z_t^{(k)} | D_s^m, D_t^{k-1}) \right)} \tag{141}$$

$$\overset{(e)}{=} M \sum_{k=1}^{n} \sqrt{2D\left(P_{\theta_t^*} \| Q | D_s^m, D_t^{k-1} \right)} \tag{142}$$

$$\overset{(f)}{\leq} Mn \sqrt{\frac{2}{n} \sum_{k=1}^{n} D\left(P_{\theta_t^*} \| Q | D_s^m, D_t^{k-1} \right)} \tag{143}$$

$$\overset{(g)}{=} M \sqrt{2nD(P_{\theta_t^*}(D_t^n) \| Q(D_t^n | D_s^m))} \tag{144}$$

$$= M \sqrt{2nI(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_s^m)}, \tag{145}$$

where in (a) we use the definition of Q, then (b) holds since we assume the loss function is bounded, (c) follows from the Pinsker's inequality, (d) and (f) follows from the Jensen's inequality, (g) holds because of the chain rule of the KL divergence.

A.6. Proof of Theorem 7

Proof. We firstly show that given any prior over Θ_s, $\Theta_{t,i-1}$ and $\Theta_{t,i}$ for any episode i, we have

$$I(D_{t,i}^{n_i}; \Theta_{t,i}, \Theta_{t,i-1}, \Theta_s | D_s^m, D_{t,i-1}^{n_{i-1}}) \tag{146}$$

$$= I(\Theta_{t,i}, \Theta_{t,i-1}, \Theta_s; D_{t,i}^{n_i}, D_{t,i-1}^{n_{i-1}}, D_s^m) - I(\Theta_{t,i-1}, \Theta_s; D_{t,i-1}^{n_{i-1}}, D_s^m) \tag{147}$$

$$= D(P_{\Theta_{t,i}, \Theta_{t,i-1}, \Theta_s}(D_{t,i}^{n_i}, D_{t,i-1}^{n_{i-1}}, D_s^m) \| Q(D_{t,i}^{n_i}, D_{t,i-1}^{n_{i-1}}, D_s^m)) - D(P_{\Theta_{t,i-1}, \Theta_s}(D_{t,i-1}^{n_{i-1}}, D_s^m) \| Q(D_{t,i-1}^{n_{i-1}}, D_s^m)) \tag{148}$$

$$= \int \left(\mathbb{E}_{\theta_s, \theta_{t,i}, \theta_{t,i-1}} \left[\log \frac{P_{\theta_s, \theta_{t,i}, \theta_{t,i-1}}(D_{t,i}^{n_i}, D_{t,i-1}^{n_{i-1}}, D_s^m)}{Q(D_{t,i}^{n_i}, D_{t,i-1}^{n_{i-1}}, D_s^m)} \right] \right. \tag{149}$$

$$\left. - \mathbb{E}_{\theta_s, \theta_{t,i-1}} \left[\log \frac{P_{\theta_s, \theta_{t,i-1}}(D_{t,i-1}^{n_{i-1}}, D_s^m)}{Q(D_{t,i-1}^{n_{i-1}}, D_s^m)} \right] \right) \omega(\theta_s, \theta_{t,i}, \theta_{t,i-1}) d\theta_s d\theta_{t,i-1} d\theta_{t,i} \tag{150}$$

$$= \int \left(\mathbb{E}_{\theta_s, \theta_{t,i}, \theta_{t,i-1}} \left[\log \frac{P_{\theta_t}(D_{t,i}^{n_i})}{Q(D_{t,i}^{n_i} | D_{t,i-1}^{n_{i-1}}, D_s^m)} \right] \right) \omega(\theta_s, \theta_{t,i}, \theta_{t,i-1}) d\theta_s d\theta_{t,i-1} d\theta_{t,i}, \tag{151}$$

where in the last equality we use the chain rule and the Assumption 2. Then the expected regret till episode l can be expressed by the conditional mutual information evaluated at $\Theta_s = \theta_s^*$, $\Theta_{t,i-1} = \theta_{t,i-1}^*$ and $\Theta_{t,i} = \theta_{t,i}^*$ at each episode i:

$$\mathcal{R}_{TV} = \sum_{i=1}^{l} I(D_{t,i}^{n_i}; \Theta_{t,i} = \theta_{t,i}^*, \Theta_{t,i-1} = \theta_{t,i-1}^*, \Theta_s = \theta_s^* | D_s^m, D_{t,i-1}^{n_{i-1}}). \tag{152}$$

A.7. Proof of Theorem 8

Proof. Under the conditions from Theorem 5, we firstly give an upper bound of the expectation term at each episode i as

$$\mathbb{E}_{\theta_s^*, \theta_{t,i}^*, \theta_{t,i-1}^*} \left[\sum_{i=1}^{n_i} \ell\left(b_i, Z_{t,i}^{(i)}\right) - \sum_{i=1}^{n_i} \ell(b_i^*, Z_{t,i}^{(i)}) \right] \leq M \sqrt{2n_i I(D_{t,i}^{n_i}; \Theta_s = \theta_s^*, \Theta_{t,i-1} = \theta_{t,i-1}^*, \Theta_{t,i} = \theta_{t,i}^* | D_s^m, D_t^{i-1})}, \tag{153}$$

where we define the conditional mutual information as

$$I(D_{t,i}^{n_i};\Theta_s=\theta_s^*,\Theta_{t,i-1}=\theta_{t,i-1}^*,\Theta_{t,i}=\theta_{t,i}^*|D_s^m,D_t^{i-1}):=\mathbb{E}_{\theta_s,\theta_{t,i},\theta_{t,i-1}}\left[\log\frac{P_{\theta_t}(D_{t,i}^{n_i})}{Q(D_{t,i}^{n_i}|D_{t,i-1}^{n_{i-1}},D_s^m)}\right] \tag{154}$$

with the mixture strategy Q. By Cauchy-Schwarz inequality,

$$\mathcal{R}_{TV}=\sum_{i=1}^{l}\mathbb{E}_{\theta_s^*,\theta_{t,i}^*,\theta_{t,i-1}^*}\left[\sum_{k=1}^{n_i}\ell\left(b_{k,i},Z_{t,i}^{(k)}\right)-\sum_{k=1}^{n_i}\ell(b_{k,i}^*,Z_{t,i}^{(k)})\right] \tag{155}$$

$$\leq\sum_{i=1}^{l}M\sqrt{2n_iI(D_{t,i}^{n_i};\Theta_s=\theta_s^*,\Theta_{t,i-1}=\theta_{t,i-1}^*,\Theta_{t,i}=\theta_{t,i}^*|D_s^m,D_t^{i-1})} \tag{156}$$

$$\leq M\sqrt{2l\sum_{i=1}^{l}n_iI(D_{t,i}^{n_i};\Theta_s=\theta_s^*,\Theta_{t,i-1}=\theta_{t,i-1}^*,\Theta_{t,i}=\theta_{t,i}^*|D_s^m,D_t^{i-1})}, \tag{157}$$

which complete the proof.

Before proving Theorem 9 and Theorem 10, let us prove the results for the OTL case first.

A.8. Proof of Theorem 11

Proof. We give the approximation on the KL divergence to see how the prior will affect the divergence,

$$\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right]=\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)P_{\theta_s^*}(D_s^m)Q(D_s^m)}{Q(D_t^n,D_s^m)P_{\theta_s^*}(D_s^m)}\right] \tag{158}$$

$$=\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*,\theta_s^*}(D_t^n,D_s^m)}{Q(D_t^n,D_s^m)}\right]-\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_s^*}(D_s^m)}{Q(D_s^m)}\right] \tag{159}$$

$$=D(P_{\theta_t^*,\theta_s^*}(D_t^n,D_s^m)\|Q(D_t^n,D_s^m))-D(P_{\theta_s^*}(D_s^m)\|Q(D_s^m)). \tag{160}$$

We can view that source samples and target samples are jointly sampled given the distribution $P_{\theta_s^*}$ and $P_{\theta_t^*}$. Using the results in [10] and [9], with the proper prior $\omega(\theta_s,\theta_t)$ and parametric conditions, the asymptotic normality of the posterior implies that

$$D(P_{\theta_t^*,\theta_s^*}(D_t^n,D_s^m)\|Q(D_t^n,D_s^m))-\frac{1}{2}\log\det\left(\begin{bmatrix}nI_t(\theta_t^*) & 0\\ 0 & mI_s(\theta_s^*)\end{bmatrix}\right)\to\log\frac{1}{2\pi e}+\log\frac{1}{\omega(\theta_s^*,\theta_t^*)}, \tag{161}$$

as both n and m goes to infinity, where the fisher information matrices are denoted by

$$I_t(\theta_t^*)=-\mathbb{E}_{\theta_t^*}\left[\frac{\partial\log P(x|\theta_t^*)}{\partial\theta_t^2}\right], \tag{162}$$

$$I_s(\theta_s^*)=-\mathbb{E}_{\theta_s^*}\left[\frac{\partial\log P(x|\theta_s^*)}{\partial\theta_s^2}\right]. \tag{163}$$

