Overcoming Saturation in Density Ratio Estimation by Iterated Regularization

Lukas Gruber [1] **Markus Holzleitner** [2] **Johannes Lehner** [1] **Sepp Hochreiter** [1 3] **Werner Zellinger** [4]

Abstract

Estimating the ratio of two probability densities from finitely many samples, is a central task in machine learning and statistics. In this work, we show that a large class of kernel methods for density ratio estimation suffers from error saturation, which prevents algorithms from achieving fast error convergence rates on highly regular learning problems. To resolve saturation, we introduce iterated regularization in density ratio estimation to achieve fast error rates. Our methods outperform its non-iteratively regularized versions on benchmarks for density ratio estimation as well as on large-scale evaluations for importance-weighted ensembling of deep unsupervised domain adaptation models.

1. Introduction

Given two i.i.d. samples $\mathbf{x} = \{x_i\}_{i=1}^m$ and $\mathbf{x}' = \{x_i'\}_{i=1}^n$, drawn from two probability measures P and Q, respectively, the problem of density ratio estimation is to approximate $\beta := \frac{\mathrm{d}P}{\mathrm{d}Q}$. This problem arises in anomaly detection (Smola et al., 2009; Hido et al., 2011), two-sample testing (Keziou & Leoni-Aubin, 2005; Kanamori et al., 2011), divergence estimation (Nguyen et al., 2007; 2010b), unsupervised domain adaptation (Shimodaira, 2000; Dinu et al., 2023), generative modeling (Mohamed & Lakshminarayanan, 2016), conditional density estimation (Schuster et al., 2020), and PU learning (Kato et al., 2019).

A large class of methods for density ratio estimation relies on the estimation of minimizers of the form Sugiyama et al. (2012b;a); Zellinger et al. (2023)

$$f^\lambda := \operatorname*{arg\,min}_{f \in \mathcal{H}} B_F(\beta, g(f)) + \frac{\lambda}{2} \|f\|^2 , \qquad (1)$$

where $g(f^\lambda)$ is a model for the unknown density ratio β with $f \in \mathcal{H}$ in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, $B_F(\beta, \widetilde{\beta}) := F(\beta) - F(\widetilde{\beta}) - \nabla F(\widetilde{\beta})[\beta - \widetilde{\beta}]$ is a Bregman divergence generated by some convex functional $F : L^1(Q) \to \mathbb{R}$, $\|.\|$ is a norm on $\mathcal{H}$, and $\lambda > 0$ is a regularization parameter. From the view point of regularization theory, Eq. (1) can be seen as *Bregmanized* density-ratio estimation. The idea of Bregmanization has been already successfully employed in the context of total variation regularization (Osher et al., 2005; Resmerita & Scherzer, 2006).

In the following, we exemplify four methods in this class. We refer to Sugiyama et al. (2012a, Section 7) and Menon & Ong (2016); Kato & Teshima (2021) for further examples.

Example 1.

- The kernel unconstrained least squares importance fitting procedure (KuLSIF) (Kanamori et al., 2009) can be obtained from Eq. (1) by $F(h) = \int_{\mathcal{X}} (h(x) - 1)^2/2 \, \mathrm{d}Q(x)$ and $g(f) = f$.
- The logistic regression approach (LR) employed in Bickel et al. (2009, Section 7) can be obtained using $F(h) = \int_{\mathcal{X}} h(x) \log(h(x)) - (1+h(x)) \log(1+h(x)) \, \mathrm{d}Q(x)$ and $g(f) = e^f$.
- The exponential approach (Exp) in Menon & Ong (2016) is realized by $F(h) = \int_{\mathcal{X}} h(x)^{-3/2} \, \mathrm{d}Q(x)$, $g(f) = e^{2f}$.
- The square loss approach (SQ) in Menon & Ong (2016) originates from $F(h) = \int_X 1/(2h(x) + 2) \, \mathrm{d}Q(x)$, $g(f) = \frac{-1+2f}{2-2f}$.