Similarly,

$$D(P_{\theta_s^*}(D_s^m)\|Q(D_s^m))-\frac{1}{2}\log\det(mI_s(\theta_s^*))\to\frac{1}{2}\log\frac{1}{2\pi e}+\log\frac{1}{\omega(\theta_s^*)}, \tag{164}$$

as m goes to infinity. Therefore,

$$\lim_{n,m\to\infty}\left(\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right]\right)=\frac{1}{2}\log\det\left(\begin{bmatrix}nI_t(\theta_t^*) & 0\\ 0 & \alpha nI_s(\theta_s^*)\end{bmatrix}\right)+\log\frac{1}{2\pi e}+\log\frac{1}{\omega(\theta_s^*,\theta_t^*)} \tag{165}$$

$$-\frac{1}{2}\log\det(mI_s(\theta_s^*))-\frac{1}{2}\log\frac{1}{2\pi e}-\log\frac{1}{\omega(\theta_s^*)} \tag{166}$$

$$=\frac{1}{2}\log\det(nI_t(\theta_t^*))+\frac{1}{2}\log\frac{1}{2\pi e}+\log\frac{1}{\omega(\theta_t^*|\theta_s^*)}. \tag{167}$$

To conclude, as both n and m goes to infinity, the conditional mutual information will converge to

$$I(D_t^n;\Theta_t=\theta_t^*,\Theta_s=\theta_s^*|D_s^m)-\frac{1}{2}\log\frac{n}{2\pi e}\to\frac{1}{2}\log I_t(\theta_t^*)+\log\frac{1}{\omega(\theta_t^*|\theta_s^*)}.$$

A.9. Proof of Theorem 12

Proof. By writing $\theta_s = (\theta_c, \theta_{sr})$ and $\theta_t = (\theta_c, \Theta_{tr,i})$, let us rewrite the conditional mutual information as

$$\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right] = \mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)P_{\theta_s^*}(D_s^m)Q(D_s^m)}{Q(D_t^n,D_s^m)P_{\theta_s^*}(D_s^m)}\right] \tag{168}$$

$$= \mathbb{E}_{\theta_c^*,\theta_{sr}^*,\theta_{tr}^*}\left[\log\frac{P_{\theta_t^*,\theta_s^*}(D_t^n,D_s^m)}{Q(D_t^n,D_s^m)}\right] - \mathbb{E}_{\theta_c^*,\theta_{sr}^*}\left[\log\frac{P_{\theta_s^*}(D_s^m)}{Q(D_s^m)}\right] \tag{169}$$

$$= D(P_{\theta_c^*,\theta_{sr}^*,\theta_{tr}^*}(D_t^n,D_s^m)\|P(D_t^n,D_s^m)) - D(P_{\theta_c^*,\theta_{sr}^*}(D_s^m)\|Q(D_s^m)). \tag{170}$$

We view that m source samples and n target samples are jointly sampled from the distribution parametrized by the parameters $\boldsymbol{\Theta} = (\Theta_c, \Theta_{sr}, \Theta_{tr,i})$. At time n, we have the asymptotic approximation under the proper prior and assumption 2 using Theorem 2.1 in [10] and [9] as

$$D(P_{\theta_c^*,\theta_{sr}^*,\theta_{tr}^*}(D_t^n,D_s^m)\|P(D_t^n,D_s^m)) - \frac{1}{2}\log\det(\mathbf{I}_{\boldsymbol{\theta}^*}) \to \frac{2d-j}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_s^*,\theta_t^*)}, \tag{171}$$

where the Fisher information matrix is defined as

$$\mathbf{I}_{\boldsymbol{\theta}^*} = \begin{bmatrix} mI_{cs}(\theta_c^*) + nI_{ct}(\theta_c^*) & mI_{cs}(\theta_c^*,\theta_{sr}^*) & nI_{ct}(\theta_c^*,\theta_{tr}^*) \\ mI_{cs}^T(\theta_c^*,\theta_{sr}^*) & mI_s(\theta_{sr}^*) & \mathbf{0} \\ nI_{ct}^T(\theta_c^*,\theta_{tr}^*) & \mathbf{0} & nI_t(\theta_{tr}^*) \end{bmatrix} \tag{172}$$

Similarly,

$$D(P_{\theta_c^*,\theta_{sr}^*}(D_s^m)\|Q(D_s^m)) - \frac{1}{2}\log\det\left(\mathbf{I}_{\boldsymbol{\theta}_s^*}\right) \to \frac{d}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_s^*)}, \tag{173}$$

where

$$\mathbf{I}_{\boldsymbol{\theta}_s^*} = m\begin{bmatrix} I_{cs}(\theta_c^*) & I_{cs}(\theta_c^*,\theta_{sr}^*) \\ I_{cs}^T(\theta_c^*,\theta_{sr}^*) & I_s(\theta_{sr}^*) \end{bmatrix} \tag{174}$$

as m goes to sufficiently large. As a consequence,

$$\lim_{n,m\to\infty}\left(\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right]\right) = \frac{1}{2}\log\det(\mathbf{I}_{\boldsymbol{\theta}^*}) + \frac{2d-j}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_s^*,\theta_t^*)} \tag{175}$$

$$- \frac{1}{2}\log\det(\mathbf{I}_{\boldsymbol{\theta}_s^*}) - \frac{d}{2}\log\frac{1}{2\pi e} - \log\frac{1}{\omega(\theta_s^*)} \tag{176}$$

$$= \frac{1}{2}\log\frac{\det(\mathbf{I}_{\boldsymbol{\theta}^*})}{\det(\mathbf{I}_{\boldsymbol{\theta}_s^*})} + \frac{d-j}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)}. \tag{177}$$

Let us examine the ratio of the determinant, using the block determinant results from [46],

$$\log\frac{\det(\mathbf{I}_{\boldsymbol{\theta}^*})}{\det(\mathbf{I}_{\boldsymbol{\theta}_s^*})} = \log\det\Bigg(mI_{cs}(\theta_c^*) + nI_{ct}(\theta_c^*) - \begin{bmatrix} mI_{cs}(\theta_c^*,\theta_{sr}^*) & nI_{ct}(\theta_c^*,\theta_{tr}^*) \end{bmatrix}\begin{bmatrix} mI_s(\theta_{sr}^*) & \mathbf{0} \\ \mathbf{0} & nI_t(\theta_{tr}^*) \end{bmatrix}^{-1}\begin{bmatrix} mI_{cs}^T(\theta_c^*,\theta_{sr}^*) \\ nI_{ct}^T(\theta_c^*,\theta_{tr}^*) \end{bmatrix}\Bigg)$$

$$+ \log\det\left(\begin{bmatrix} mI_s(\theta_{sr}^*) & \mathbf{0} \\ \mathbf{0} & nI_t(\theta_{tr}^*) \end{bmatrix}\right) - \log\det\left(m(I_{cs}(\theta_c^*) - I_{cs}(\theta_c^*,\theta_{sr}^*)I_s^{-1}(\theta_{sr}^*)I_{cs}^T(\theta_c^*,\theta_{sr}^*))\right)$$

$$- \log\det\left(mI_s(\theta_{sr}^*)\right) \tag{178}$$

$$= \log\det(m\Delta_s + n\Delta_t) - \log\det(m\Delta_s) + \log\det(nI_t(\theta_{tr}^*)) \tag{179}$$

$$= \log\det(\mathbf{I}_j + \frac{n}{m}\Delta_t\Delta_s^{-1}) + \log\det(nI_t(\theta_{tr}^*)), \tag{180}$$

where we define $\Delta_s = I_{cs}(\theta_c^*) - I_{cs}(\theta_c^*,\theta_{sr}^*)I_s^{-1}(\theta_{sr}^*)I_{cs}^T(\theta_c^*,\theta_{sr}^*)$ and $\Delta_t = I_{ct}(\theta_c^*) - I_{ct}(\theta_c^*,\theta_{tr}^*)I_t^{-1}(\theta_{tr}^*)I_{ct}^T(\theta_c^*,\theta_{tr}^*)$. $\mathbf{I}_j$ denotes the identity matrix with size j and $\boldsymbol{\theta}^* = (\theta_c^*,\theta_{sr}^*,\theta_{tr}^*)$ denotes the true parameters. With a little abuse of notation, we define the Fisher information matrices as

$$I_{cs}(\theta_c^*) = -\mathbb{E}_{\theta_s^*}\left[\nabla^2_{\Theta_c} \log P(Z_s|\Theta_c, \theta_{sr}^*)\right]\Big|_{\Theta_c=\theta_c^*} \in \mathbb{R}^{j\times j}, \tag{181}$$

$$I_{ct}(\theta_c^*) = -\mathbb{E}_{\theta_t^*}\left[\nabla^2_{\Theta_c} \log P(Z_t|\Theta_c, \theta_{tr}^*)\right]\Big|_{\Theta_c=\theta_c^*} \in \mathbb{R}^{j\times j}, \tag{182}$$

$$I_{s}(\theta_{sr}^*) = -\mathbb{E}_{\theta_s^*}\left[\nabla^2_{\Theta_{sr}} \log P(Z_s|\theta_c^*, \Theta_{sr})\right]\Big|_{\Theta_{sr}=\theta_{sr}^*} \in \mathbb{R}^{(d-j)\times(d-j)}, \tag{183}$$

$$I_{t}(\theta_{tr}^*) = -\mathbb{E}_{\theta_t^*}\left[\nabla^2_{\Theta_{tr}} \log P(Z_t|\theta_c^*, \Theta_{tr})\right]\Big|_{\Theta_{tr}=\theta_{tr}^*} \in \mathbb{R}^{(d-j)\times(d-j)}, \tag{184}$$

$$I_{cs}(\theta_c^*, \theta_{sr}^*) = -\mathbb{E}_{\theta_s^*}\left[\frac{\partial \log P(Z_s|\theta_c^*, \theta_{sr}^*)}{\partial\Theta_{c,i}\partial\Theta_{sr,k}}\right]_{\substack{i=1,\cdots,j,\\ k=1,\cdots,d-j}} \in \mathbb{R}^{j\times(d-j)}, \tag{185}$$

$$I_{ct}(\theta_c^*, \theta_{tr}^*) = -\mathbb{E}_{\theta_t^*}\left[\frac{\partial \log P(Z_t|\theta_c^*, \theta_{tr}^*)}{\partial\Theta_{c,i}\partial\Theta_{tr,k}}\right]_{\substack{i=1,\cdots,j,\\ k=1,\cdots,d-j}} \in \mathbb{R}^{j\times(d-j)}. \tag{186}$$

Putting everything together, by setting $m = cn^p$ we reach

$$I(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m) - \frac{1}{2}\log\det(nI_t(\theta_{tr}^*)) - \frac{1}{2}\log\det(\mathbf{I}_j + \frac{1}{cn^{p-1}}\Delta_t\Delta_s^{-1}) \to \frac{d-j}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)}, \tag{187}$$

as both n and m goes to infinity.

Now we can use the results of Theorem 4 and Theorem 5 to proof Theorem 9 and Theorem 10 for ITL case.

A.10. Proof of Theorem 9

Proof. From Theorem 11, we have that

$$I(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m) - \frac{1}{2}\log\frac{n}{2\pi e} \to \frac{1}{2}\log I_t(\theta_t^*) + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)}. \tag{188}$$

By the i.i.d. property, additionally we have

$$I(Z_t', D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m) - \frac{1}{2}\log\frac{n+1}{2\pi e} \to \frac{1}{2}\log I_t(\theta_t^*) + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)}. \tag{189}$$

Therefore as both m and n go to sufficiently large, the instantaneous prediction yields

$$I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_t^n, D_s^m) = I(Z_t', D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m) - I(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m) \tag{190}$$

$$= \frac{1}{2}\log\frac{n+1}{2\pi e} - \frac{1}{2}\log\frac{n}{2\pi e} \tag{191}$$

$$= \frac{1}{2}\log(1+\frac{1}{n}) \tag{192}$$

$$\asymp \frac{1}{n}, \tag{193}$$

which is the typical result for the optimal rate of $O(\frac{1}{n})$.

A.11. Proof of Theorem 10

Proof. From Theorem 12, as we assume $m = cn^p$ for some positive constant c, the asymptotic approximation of the expected regret is expressed as

$$I(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^*|D_s^m) - \frac{1}{2}\log\det(nI_t(\theta_{tr}^*)) - \frac{1}{2}\log\det(\mathbf{I}_j + \frac{n}{cn^p}\Delta_t\Delta_s^{-1}) \to \frac{d-j}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)}. \tag{194}$$

If we take Z_t' take into consideration, we similarly have

$$I(Z_t', D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_s^m) - \frac{1}{2}\log\det((n+1)I_t(\theta_{tr}^*)) - \frac{1}{2}\log\det(\mathbf{I}_j + \frac{n+1}{cn^p}\Delta_t\Delta_s^{-1}) \tag{195}$$

$$\to \frac{d-j}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta_t^*|\theta_s^*)}. \tag{196}$$

As a consequence, when both n and m go to infinity,

$$I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_t^n, D_s^m) = I(Z_t', D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_s^m) - I(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_s^m) \tag{197}$$

$$= \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\log\det(cn^p\Delta_s + (n+1)\Delta_t) - \frac{1}{2}\log\det(cn^p\Delta_s + n\Delta_t) \tag{198}$$

We will use the expansion of determinant:

$$\det(\mathbf{I} + \frac{1}{n^k}A) = 1 + \frac{1}{n^k}\mathrm{Tr}(A) + o(1/n^k). \tag{199}$$

For $0 \le p < 1$:

$$I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_t^n, D_s^m) = \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\log\det(cn^p\Delta_s + (n+1)\Delta_t) - \frac{1}{2}\log\det(cn^p\Delta_s + n\Delta_t) \tag{200}$$

$$= \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\log\det(\mathbf{I}_j + \frac{cn^p}{n+1}\Delta_s\Delta_t^{-1}) - \frac{1}{2}\log\det(\mathbf{I}_j + \frac{cn^p}{n}\Delta_s\Delta_t^{-1}) \tag{201}$$

$$= \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\frac{cn^p}{n+1}\mathrm{Tr}(\Delta_s\Delta_t^{-1}) - \frac{1}{2}\frac{cn^p}{n}\mathrm{Tr}(\Delta_s\Delta_t^{-1}) + o(\frac{1}{n}) \tag{202}$$

$$= \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\frac{cn^p}{n(n+1)}\mathrm{Tr}(\Delta_s\Delta_t^{-1}) + o(\frac{1}{n}) \tag{203}$$

$$\asymp \frac{d}{n} \tag{204}$$

For $p \ge 1$:

$$I(Z_t'; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_t^n, D_s^m) = \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\log\det(cn^p\Delta_s + (n+1)\Delta_t) - \frac{1}{2}\log\det(cn^p\Delta_s + n\Delta_t) \tag{205}$$

$$= \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\log\det(\mathbf{I}_j + \frac{n+1}{cn^p}\Delta_t\Delta_s^{-1}) - \frac{1}{2}\log\det(\mathbf{I}_j + \frac{n}{cn^p}\Delta_t\Delta_s^{-1}) \tag{206}$$

$$= \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{1}{2}\frac{n+1}{cn^p}\mathrm{Tr}(\Delta_t\Delta_s^{-1}) - \frac{1}{2}\frac{n}{cn^p}\mathrm{Tr}(\Delta_t\Delta_s^{-1}) + o(\frac{1}{n^p}) \tag{207}$$

$$= \frac{d-j}{2}\log(1+\frac{1}{n}) + \frac{1}{2cn^p}\mathrm{Tr}(\Delta_t\Delta_s^{-1}) + o(\frac{1}{n^p}) \tag{208}$$

$$\asymp \frac{d-j}{n} + \frac{j}{n^p} \tag{209}$$

To conclude,

$$\mathcal{R}_I \asymp \frac{d-j}{n} + \frac{j}{n \vee n^p}, \tag{210}$$

which completes the proof.

A.12. Proof of Corollary 1

Proof. From Theorem 12, we have

$$I(D_t^n; \Theta_t = \theta_t^*, \Theta_s = \theta_s^* | D_s^m) = \frac{1}{2}\log\det(nI_t(\theta_{tr}^*)) + \frac{1}{2}\log\det(\mathbf{I}_j + \frac{n}{m}\Delta_t\Delta_s^{-1}) + O(1). \tag{211}$$

This quantity depends on the source sample size $m = cn^p$, assume $m = cn^p$ with some positive constant c. For $0 \le p < 1$:

$$I(D_t^n;\Theta_t=\theta_t^*,\Theta_s=\theta_s^*|D_s^m)=\frac{1}{2}\log\det(nI_t(\theta_{tr}^*))+\frac{1}{2}\log\det(cn^p\Delta_s+n\Delta_t)-\frac{1}{2}\log\det(cn^p\Delta_s)+O(1) \tag{212}$$

$$=\frac{1}{2}\log\det(nI_t(\theta_{tr}^*))+\frac{j}{2}\log n-\frac{jp}{2}\log n+\frac{1}{2}\frac{1}{n^{1-p}}\mathrm{Tr}(c\Delta_s\Delta_t^{-1})-\frac{1}{2}\log\det(c\Delta_s)+O(1) \tag{213}$$

$$\asymp(d-j)\log n+j(1-p)\log n. \tag{214}$$

For $p\geq 1$:

$$I(D_t^n;\Theta_t=\theta_t^*,\Theta_s=\theta_s^*|D_s^m)=\frac{1}{2}\log\det(nI_t(\theta_{tr}^*))+\frac{1}{2}\log\det(cn^p\Delta_s+n\Delta_t)-\frac{1}{2}\log\det(cn^p\Delta_s)+O(1) \tag{215}$$

$$=\frac{1}{2}\log\det(nI_t(\theta_{tr}^*))+\frac{1}{2}\log\det(\mathbf{I}_j+\frac{1}{cn^{p-1}}\Delta_t\Delta_s^{-1})+O(1) \tag{216}$$

$$=\frac{1}{2}\log\det(nI_t(\theta_{tr}^*))+\frac{1}{2}\frac{1}{cn^{p-1}}\mathrm{Tr}(\Delta_t\Delta_s^{-1})+O(1)+o(\frac{1}{n^{p-1}}) \tag{217}$$

$$\asymp(d-j)\log n+\frac{j}{n^{p-1}}, \tag{218}$$

which completes the proof.

A.13. Proof of Theorem 13

Proof. Let us give an asymptotic estimation on $I(D_{t,i}^{n_i};\Theta_{t,i}=\theta_{t,i}^*|D_s^m,D_{t,i}^{n_i})$ as all n_i, n_{i-1} and m are large enough shown in the following theorem.