However, state-of-the-art error guarantees as Kanamori et al. (2012b); Que & Belkin (2013); Gizewski et al. (2022); Zellinger et al. (2023) for the methods above are sub-optimal for problems of high regularity, i.e., they don't achieve optimal convergence rates when increasing the number of samples, see Theorem 1 and Figure 1. This issue is called *saturation* in inverse problems (Engl et al., 1996b) and learning theory (Bauer et al., 2007; Li et al., 2022), and its negative effect in supervised learning has been recently highlighted (Beugnot et al., 2021).

[1]ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning, Johannes Kepler University Linz, Austria [2]MaLGa Center, Department of Mathematics, University of Genoa [3]NXAI GmbH, Linz, Austria [4]Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences. Correspondence to: Werner Zellinger <werner.zellinger@ricam.oeaw.ac.at>, Lukas Gruber <gruber@ml.jku.at>.

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

Inspired by the Bregman method (Bregman, 1967) and Bregmanization (Osher et al., 2005; Resmerita & Scherzer, 2006), we approach saturation of Eq. (1) by iterated regularization (King & Chillingworth, 1979) of the form

$$f^{\lambda,t+1} := \arg\min_{f\in\mathcal{H}} B_F(\beta, g(f)) + \frac{\lambda}{2}\left\|f - f^{\lambda,t}\right\|^2 \quad (2)$$

with $f^{\lambda,0} = 0$. We prove that iterative density ratio estimation methods, according to Eq. (2), do not suffer from saturation. For $t = 1$, Eq. (2) includes, e.g., approaches in the example above. As a consequence, we are able to improve the empirical performance of these approaches on benchmark experiments, using a higher iteration number $t > 1$ ($t \leq 10$ in our experiments).

Density ratio estimation is an integral part of many parameter choice methods in domain adaptation, see e.g. Sugiyama et al. (2007); You et al. (2019); Dinu et al. (2023). Based on our iteratively regularized approaches, we are able to increase the performance of importance-weighted ensembling in domain adaptation on several large-scale domain adaptation tasks.

Our main contributions are summarized as follows:

- We introduce iterated Tikhonov regularization for a large class of methods for density ratio estimation.
- We provide the (to the best of our knowledge first) non-saturating error bounds for density ratio estimation, which are optimal for the square loss approach (Menon & Ong, 2016) in the sense of Caponnetto & De Vito (2007).
- Our new iteratively regularized methods outperform its non-iteratively regularized versions on average on all benchmark datasets.
- Consistent performance improvements are also observed in the application of importance-weighted ensembling of deep unsupervised domain adaptation models.

2. Related Work

It was first observed in Sugiyama et al. (2012b) that many loss functions for density ratio estimation can be modeled by a Bregman divergence $B_F(\beta, g(f))$ as used in Eq. (1), see also Menon & Ong (2016); Kato & Teshima (2021) and references therein.

The study of saturation issues in regularization dates back to inverse problems literature (Neubauer, 1997; Engl et al., 1996b) and has been further investigated in learning theory (Bauer et al., 2007; Caponnetto & De Vito, 2007; Li et al., 2022; Lin et al., 2020); which includes recent studies for general (self-concordant) convex loss functions (Beugnot et al., 2021) and vector-valued output spaces (Meunier et al., 2024). Our work can be regarded as an extension of many of these investigations from supervised learning to density ratio estimation.

There are some theoretical results on the large class of algorithms following variations of Eq. (1), see Kanamori et al. (2012b); Kato & Teshima (2021); Gizewski et al. (2022); Zellinger et al. (2023). However, to the best of our knowledge, none of these works achieves the optimal error rate discovered in Caponnetto & De Vito (2007) for supervised learning (for $r > \frac{1}{2}$ in Assumption 1). We are also not aware of any approach provably overcoming the saturation problem of Tikhonov regularization $\|f\|^2$ in Eq. (1), except Nguyen et al. (2023) who study iterated Lavrentiev regularization only for the special case of KuLSIF. They include only a small numerical visualization and their pointwise error measure is different from our B_F. Our approach overcomes saturation and provably achieves the optimal rate of square loss.