Theorem 14. *Under Assumptions 1 and 2, with $\Theta_s,\Theta_{t,i-1}.\Theta_{t,i}\in\mathbb{R}^d$ defined in the paper and as $n_{i-1},n_i,m\to\infty$, the mixture strategy with proper prior $\omega(\Theta_s,\Theta_{t,i-1},\Theta_{t,i})$ yields*

$$\begin{aligned}I(D_{t,i}^{n_i};\Theta_{t,i}=\theta_{t,i}^*|D_s^m,D_{t,i-1}^{n_{i-1}})&-\log\det\left(\mathbf{I}_{j_i\times j_i}+\frac{n_i}{m+n_{i-1}}\Delta_{ct,i}\Delta_{cst,i}^{-1}\right)\\&-\log\det(\mathbf{I}_{c_i\times c_i}+\frac{n_i}{n_{i-1}}\Delta_{t,i}\Delta_{t,i-1}^{-1})-\log\det(n_iI_{t,i}(\theta_{tr,i}^*))\\&\to(d-j_i-c_i)\log\frac{1}{2\pi e}+\frac{2}{\omega(\theta_{t,i}^*|\theta_{t,i-1}^*,\theta_s^*)}.\end{aligned} \tag{219}$$

Proof. Rewrite the conditional mutual information in terms of the KL divergence as

$$I(D_{t,i}^{n_i};\Theta_{t,i}=\theta_{t,i}^*|D_s^m,D_{t,i-1}^{n_{i-1}})=\mathbb{E}_{\theta_s^*,\theta_{t,i}^*,\theta_{t,i-1}^*}\left[\log\frac{P_{\theta_{t,i}^*}(D_{t,i}^{n_i})}{Q(D_{t,i}^{n_i}|D_{t,i-1}^{n_{i-1}},D_s^m)}\right] \tag{220}$$

$$=D(P_{\theta_s^*,\theta_{t,i}^*,\theta_{t,i-1}^*}(D_s^m,D_{t,i-1}^{n_{i-1}},D_{t,i}^{n_i})\|Q(D_s^m,D_{t,i-1}^{n_{i-1}},D_{t,i}^{n_i}))-D(P_{\theta_s^*,\theta_{t,i-1}^*}(D_s^m,D_{t,i-1}^{n_{i-1}})\|Q(D_s^m,D_{t,i-1}^{n_{i-1}})). \tag{221}$$

Let us align the parameter as $\boldsymbol{\Theta}=(\Theta_{c,i},\Theta_{v,i},\Theta_{sr,i},\Theta_{tr,i-1},\Theta_{tr,i})$ and $\boldsymbol{\Theta}_s=(\Theta_{c,i},\Theta_{v,i},\Theta_{sr,i},\Theta_{tr,i-1})$ and we define a set of Fisher information matrices as

$$I_{cs,i}(\theta_{c,i}^*)=-\mathbb{E}_{\theta_s^*}\left[\nabla^2_{\Theta_{c,i}}\log P(Z_s|\Theta_{c,i},\theta_{sr,i}^*)\right]\Big|_{\Theta_{c,i}=\theta_{c,i}^*}\in\mathbb{R}^{j_i\times j_i}, \tag{222}$$

$$I_{s,i}(\theta_{sr,i}^*)=-\mathbb{E}_{\theta_s^*}\left[\nabla^2_{\Theta_{sr,i}}\log P(Z_s|\theta_{c,i}^*,\Theta_{sr,i})\right]\Big|_{\Theta_{sr,i}=\theta_{sr,i}^*}\in\mathbb{R}^{(d-j_i)\times(d-j_i)}, \tag{223}$$

$$I_{ct,i-1}(\theta_{c,i}^*)=-\mathbb{E}_{\theta_{t,i-1}^*}\left[\nabla^2_{\Theta_{c,i}}\log P(Z_{t,i-1}|\Theta_{c,i},\theta_{v,i}^*,\theta_{tr,i-1}^*)\right]\Big|_{\Theta_{c,i}=\theta_{c,i}^*}\in\mathbb{R}^{j_i\times j_i}, \tag{224}$$

$$I_{ct,i}(\theta_{c,i}^*)=-\mathbb{E}_{\theta_t^*}\left[\nabla^2_{\Theta_{c,i}}\log P(Z_{t,i}|\Theta_{c,i},\theta_{v,i}^*,\theta_{tr,i}^*)\right]\Big|_{\Theta_{c,i}=\theta_{c,i}^*}\in\mathbb{R}^{j_i\times j_i}, \tag{225}$$

$$I_{t,i-1}(\theta_{tr,i-1}^*)=-\mathbb{E}_{\theta_{t,i-1}^*}\left[\nabla^2_{\Theta_{tr,i-1}}\log P(Z_{t,i-1}|\theta_{c,i}^*,\theta_{v,i}^*,\Theta_{tr,i-1})\right]\Big|_{\Theta_{tr,i-1}=\theta_{tr,i-1}^*}\in\mathbb{R}^{(d-c_i-j_i)\times(d-c_i-j_i)}, \tag{226}$$

$$I_{t,i}(\theta_{tr,i}^*)=-\mathbb{E}_{\theta_t^*}\left[\nabla^2_{\Theta_{tr,i}}\log P(Z_{t,i}|\theta_{c,i}^*,\Theta_{tr,i})\right]\Big|_{\Theta_{tr,i}=\theta_{tr}^*}\in\mathbb{R}^{(d-c_i-j_i)\times(d-c_i-j_i)}, \tag{227}$$

$$I_{v_{i-1}}(\theta_{v,i}^*)=-\mathbb{E}_{\theta_{t,i-1}^*}\left[\nabla^2_{\Theta_{v,i}}\log P(Z_{t,i-1}|\theta_{c,i}^*,\Theta_{v,i},\theta_{tr,i-1}^*)\right]\Big|_{\Theta_{v,i}=\theta_{v,i}^*}\in\mathbb{R}^{c_i\times c_i}, \tag{228}$$

$$I_{v_i}(\theta^*_{v,i}) = -\mathbb{E}_{\theta^*_{t,i}}\left[\nabla^2_{\Theta_{v,i}} \log P(Z_{t,i}|\theta^*_{c,i}, \Theta_{v,i}, \theta^*_{tr,i})\right]\Big|_{\Theta_{v,i}=\theta^*_{v,i}} \in \mathbb{R}^{c_i\times c_i}, \tag{229}$$

$$I_{sc}(\theta^*_{c,i}, \theta^*_{sr,i}) = -\mathbb{E}_{\theta^*_s}\left[\frac{\partial \log P(Z_s|\theta^*_{c,i}, \theta^*_{sr,i})}{\partial\Theta^p_{c,i}\partial\Theta^q_{sr,i}}\right]_{\substack{p=1,\cdots,j_i,\\ q=1,\cdots,d-j_i}} \in \mathbb{R}^{j_i\times(d-j_i)}, \tag{230}$$

$$I_{ctr,i-1}(\theta^*_{c,i}, \theta^*_{tr,i-1}) = -\mathbb{E}_{\theta^*_{t,i-1}}\left[\frac{\partial \log P(Z_{t,i-1}|\Theta_{c,i}=\theta^*_{c,i}, \theta^*_{v,i}, \Theta_{tr,i-1}=\theta^*_{tr,i-1})}{\partial\Theta^p_{c,i}\partial\Theta^q_{tr,i-1}}\right]_{\substack{p=1,\cdots,j_i,\\ q=1,\cdots,d-c_i-j_i}}$$
$$\in \mathbb{R}^{j_i\times(d-c_i-j_i)}, \tag{231}$$

$$I_{ctr,i}(\theta^*_c, \theta^*_{tr,i}) = -\mathbb{E}_{\theta^*_t}\left[\frac{\partial \log P(Z_{t,i}|\Theta_{c,i}=\theta^*_{c,i}, \theta^*_{v,i}, \Theta_{tr,i}=\theta^*_{tr,i})}{\partial\Theta^p_{c,i}\partial\Theta^q_{tr,i}}\right]_{\substack{p=1,\cdots,j_i,\\ q=1,\cdots,d-c_i-j_i}}$$
$$\in \mathbb{R}^{j_i\times(d-c_i-j_i)}, \tag{232}$$

$$I_{vt,i-1}(\theta^*_{v,i}, \theta^*_{tr,i-1}) = -\mathbb{E}_{\theta^*_t}\left[\frac{\partial \log P(Z_{t,i-1}|\theta^*_{c,i}, \Theta_{v,i}=\theta^*_{v,i}, \Theta_{tr,i-1}=\theta^*_{tr,i-1})}{\partial\Theta^p_{v,i}\partial\Theta^q_{tr,i-1}}\right]_{\substack{p=1,\cdots,c_i,\\ q=1,\cdots,d-c_i-j_i}}$$
$$\in \mathbb{R}^{c_i\times(d-c_i-j_i)}, \tag{233}$$

$$I_{vt,i}(\theta^*_v, \theta^*_{tr,i}) = -\mathbb{E}_{\theta^*_t}\left[\frac{\partial \log P(Z_{t,i}|\theta^*_{c,i}, \Theta_{v,i}=\theta^*_{v,i}, \Theta_{tr,i}=\theta^*_{tr,i})}{\partial\Theta^p_{v,i}\partial\Theta^q_{tr,i}}\right]_{\substack{p=1,\cdots,c_i,\\ q=1,\cdots,d-c_i-j_i}} \in \mathbb{R}^{c_i\times(d-c_i-j_i)}, \tag{234}$$

$$I_{cv_{i-1}}(\theta^*_{c,i}, \theta^*_{v,i}) = -\mathbb{E}_{\theta^*_{t,i-1}}\left[\frac{\partial \log P(Z_{t,i-1}|\Theta_{c,i}=\theta^*_{c,i}, \Theta_{v,i}=\theta^*_{v,i}, \theta^*_{tr,i-1}=\theta^*_{tr,i-1})}{\partial\Theta^p_{v,i}\partial\Theta^q_{tr,i-1}}\right]_{\substack{p=1,\cdots,j_i,\\ q=1,\cdots,c_i}} \in \mathbb{R}^{j_i\times c_i}, \tag{235}$$

$$I_{cv_i}(\theta^*_{c,i}, \theta^*_{v,i}) = -\mathbb{E}_{\theta^*_t}\left[\frac{\partial \log P(Z_{t,i}|\Theta_{c,i}=\theta^*_{c,i}, \Theta_{v,i}=\theta^*_{v,i}, \theta^*_{tr,i})}{\partial\Theta^p_{v,i}\partial\Theta^q_{tr,i}}\right]_{\substack{p=1,\cdots,j_i,\\ q=1,\cdots,c_i}} \in \mathbb{R}^{j_i\times c_i}, \tag{236}$$

where Θ^p (Θ^q) denotes the pth (qth) element in Θ. To simplify the notations, we omit the function variables for all Fisher information matrices (for example, we write $I_{cs,i}(\theta^*_{c,i})$ as $I_{cs,i}$). Then the asymptotic normality implies that

$$D(P_{\theta^*_s,\theta^*_{t,i},\theta^*_{t,i-1}}(D^m_s, D^{n_{i-1}}_{t,i-1}, D^{n_i}_{t,i}) \| Q(D^m_s, D^{n_{i-1}}_{t,i-1}, D^{n_i}_{t,i})) - \frac{1}{2}\log\det(\mathbf{I}_{\boldsymbol{\theta}^*}) \to \frac{3d-2j_i-c_i}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta^*_s,\theta^*_t)}, \tag{237}$$

where the Fisher information matrix is defined as

$$\mathbf{I}_{\boldsymbol{\theta}^*} = \begin{bmatrix} mI_{cs,i} + n_{i-1}I_{ct,i-1} + n_iI_{ct,i} & n_iI_{cv_i} + n_{i-1}I_{cv_{i-1}} & mI_{sc,i} & n_{i-1}I_{ctr,i-1} & n_iI_{ctr,i} \\ n_iI^T_{ct,i} + n_{i-1}I^T_{ct,i-1} & n_iI_{v_i} + n_{i-1}I_{v_{i-1}} & \mathbf{0} & n_{i-1}I_{vt,i-1} & n_iI_{vt,i} \\ mI^T_{sc,i} & \mathbf{0} & mI_{s,i} & \mathbf{0} & \mathbf{0} \\ n_{i-1}I^T_{ctr,i-1} & n_{i-1}I^T_{vt,i-1} & \mathbf{0} & n_{i-1}I_{t,i-1} & \mathbf{0} \\ n_iI^T_{ctr,i} & n_iI^T_{vt,i} & \mathbf{0} & \mathbf{0} & n_iI_{t,i} \end{bmatrix}, \tag{238}$$

as all n_i, n_{i-1} and m go to infinity. Similarly,

$$D(P_{\theta^*_s,\theta^*_{t,i-1}}(D^m_s, D^{n_{i-1}}_{t,i-1}) \| Q(D^m_s, D^{n_{i-1}}_{t,i-1})) - \frac{1}{2}\log\det\left(\mathbf{I}_{\theta^*_s}\right) \to \frac{2d-j_i}{2}\log\frac{1}{2\pi e} + \log\frac{1}{\omega(\theta^*_s,\theta^*_t)}, \tag{239}$$

where

$$\mathbf{I}_{\theta^*_s} = \begin{bmatrix} mI_{cs,i} + n_{i-1}I_{ct,i-1} & n_{i-1}I_{cv_{i-1}} & mI_{sc,i} & n_{i-1}I_{ctr,i-1} \\ n_{i-1}I^T_{cv_i-1} & n_{i-1}I_{v_i-1} & \mathbf{0} & n_{i-1}I_{vt,i-1} \\ mI^T_{sc,i} & \mathbf{0} & mI_{s,i} & \mathbf{0} \\ n_{i-1}I^T_{ctr,i-1} & n_{i-1}I^T_{vt,i-1} & \mathbf{0} & n_{i-1}I_{t,i-1} \end{bmatrix}. \tag{240}$$

Then by subtraction, we have

$$\frac{\log\det(\mathbf{I}_{\boldsymbol{\theta}^*})}{\log\det(\mathbf{I}_{\boldsymbol{\theta}^*_s})} = \log\det\left(\mathbf{I}_{j_i\times j_i} + \frac{n_i}{m+n_{i-1}}\Delta_{ct,i}\Delta_{cst,i}^{-1}\right) + \log\det(\mathbf{I}_{c_i\times c_i} + \frac{n_i}{n_{i-1}}\Delta_{t,i}\Delta_{t,i-1}^{-1}) + \log\det(n_i I_{t,i}), \tag{241}$$

where $\Delta_{ct,i} = I_{ct,i} - I_{ctr,i}I_{t,i}^{-1}I_{ctr,i}^T$, $\Delta_{cst,i} = \frac{m}{m+n_{i-1}}(I_{cs} - I_{sc,i}I_{s,i}^{-1}I_{sc,i}^T) + \frac{n_{i-1}}{m+n_{i-1}}(I_{ct,i-1} - I_{ctr,i-1}I_{t,i-1}^{-1}I_{ctr,i-1}^T)$, $\Delta_{t,i} = I_{v_i} - I_{vt,i}I_{t,i}^{-1}I_{vt,i}^T$, and $\Delta_{t,i-1} = I_{v_{i-1}} - I_{vt,i-1}I_{t,i-1}^{-1}I_{vt,i-1}^T$. Then we have the asymptotic estimation as

$$\begin{aligned} I(D_{t,i}^{n_i};\Theta_{t,i}=\theta_{t,i}^*|D_s^m, D_{t,i-1}^{n_{i-1}}) \to &\log\det\left(\mathbf{I}_{j_i\times j_i} + \frac{n_i}{m+n_{i-1}}\Delta_{ct,i}\Delta_{cst,i}^{-1}\right) \\ &+ \log\det(\mathbf{I}_{c_i\times c_i} + \frac{n_i}{n_{i-1}}\Delta_{t,i}\Delta_{t,i-1}^{-1}) + \log\det(n_i I_{t,i}) \\ &+ (d-j_i-c_i)\log\frac{1}{2\pi e} + \frac{2}{\omega(\theta_{t,i}^*|\theta_{t,i-1}^*,\theta_s^*)}. \end{aligned} \tag{242}$$

Based on the theorem above, by putting things together, finally we reach

$$\begin{aligned} \mathcal{R}_{TV} \le & M\Bigg(l\sum_{i=1}^{l} n_i\Big(\log\det\left(\mathbf{I}_{j_i\times j_i} + \frac{n_i}{m+n_{i-1}}\Delta_{ct,i}\Delta_{cst,i}^{-1}\right) + \log\det(\mathbf{I}_{c_i\times c_i} + \frac{n_l}{n_{l-1}}\Delta_{t,i}\Delta_{t,i-1}^{-1}) + \log\det(n_i I_{t,i}) \\ & + (d-j_i-c_i)\log\frac{1}{2\pi e} + \frac{2}{\omega(\theta_{t,i}^*|\theta_{t,i-1}^*,\theta_s^*)}\Big)\Bigg)^{\frac{1}{2}}. \end{aligned} \tag{243}$$

Since $n_{l-1} \asymp n_l$ and $m \asymp n_l^p$, using the same procedure from Corollary 1 we will arrive at

$$\mathcal{R}_{TV} \lesssim \sqrt{k\sum_{l=1}^{k} n_l\Big(j_i(1\wedge n_l^{1-p}) + c_i + (d-c_i-j_i)\log n_i + \frac{2}{\omega(\theta_{t,i}^*|\theta_{t,i-1}^*,\theta_s^*)}\Big)}. \tag{244}$$

A.14. Proof of Proposition 1

Proof. Since we have

$$\mathcal{R}_I = \mathbb{E}_{\theta_t^*,\theta_s^*}\left[\log\frac{P_{\theta_t^*}(Z_t')}{Q(Z_t'|D_t^n, D_s^m)}\right], \tag{245}$$

where we use the mixture strategy for the conditional distribution Q as

$$\mathbb{E}\left[\log Q(Z_t'|D_t^n, D_s^m)\right] = \mathbb{E}\left[\log\frac{\int P_{\theta_t}(D_t^n, Z_t')P_{\theta_s}(D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s}{\int P_{\theta_t}(D_t^n)P_{\theta_s^*}(D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s}\right] \tag{246}$$

$$= \mathbb{E}\left[\log\frac{\int P_{\theta_t}(Z_t')P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s)d\theta_t P(\theta_s|D_s^m)d\theta_s}{\int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s)d\theta_t P(\theta_s|D_s^m)d\theta_s}\right] \tag{247}$$

$$= \mathbb{E}\left[\log\frac{\int P_{\theta_t}(Z_t')P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s^*)d\theta_t}{\int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s^*)d\theta_t}\right] \tag{248}$$

$$= \mathbb{E}\left[\log\int P_{\theta_t}(Z_t')P_\omega(\theta_t|D_t^n)d\theta_t\right] \tag{249}$$

$$\le \max_{\tilde{\theta}_t\in\mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} \mathbb{E}\left[\log\int P_{\tilde{\theta}_t}(Z_t')P_\omega(\theta_t|D_t^n)d\theta_t\right] \tag{250}$$

$$= \mathbb{E}\left[\log\int P_{\tilde{\theta}_t}(Z_t')P_\omega(\theta_t|D_t^n)d\theta_t\right] \tag{251}$$

$$= \mathbb{E}\left[\log P_{\tilde{\theta}_t}(Z_t')\right]. \tag{252}$$

The inequality holds as $\tilde{\theta}_t = \mathrm{argmin}_{\theta_t\in\mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} D_{\mathrm{KL}}(P_{\theta_t^*}(Z_t)\|P_{\theta_t}(Z_t))$ and we define the conditional posterior as

$$P_\omega(\theta_t|D_t) = \frac{\omega(\theta_t|\theta_s^*)P_{\theta_t}(D_t^n)}{\int \omega(\theta_t|\theta_s^*)P_{\theta_t}(D_t^n)d\theta_t}. \tag{253}$$

The domain of this posterior is the same as the support of $\omega(\theta_s|\theta_t)$. Therefore, minimizing the KL divergence $D_{\mathrm{KL}}(P_{\theta_t^*}(Z_t)\|P_{\theta_t}(Z_t))$ w.r.t. θ_t is equivalent to maximizing the cross entropy as

$$\mathrm{argmin}_{\theta_t\in\mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} D_{\mathrm{KL}}(P_{\theta_t^*}(Z_t)\|P_{\theta_t}(Z_t)) = \mathrm{argmax}_{\theta_t\in\mathrm{supp}(\omega(\Theta_t|\Theta_s^*))}\mathbb{E}_{\theta_t^*}[\log P_{\theta_t}(Z_t)]. \tag{254}$$

Then the result follows.