As noted above, Eq. (2) is inspired by the Bregman method (Bregman, 1967) and Bregmanization (Osher et al., 2005; Resmerita & Scherzer, 2006) as studied in Inverse Problems. It is demonstrated in Resmerita & Scherzer (2006), that regularization by Bregman distance can be considered as an instance of *enhancing techniques*, commonly used in image processing. We refer to the book (Scherzer et al., 2009) for in-depth convergence studies.

Eq. (4) identifies the proposed methods as iterated Tikhonov regularizations, which have been studied in Inverse Problems literature, e.g., for linear problems (Hanke & Groetsch, 1998), for non-linear problems (Scherzer, 1993), in the continuous case (Scherzer & Groetsch, 2001), for non-quadratic penalties (Tadmor et al., 2004; Kindermann et al., 2023) and for non-linear operators combined with non-linear penalty (Modin et al., 2019). All these studies can serve as a basis for analysing Eq. (2).

Our analysis of the sampling behaviour extends fast error rates from kernel regression (Beugnot et al., 2021) (which relies on Engl et al. (1996a); Bauer et al. (2007)) to density ratio estimation, using the duality between divergence estimation (Reid & Williamson, 2011) and binary classification (Menon & Ong, 2016), and the self-concordance property of associated loss functions (Bach, 2010; Marteau-Ferey et al., 2019b).

It is important to note that iterated Tikhonov regularization is intrinsically related to the *proximal point method* (Rockafellar, 1976).

The focus of our work is to improve density ratio estimation for methods with theoretical guarantees such as kernel methods and infinitely wide neural networks, see Jacot et al. (2018). For general deep learning, our theory does not apply, as additional regularizations appear (Gidel et al., 2019). However, some recent works identify scientific deep den-

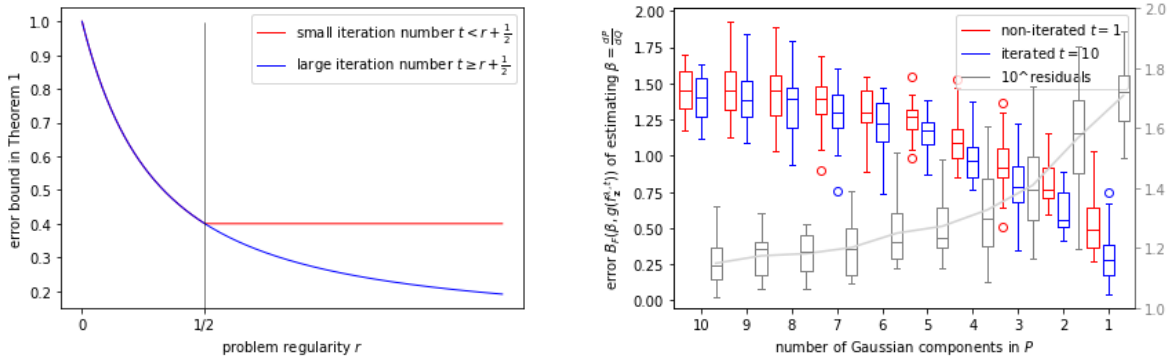

Figure 1. Saturation issue of classical methods Eq. (1) versus our novel iteratively regularized approach Eq. (2). Left: Error rates proven in Theorem 1 for classical methods (red) and ours (blue). Right: Error for classical KuLSIF (Kanamori et al., 2009) method with $F(h) = \int (h(x) - 1)^2 / 2q(x)\, \mathrm{d}x$ in Eq. (1) (red), and our iteratively regularized approach (blue) applied to Gaussian mixture P and Gaussian distribution Q; smaller number of components allows higher regularity index r. The residual differences (grey) between the methods increase with higher regularity.