A.15. Proof of Proposition 2

Proof. We need to prove there exists a prior $\omega(\theta_s, \theta_t)$, the expected regrets such that

$$\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right] > \mathbb{E}_{\theta_t^*}\left[\log\frac{P_{\theta_t^*}(D_t^n)}{\hat{Q}(D_t^n)}\right]. \tag{255}$$

It is equivalent to prove that,

$$\mathbb{E}_{\theta_t^*,\theta_s^*}\left[\log\frac{Q(D_s^m)\hat{Q}(D_t^n)}{Q(D_t^n, D_s^m)}\right] > 0. \tag{256}$$

Let us examine the logarithmic term,

$$\log\frac{Q(D_s^m)\hat{Q}(D_t^n)}{Q(D_t^n, D_s^m)} = \log\frac{\hat{Q}(D_t^n)\int P_{\theta_t,\theta_s}(D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s}{\int P_{\theta_t,\theta_s}(D_t^n, D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s} \tag{257}$$

$$= \log\frac{\hat{Q}(D_t^n)\int P_{\theta_s}(D_s^m)\omega(\theta_s)d\theta_s}{\int P_{\theta_t,\theta_s}(D_t^n, D_s^m)\omega(\theta_t,\theta_s)d\theta_t d\theta_s} \tag{258}$$

$$= \log\frac{1}{\int\int \hat{Q}(\theta_t|D_t^n)\frac{\omega(\theta_t|\theta_s)}{\hat{\omega}(\theta_t)}d\theta_t Q(\theta_s|D_s^m)d\theta_s}. \tag{259}$$

When both m and n are sufficient enough and the marginal prior distribution $\omega(\theta_s)$ and $\hat{\omega}(\theta_t)$ are proper, $\hat{Q}(\theta_t|D_t^n)$ and $Q(\theta_s|D_s^m)$ will be concentrated near θ_t^* and θ_s^*. Then the above equation becomes

$$\log\frac{Q(D_s^m)\hat{Q}(D_t^n)}{Q(D_t^n, D_s^m)} = -\log\int_{\theta_s^*-\delta_s}^{\theta_s^*+\delta_s}\int_{\theta_t^*-\delta_t}^{\theta_t^*+\delta_t}\hat{Q}(\theta_t|D_t^n)\frac{\omega(\theta_t|\theta_s^*)}{\hat{\omega}(\theta_t)}d\theta_t Q(\theta_s|D_s^m)d\theta_s, \tag{260}$$

for some small δ_t and δ_s. Since the prior $\omega(\theta_t|\theta_s)$ is imposed improperly, then $\omega(\theta_t|\theta_s)$ has zero density around θ_t^*, then since for any $\hat{\omega}(\theta_t) > 0$, the following inequality holds.

$$\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\frac{\hat{Q}(D_t^n)Q(D_s^m)}{Q(D_t^n, D_s^m)}\right] = -\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\int_{\theta_s^*-\delta_s}^{\theta_s^*+\delta_s}\int_{\theta_t^*-\delta_t}^{\theta_t^*+\delta_t}\hat{Q}(\theta_t|D_t^n)\frac{\omega(\theta_t|\theta_s^*)}{\hat{\omega}(\theta_t)}d\theta_t Q(\theta_s|D_s^m)d\theta_s\right] \tag{261}$$

$$> -\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log\int_{\theta_s^*-\delta_s}^{\theta_s^*+\delta_s}\int_{\theta_t^*-\delta_t}^{\theta_t^*+\delta_t}\hat{Q}(\theta_t|D_t^n)\frac{\hat{\omega}(\theta_t)}{\hat{\omega}(\theta_t)}d\theta_t Q(\theta_s|D_s^m)d\theta_s\right] \tag{262}$$

$$= 0, \tag{263}$$

when the source and target are sufficiently large. Therefore, it implies that

$$\mathcal{R}_{\omega(\Theta_s,\Theta_t)}(n) > \mathcal{R}_{\hat{\omega}(\Theta_t)}(n). \tag{264}$$

Furthermore, if we rewrite the regret as

$$
\begin{aligned}
\mathcal{R}_O &= \mathbb{E}_{\theta_t^*}\left[\log \frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right] \\
&= H_{P_{\theta_t^*}}(D_t^n) - \mathbb{E}_{\theta_t^*}\left[\log Q(D_t^n|D_s^m)\right] \\
&= H_{P_{\theta_t^*}}(D_t^n) - \mathbb{E}_{\theta_t^*}\left[\int\int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s)d\theta_t Q(\theta_s|D_s^m)d\theta_s\right] \\
&\overset{(a)}{=} H_{P_{\theta_t^*}}(D_t^n) - \mathbb{E}_{\theta_t^*}\left[\int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s^*)d\theta_t\right] \\
&\overset{(b)}{\geq} H_{P_{\theta_t^*}}(D_t^n) - \mathbb{E}_{\theta_t^*}\left[P_{\tilde{\theta}_t}(D_t^n)\right] \\
&= D_{\mathrm{KL}}(P_{\theta_t^*}(D_t^n)\|P_{\tilde{\theta}_t}(D_t^n)) \\
&\overset{(c)}{=} nD_{\mathrm{KL}}(P_{\theta_t^*}(Z_t)\|P_{\tilde{\theta}_t}(Z_t)),
\end{aligned}
$$

where (a) follows that when m goes to sufficiently large and we assume that the prior $\omega(\theta_s)$ is proper, the posterior $Q(\theta_s|D_s^m)$ will approach the true parameter θ_s^*. (b) holds as we define

$$\tilde{\theta}_t = \mathrm{argmin}_{\theta_t \in \mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} D_{\mathrm{KL}}(P_{\theta_t^*}(Z_t)\|P_{\theta_t}(Z_t)), \tag{265}$$

which is equivalent to maximize the cross entropy as

$$\tilde{\theta}_t = \mathrm{argmax}_{\theta_t \in \mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} \mathbb{E}_{\theta_t^*}\left[\log P_{\theta_t}(Z_t)\right]. \tag{266}$$

With the Assumption 1, with the i.i.d. property we have

$$\tilde{\theta}_t = \mathrm{argmax}_{\theta_t \in \mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} \mathbb{E}_{\theta_t^*}\left[\log P_{\theta_t}(D_t^n)\right]. \tag{267}$$

Therefore, (b) holds as

$$\mathbb{E}_{\theta_t^*}\left[\int P_{\theta_t}(D_t^n)\omega(\theta_t|\theta_s^*)d\theta_t\right] \leq \max_{\tilde{\theta}_t \in \mathrm{supp}(\omega(\Theta_t|\Theta_s^*))} \mathbb{E}_{\theta_t^*}\left[\int P_{\tilde{\theta}_t}(D_t^n)\omega(\theta_t|\theta_s^*)d\theta_t\right] = \mathbb{E}_{\theta_t^*}\left[P_{\tilde{\theta}_t}(D_t^n)\right], \tag{268}$$

and finally (c) follows as target data are drawn i.i.d. from $P_{\theta_t^*}$. With the dual properties, the proof of the upper bound follows the same procedures as the lower upper.

A.16. Proof of Proposition 3

Proof. We need to prove under the certain assumptions, there exists a prior $\omega(\theta_s, \theta_t)$, the expected regrets such that

$$\mathbb{E}_{\theta_s^*,\theta_t^*}\left[\log \frac{P_{\theta_t^*}(D_t^n)}{Q(D_t^n|D_s^m)}\right] < \mathbb{E}_{\theta_t^*}\left[\log \frac{P_{\theta_t^*}(D_t^n)}{\hat{Q}(D_t^n)}\right]. \tag{269}$$

Rewrite the expectation and it is equivalent to prove that

$$\mathbb{E}_{\theta_t^*,\theta_s^*}\left[\log \frac{Q(D_s^m)\hat{Q}(D_t^n)}{Q(D_t^n, D_s^m)}\right] < 0. \tag{270}$$

Similarly, when D_s^m is sufficient enough, the density $P(\theta_s^*|D_s^m)$ will concentrate around θ_s^*, say $P(|\theta_s - \theta_s^*| < \delta_s) \to 0$ as m goes to infinity, furthermore if D_t^n is also very large, $p(\theta_t|D_t^n)$ will be concentrated near θ_t^* such that

$$\log \frac{Q(D_s^m)\hat{Q}(D_t^n)}{Q(D_t^n, D_s^m)} = -\log \int_{\theta_s^*-\delta_s}^{\theta_s^*+\delta_s}\int_{\theta_t^*-\delta_t}^{\theta_t^*+\delta_t} \hat{Q}(\theta_t|D_t^n)\frac{\omega(\theta_t|\theta_s^*)}{\hat{\omega}(\theta_t)}d\theta_t Q(\theta_s|D_s^m)d\theta_s. \tag{271}$$

Since we assume the support $\omega(\theta_t|\theta_s)$ is a proper subset of Θ and $\omega(\theta_t|\theta_s)$ is proper over θ_t, that is, this conditional prior has positive density around θ_t^*. Let us define

$$\hat{\Omega} = \int_{\theta_t^*-\delta_t}^{\theta_t^*+\delta_t} \hat{\omega}(\theta_t)d\theta_t, \quad \text{and } \Omega = \int_{\theta_t^*-\delta_t}^{\theta_t^*+\delta_t} \omega(\theta_t|\theta_s)d\theta_t. \tag{272}$$

Then there always exists a prior such that for any θ_s around θ_s^*,

$$\Omega - \hat{\Omega} = \Delta > 0, \tag{273}$$

with the choice of the prior

$$\omega(\theta_t|\theta_s) = \hat{\omega}(\theta_t) + \frac{\Delta}{2\delta_t}. \tag{274}$$

This specific prior will lead to $\log \frac{\hat{\omega}(\theta_t)}{\omega(\theta_t|\theta_s)} < 1$ and the quantity above will be strictly less than zero, which is, positive transfer.

A.17. Proof of Lemma 1

Proof. First we rewrite the conditional probability by,

$$Q(D_t^n|D_s^m) = \int_{\theta_s} \int_{\theta_t|\theta_s} P(D_t^n|\theta_t) P(\theta_t|\theta_s) d\theta_t P(\theta_s|D_s^m) d\theta_s \tag{275}$$

$$= \mathbb{E}_{\theta_s|D_s^m} \left[\int_{\theta_s - c}^{\theta_s + c} \frac{1}{2c} P(D_t^n|\theta_t) d\theta_t \right] \tag{276}$$

$$= \frac{1}{2c} \mathbb{E}_{\theta_s|D_s^m} \left[\int_{\theta_s - c}^{\theta_s + c} (\theta_t)^{k_t} (1-\theta_t)^{n-k_t} d\theta_t \right]. \tag{277}$$

Here we denote the integral by,

$$I(n,k) = \binom{n}{k} \int_0^a x^k (1-x)^{n-k} dx \tag{278}$$

Then we rewrite $I(n,k)$ as follows:

$$I(n,k) = \binom{n}{k} \left(\left[\frac{-x^k (1-x)^{n-k+1}}{n-k+1} \right]_0^a + \frac{k}{n-k+1} \int_0^a x^{k-1} (1-x)^{n-k+1} dx \right) \tag{279}$$

$$= \binom{n}{k} \frac{-a^k (1-a)^{n-k+1}}{n-k+1} + \binom{n}{k} \frac{k}{n-k+1} \binom{n}{k-1}^{-1} I(n,k-1) \tag{280}$$

$$= \binom{n}{k} \frac{-a^k (1-a)^{n-k+1}}{n-k+1} + I(n,k-1). \tag{281}$$

By induction, we have

$$I(n,k) = \sum_{i=1}^{k} \binom{n}{i} \frac{-a^i (1-a)^{n-i+1}}{n-i+1} + I(a,0) \tag{282}$$

$$= \sum_{i=1}^{k} \binom{n}{i} \frac{-a^i (1-a)^{n-i+1}}{n-i+1} + \frac{1-(1-a)^{n+1}}{n+1}. \tag{283}$$

Hence,

$$\int_0^a x^k (1-x)^{n-k} dx = \binom{n}{k}^{-1} \left(\sum_{i=1}^{k} \binom{n}{i} \frac{-a^i (1-a)^{n-i+1}}{n-i+1} + \frac{1-(1-a)^{n+1}}{n+1} \right) \tag{284}$$

and for any $b > a$,

$$\int_0^b x^k (1-x)^{n-k} dx = \binom{n}{k}^{-1} \left(\sum_{i=1}^{k} \binom{n}{i} \frac{-b^i (1-b)^{n-i+1}}{n-i+1} + \frac{1-(1-b)^{n+1}}{n+1} \right). \tag{285}$$

By subtraction,

$$\frac{1}{b-a}\int_a^b x^k(1-x)^{n-k}dx = \frac{1}{b-a}\binom{n}{k}^{-1}\left(\sum_{i=1}^{k}\binom{n}{i}\frac{a^i(1-a)^{n-i+1}-b^i(1-b)^{n-i+1}}{n-i+1}+\frac{(1-a)^{n+1}-(1-b)^{n+1}}{n+1}\right). \tag{286}$$

A.18. Experimental setups of EMPU algorithms

In this section, we listed all experimental setups for different algorithms under different tasks. We use the same linear model and learning rate in the OTL algorithm as the linear EMPU algorithm. The experimental setup of linear EMPU algorithm is summarized in Table 8. The details for EMPU with neural network algorithms are summarized in Table 9. The experimental setups of Gibbs EMPU is summarized in Table 10. Table 11 concludes the setups for the RNN with LSTM architecture.

Table 8
Experimental Setups of EMPU algorithms.

Tasks	N	η	c
Office-Caltech-10		0.01	0.1
MNIST vs USPS		0.01	0.1
20 Newsgroup	20	0.1	1
GIST		0.001	0.1
SIFT-SPM		0.001	0.01

Table 9
Experimental Setups of EMPU with neural network algorithms.

Tasks	Network Layout	Optimizer	N	η	c
Office-Caltech-10	[800,16,10]			0.01	0.01
MNIST vs USPS	[256,16,10]			0.01	0.1
20 Newsgroup	[100,4,2]	Adam	20	0.01	0.01
GIST	[512,16,2]			0.001	0.01
SIFT-SPM	[4200,16,2]			0.001	0.01

Table 10
Experimental Setups of Gibbs EMPU algorithms.

Tasks	N	$1/\gamma$	η	σ_p
Office-Caltech-10		1e-5	0.01	0.01
MNIST vs USPS		1e-5	0.01	0.1
20 Newsgroup	20	1e-5	0.1	0.01
GIST		1e-4	0.001	0.01
SIFT-SPM		1e-6	0.0001	0.01

Table 11
Experimental Setups of RNN algorithms.

Tasks	Hidden Layer Number	Hidden Layer Dimension	Optimizer	η
Office-Caltech-10		100		0.001
MNIST vs USPS		50		0.01
20 Newsgroup	1	20	Adam	0.001
GIST		100		0.001
SIFT-SPM		100		0.001

References

[1] G. Alirezaei, R. Mathar, On exponentially concave functions and their impact in information theory, in: 2018 Information Theory and Applications Workshop, ITA, IEEE, 2018, pp. 1–10.
[2] C.E. Antoniak, Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems, Ann. Stat. (1974) 1152–1174.
[3] S. Badreddine, M. Spranger, Injecting prior knowledge for transfer learning into reinforcement learning algorithms using logic tensor networks, arXiv preprint, arXiv:1906.06576, 2019.
[4] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (2010) 151–175.
[5] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer Science & Business Media, 2013.
[6] H.S. Bhatt, A. Rajkumar, S. Roy, Multi-source iterative adaptation for cross-domain classification, in: IJCAI, 2016, pp. 3691–3697.