sity ratio estimation problems. For example Rhodes et al. (2020) and Kato & Teshima (2021); Kiryo et al. (2017) (PU-learning) and Srivastava et al. (2023) discuss problems (e.g., density-chasm) appearing for very different densities; note that saturation also happens for very similar densities (see Figure 2). Notably, Choi et al. (2021) propose an approach by training normalizing flows to obtain closer and simpler densities. Another important recent line of research is density ratio estimation for correcting deep generative diffusion models (Kim et al., 2023). Notably, in Kim et al. (2024), the Bregman divergence approach in Eq. (1) is extended to time-dependent correction and they identify further interesting Bregman divergences.

In our empirical studies, we rely on the benchmark experiments for density ratio estimation of Kanamori et al. (2012b) and the large scale-domain adaptation framework of Ragab et al. (2023) with importance weighting based parameter choice as introduced in Dinu et al. (2023).

3. Notation

Class Probability Estimation Let us denote by $\mathcal{X}$ a (compact) input space and by $\mathcal{Y} := \{-1, 1\}$ the binary label space and assume that an input x is drawn with probability[1] $\frac{1}{2}$ from P and Q. That is, there is a probability measure ρ on $\mathcal{X} \times \mathcal{Y}$, with conditional measures $\rho(x|y = 1) := P(x), \rho(x|y = -1) := Q(x)$ and marginal measure $\rho_{\mathcal{Y}}$ defined as Bernoulli measure assigning probability $\frac{1}{2}$ to both events $y = 1, y = -1$, see Reid & Williamson (2010). That is, the sample $\mathbf{z} := (x_i, 1)_{i=1}^m \cup (x'_i, -1)_{i=1}^n$ can be regarded as an i.i.d. sample of an $\mathcal{X} \times \mathcal{Y}$-valued random variable Z with measure ρ.

[1] See the supplementary material of (Menon & Ong, 2016) for probabilities $\pi \neq \frac{1}{2}$.

For a *loss* function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$, we measure by $\ell(1, f(x))$ and $\ell(-1, f(x))$, respectively, the error of a classifier $f(x)$ which predicts whether x is drawn from P or Q. A loss function ℓ is called *strictly proper composite* (Buja et al., 2005) if there exists an invertible link function $\Psi : [0, 1] \to \mathbb{R}$ so that the Bayes optimal model $f^* := \arg\min_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$ with expected risk $\mathcal{R}(f) := \mathbb{E}_{(x,y)\sim\rho}[\ell(x, f(x))]$ is achieved by $f^* = \Psi \circ \rho(y = 1|x)$. We denote by $G(u) := u\ell(1, \Psi(u)) + (1 - u)\ell(-1, \Psi(u))$ the conditional *Bayes risk* of ℓ. We will also denote by $\ell_z : \mathcal{H} \to \mathbb{R}$ the function defined by $\ell_z(f) := \ell(y, f(x))$ for $z = (x, y)$.

Throughout this work, we neglect pathological loss functions and only consider strictly proper composite losses ℓ with twice differentiable Bayes risk G.

Learning in RKHS In this work, we focus on models $f \in \mathcal{H}$ in a reproducing kernel Hilbert space $\mathcal{H}$ with continuous bounded kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Furthermore, for simplicity, for any considered loss function ℓ, we will assume that the expected risk minimizer $f_{\mathcal{H}} := \arg\min_{f\in\mathcal{H}} \mathcal{R}(f)$ exists and is in $\mathcal{H}$.

4. Iterated Regularization

We rely on the following lemma to approximate the solution of Eq. (2) from the data $\mathbf{z}$.

Lemma 1 (Menon & Ong (2016)). *Any strictly proper composite loss ℓ with invertible link $\Psi : [0, 1] \to \mathbb{R}$ and twice differentiable Bayes risk $G : [0, 1] \to \mathbb{R}$ satisfies*

$$\frac{1}{2} B_F(\beta, g(f)) = \mathcal{R}(f) - \mathcal{R}(f^*) \tag{3}$$

with $F(h) := -\int_{\mathcal{X}} (1 + h(x)) G\left(\frac{h(x)}{1+h(x)}\right) \mathrm{d}Q(x)$ *and*

$g(f) := \frac{\Psi^{-1}\circ f}{1-\Psi^{-1}\circ f}$.

Notably, the construction in Lemma 1 includes all approaches in Example 1.

Example 2.

- For KuLSIF, Eq. (3) holds with $\ell(-1,v) = \frac{1}{2}v^2$, $\ell(1,v) = -v$ and $\Psi(v) = \frac{v}{1-v}$.
- LR realizes Eq. (3) with $\ell(-1,v) = \log(1+e^v)$, $\ell(1,v) = \log(1+e^{-v})$ and the link function $\Psi(v) = -\log(\frac{1}{v}-1)$.
- Exp can be obtained using $\ell(-1,v) = e^v, \ell(1,v) = e^{-v}$ and $\Psi(v) = -\frac{1}{2}\log(\frac{1}{v}-1)$.
- SQ uses $\ell(-1,v) = (1+v)^2, \ell(1,v) = (1-v)^2$ and $\Psi(v) = \frac{v+1}{2}$.

Eq. (3) allows us to estimate the density ratio β by $g(f_{\mathbf{z}}^{\lambda,t})$ with $f_{\mathbf{z}}^{\lambda,0} = 0$ and the empirical risk minimizer

$$f_{\mathbf{z}}^{\lambda,t+1} := \operatorname*{arg\,min}_{f\in\mathcal{H}} \frac{1}{|\mathbf{z}|}\sum_{i=1}^{m+n} \ell(y_i, f(x_i)) + \frac{\lambda}{2}\left\|f - f_{\mathbf{z}}^{\lambda,t}\right\|^2. \tag{4}$$

In the following, we will discuss, how to compute $f_{\mathbf{z}}^{\lambda,t+1}$.

4.1. Algorithmic Realization

To solve Eq. (4), we follow the representer theorem (Wahba, 1990; Schölkopf et al., 2001) , make the model ansatz $f(\cdot) = \sum_{i=1}^{m+n}\alpha_i k(x_i,\cdot)$ and apply the conjugate gradient (CG) method for minimization, except for KuLSIF in Example 2, for which an explicit solution is available, see Appendix C. Originally, the CG method was developed to solve linear equations with a positive semi-definite matrix. There, the goal is to compute a sequence of conjugate search directions, where two vectors are considered conjugate if their dot product with respect to the given matrix is zero. These search directions arise as gradients of an associated quadratic optimization problem, see e.g. Golub & Van Loan (2013, Section 11.3).

The CG-method can be generalized to deal with nonlinear problems like Eq. (4), which leads to a minimization problem of a strongly convex smooth function in each iteration step. Therefore, several CG approaches can be applied, see, e.g., Hager & Zhang (2006). The method which is implemented in Python Scipy (Virtanen et al., 2020) and which we are using, is based on an additional Polak-Ribiere line search scheme and it is guaranteed to converge globally in our case (Hager & Zhang, 2006; Cohen, 1972). In the upcoming section we will also briefly discuss how this algorithm choice affects our theoretical guarantees.

5. Learning Theoretic Analysis

In learning theory, the regularity of learning problems is traditionally encoded by *source conditions*, see, e.g., Bauer et al. (2007); Caponnetto & De Vito (2007); Marteau-Ferey et al. (2019b).

Assumption 1 (source condition)**.** There exist $r > 0, v \in \mathcal{H}$ such that $f_{\mathcal{H}} = \mathbf{H}(f_{\mathcal{H}})^r v$ with the expected Hessian $\mathbf{H}(f) := \mathbb{E}[\nabla^2 \ell_Z(f)]$.