[7] X. Cao, D. Wipf, F. Wen, G. Duan, J. Sun, A practical transfer learning algorithm for face verification, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3208–3215.
[8] B.E. Chérief-Abdellatif, P. Alquier, Finite sample properties of parametric MMD estimation: robustness to misspecification and dependence, Bernoulli 28 (2022) 181–213.
[9] B.S. Clarke, Asymptotic normality of the posterior in relative entropy, IEEE Trans. Inf. Theory 45 (1999) 165–176.
[10] B.S. Clarke, A.R. Barron, Information-theoretic asymptotics of Bayes methods, IEEE Trans. Inf. Theory 36 (1990) 453–471.
[11] C. Cortes, M. Mohri, M. Riley, A. Rostamizadeh, Sample selection bias correction theory, in: International Conference on Algorithmic Learning Theory, Springer, 2008, pp. 38–53.
[12] T.M. Cover, E. Ordentlich, Universal portfolios with side information, IEEE Trans. Inf. Theory 42 (1996) 348–363.
[13] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, arXiv preprint, arXiv: 1810.04805, 2018.
[14] L. Duan, I.W. Tsang, D. Xu, T.S. Chua, Domain adaptation from multiple sources via auxiliary classifiers, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 289–296.
[15] M. Feder, N. Merhav, M. Gutman, Universal prediction of individual sequences, IEEE Trans. Inf. Theory 38 (1992) 1258–1270.
[16] T.S. Ferguson, A Bayesian analysis of some nonparametric problems, Ann. Stat. (1973) 209–230.
[17] S. Ghosal, A. Van der Vaart, Fundamentals of Nonparametric Bayesian Inference, vol. 44, Cambridge University Press, 2017.
[18] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2066–2073.
[19] A. Graves, Supervised sequence labelling, in: Supervised Sequence Labelling with Recurrent Neural Networks, Springer, 2012, pp. 5–13.
[20] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, B. Schölkopf, Covariate Shift by Kernel Mean Matching, Dataset Shift in Machine Learning, vol. 3, 2009, p. 5.
[21] D. Haussler, J. Kivinen, M.K. Warmuth, Sequential prediction of individual sequences under general loss functions, IEEE Trans. Inf. Theory 44 (1998) 1906–1925.
[22] D. Haussler, M. Opper, General bounds on the mutual information between a parameter and n conditionally independent observations, in: Proceedings of the Eighth Annual Conference on Computational Learning Theory, 1995, pp. 402–411.
[23] J. He, R. Lawrence, A graphbased framework for multi-task multi-view learning, in: ICML, 2011, pp. 25–32.
[24] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[25] J. Hoffman, T. Darrell, K. Saenko, Continuous manifold based adaptation for evolving visual domains, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 867–874.
[26] J.H. Huggins, T. Campbell, M. Kasprzak, T. Broderick, Practical bounds on the error of Bayesian posterior approximations: a nonasymptotic approach, arXiv preprint, arXiv:1809.09505, 2018.
[27] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 550–554, https://doi.org/10.1109/34.291440.
[28] Z. Kang, B. Yang, S. Yang, X. Fang, C. Zhao, Online transfer learning with multiple source domains for multi-class classification, Knowl.-Based Syst. 190 (2020) 105149.
[29] A. Kumagai, T. Iwata, Learning future classifiers without additional data, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016.
[30] I. Kuzborskij, F. Orabona, Stability and hypothesis transfer learning, in: International Conference on Machine Learning, PMLR, 2013, pp. 942–950.
[31] A. Lazaric, Transfer in reinforcement learning: a framework and a survey, in: Reinforcement Learning, Springer, 2012, pp. 143–173.
[32] A. Lazaric, M. Restelli, A. Bonarini, Transfer of samples in batch reinforcement learning, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 544–551.
[33] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'06, IEEE, 2006, pp. 2169–2178.
[34] Y. LeCun, C. Cortes, MNIST handwritten digit database, URL: http://yann.lecun.com/exdb/mnist/, 2010.
[35] H. Liu, M. Long, J. Wang, Y. Wang, Learning to adapt to evolving domains, Adv. Neural Inf. Process. Syst. 33 (2020).
[36] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: International Conference on Machine Learning, PMLR, 2015, pp. 97–105.
[37] M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, Transfer feature learning with joint distribution adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2200–2207.
[38] M. Long, H. Zhu, J. Wang, M.I. Jordan, Deep transfer learning with joint adaptation networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 2208–2217.
[39] M. Mancini, S.R. Bulo, B. Caputo, E. Ricci, Adagraph: unifying predictive and continuous domain adaptation through graphs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6568–6577.
[40] N. Merhav, M. Feder, Universal prediction, IEEE Trans. Inf. Theory 44 (1998) 2124–2147.
[41] B. Mieth, J.R. Hockley, N. Görnitz, M.M.C. Vidovic, K.R. Müller, A. Gutteridge, D. Ziemek, Using transfer learning from prior reference knowledge to improve the clustering of single-cell rna-seq data, Sci. Rep. 9 (2019) 1–14.
[42] R.M. Neal, Markov chain sampling methods for Dirichlet process mixture models, J. Comput. Graph. Stat. 9 (2000) 249–265.
[43] J. Paisley, A Simple Proof of the Stick-Breaking Construction of the Dirichlet Process. Department of Computer Science, Princeton University, 2010.
[44] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2009) 1345–1359.
[45] W. Pan, E. Xiang, N. Liu, Q. Yang, Transfer learning in collaborative filtering for sparsity reduction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2010.
[46] P.D. Powell, Calculating determinants of block matrices, arXiv preprint, arXiv:1112.4379, 2011.
[47] I. Redko, A. Habrard, M. Sebban, Theoretical analysis of domain adaptation with optimal transport, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2017, pp. 737–753.
[48] I. Redko, E. Morvant, A. Habrard, M. Sebban, Y. Bennani, A survey on domain adaptation theory: learning bounds and theoretical guarantees. arXiv e-prints, arXiv:2004.11829v6 [cs.LG], 2020.
[49] M.T. Rosenstein, Z. Marx, L.P. Kaelbling, T.G. Dietterich, To transfer or not to transfer, in: NIPS 2005 Workshop on Transfer Learning, 2005, pp. 1–4.
[50] S. Ruder, M.E. Peters, S. Swayamdipta, T. Wolf, Transfer learning in natural language processing, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, 2019, pp. 15–18.
[51] D. Russo, B. Van Roy, An information-theoretic analysis of Thompson sampling, J. Mach. Learn. Res. 17 (2016) 2442–2471.
[52] B. Shahbaba, R. Neal, Nonlinear models using Dirichlet process mixtures, J. Mach. Learn. Res. 10 (2009).
[53] Y. Shkel, M. Raginsky, S. Verdú, Sequential prediction with coded side information under logarithmic loss, in: Algorithmic Learning Theory, 2018, pp. 753–769.
[54] C. Siagian, L. Itti, Rapid biologically-inspired scene classification using features shared with visual attention, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 300–312.

[55] M.E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: a survey, J. Mach. Learn. Res. 10 (2009).
[56] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7167–7176.
[57] B. Wang, J. Mendez, M. Cai, E. Eaton, Transfer learning via minimizing the performance gap between domains, Adv. Neural Inf. Process. Syst. 32 (2019).
[58] H. Wang, H. He, D. Katabi, Continuously indexed domain adaptation, arXiv preprint, arXiv:2007.01807, 2020.
[59] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, Q. Yang, Transfer learning with dynamic distribution adaptation, ACM Trans. Intell. Syst. Technol. 11 (2020) 1–25.
[60] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, P.S. Yu, Visual domain adaptation with manifold embedded distribution alignment, in: Proceedings of the 26th ACM International Conference on Multimedia, 2018, pp. 402–410.
[61] M. Wang, W. Deng, Deep visual domain adaptation: a survey, Neurocomputing 312 (2018) 135–153.
[62] Z. Wang, Z. Dai, B. Póczos, J. Carbonell, Characterizing and avoiding negative transfer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11293–11302.
[63] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (2016) 9.
[64] J. Wu, J. He, Continuous transfer learning with label-informed distribution alignment, arXiv preprint, arXiv:2006.03230, 2020.
[65] Q. Wu, X. Zhou, Y. Yan, H. Wu, H. Min, Online transfer learning by leveraging multiple source domains, Knowl. Inf. Syst. 52 (2017) 687–707.
[66] X. Wu, J.H. Manton, U. Aickelin, J. Zhu, Information-theoretic analysis for transfer learning, in: 2020 IEEE International Symposium on Information Theory (ISIT), IEEE, 2020, pp. 2819–2824.
[67] X. Wu, J.H. Manton, U. Aickelin, J. Zhu, Online transfer learning: negative transfer and effect of prior knowledge, arXiv preprint, arXiv:2105.01445, 2021.
[68] Q. Xie, A.R. Barron, Asymptotic minimax regret for data compression, gambling, and prediction, IEEE Trans. Inf. Theory 46 (2000) 431–445.
[69] A. Xu, M. Raginsky, Information-theoretic analysis of generalization capability of learning algorithms, Adv. Neural Inf. Process. Syst. 30 (2017).
[70] A. Xu, M. Raginsky, Minimum excess risk in Bayesian learning, arXiv preprint, arXiv:2012.14868, 2020.
[71] Y. Yan, Q. Wu, M. Tan, M.K. Ng, H. Min, I.W. Tsang, Online heterogeneous transfer by hedge ensemble of offline and online decisions, IEEE Trans. Neural Netw. Learn. Syst. 29 (2017) 3252–3263.
[72] Q. Yang, Y. Zhang, W. Dai, S.J. Pan, Transfer Learning, Cambridge University Press, 2020.
[73] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q.V. Le, XLNet: generalized autoregressive pretraining for language understanding, arXiv preprint, arXiv:1906.08237, 2019.
[74] C. Yu, J. Wang, Y. Chen, M. Huang, Transfer learning with dynamic adversarial adaptation network, in: 2019 IEEE International Conference on Data Mining, ICDM, IEEE, 2019, pp. 778–786.
[75] Y. Zhan, M.E. Taylor, Online transfer learning in reinforcement learning domains, arXiv preprint, arXiv:1507.00436, 2015.
[76] C. Zhang, L. Zhang, J. Ye, Generalization bounds for domain adaptation, Adv. Neural Inf. Process. Syst. 4 (2012) 3320.
[77] J. Zhang, W. Li, P. Ogunbona, Joint geometrical and statistical alignment for visual domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1859–1867.
[78] T. Zhang, Information-theoretic upper and lower bounds for statistical estimation, IEEE Trans. Inf. Theory 52 (2006) 1307–1321.
[79] W. Zhang, L. Deng, D. Wu, Overcoming negative transfer: a survey, arXiv preprint, arXiv:2009.00909, 2020.
[80] Y. Zhang, T. Liu, M. Long, M. Jordan, Bridging theory and algorithm for domain adaptation, in: International Conference on Machine Learning, PMLR, 2019, pp. 7404–7413.
[81] P. Zhao, S.C. Hoi, J. Wang, B. Li, Online transfer learning, Artif. Intell. 216 (2014) 76–102.
[82] J. Zhu, Semi-supervised learning: the case when unlabeled data is equally useful, in: Conference on Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 709–718.
[83] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2020) 43–76.