To interpret Assumption 1, let us assume $\ell_z(f) := (y - f(x))^2$ is the square loss. Then Assumption 1 recovers the polynomial source condition used in Caponnetto & De Vito (2007). Starting from $r = 0$, the regularity of $f_{\mathcal{H}}$ increases with increasing r, essentially reducing the number of eigenvectors of the covariance operator T needed to well approximate $f_{\mathcal{H}}$. However, it is known that optimal learning rates are only possible under a further condition on the *capacity* of the space $\mathcal{H}$ (Marteau-Ferey et al., 2019b).

Assumption 2 (capacity condition)**.** There exist $\alpha \geq 1$ and $S > 0$ such that $\mathrm{df}_\lambda \leq S\lambda^{-\frac{1}{\alpha}}$ with the degrees of freedom

$$\mathrm{df}_\lambda := \mathbb{E}\left[\left\|\mathbf{H}_\lambda(f_{\mathcal{H}})^{-\frac{1}{2}}\nabla\ell_Z(f_{\mathcal{H}})\right\|^2\right] \tag{5}$$

and $\mathbf{H}_\lambda(f) := \mathbf{H}(f) + \lambda I$.

The *degrees of freedom* term df_λ is also called *Fisher information* (Van der Vaart, 2000), appears in spline smoothing (Wahba, 1990), and reduces to the *effective dimension* (Caponnetto et al., 2005) for the square loss. In the latter case, a bigger α implies that fewer eigenvectors are needed to approximate elements from the image space of T with a given capacity (Blanchard & Mücke, 2018).

Of course, it is not possible to achieve fast convergence with *any* loss function. We therefore focus on the large class of convex *pseudo self-concordant* losses, see Bach (2010); Ostrovskii & Bach (2021).

Assumption 3 (pseudo self-concordance)**.** For any $y \in \mathcal{Y}$, the function $\ell_y : \mathbb{R} \to \mathbb{R}$ defined by $\ell_y(\eta) := \ell(y,\eta)$ is convex, three times differentiable and satisfies

$$\left|\ell_y'''(\eta)\right| \leq \ell_y''(\eta). \tag{6}$$

It is important to note that all methods in Example 2 (cf. Bach (2010) for LR) satisfy Assumption 3. This allows us to make the following statement.

Theorem 1. *Let the Assumptions 1–4 be satisfied, let* $t \in \mathbb{N}\setminus\{0\}$, $\delta \in (0,1]$ *and denote by* $s := \min\{r+\frac{1}{2}, t\}$. *Then there exist quantities* $c, C > 0$ *not depending on* m, n *such that the minimizer* $f_{\mathbf{z}}^{\lambda,t}$ *of Eq. (4) with*

$$\lambda = c(m+n)^{-\frac{\alpha}{1+\alpha(2r+1)}} \tag{7}$$

satisfies

$$B_F(\beta, g(f_{\mathbf{z}}^{\lambda,t})) - B_F(\beta, g(f_{\mathcal{H}})) \leq C \cdot (m+n)^{-\frac{2s\alpha}{2s\alpha+1}}$$

with probability at least $1-\delta$ *and for large enough sample sizes* m, n.

Theorem 1 provides a fast error rate whenever $t \geq r + \frac{1}{2}$. For learning problems of high regularity $r > \frac{1}{2}$, an iteration number $t > 1$ is required. Otherwise the error will saturate at a rate of $(m+n)^{-\frac{2}{3}}$, which is slower than the achievable rate $(m+n)^{-\frac{2r\alpha+\alpha}{2r\alpha+\alpha+1}}$.

Remark 1. For $r \leq \frac{1}{2}, t = 1$, Theorem 1 recovers the state-of-the-art error rate of Zellinger et al. (2023). In addition, it extends the state of the art for current approaches for $r > \frac{1}{2}, t = 1$. In case of square loss, it is optimal (cf. Caponnetto & De Vito (2007); Blanchard & Mücke (2018)). For $r > \frac{1}{2}, t > 1$, Theorem 1 provides the first non-saturating error rate for density ratio estimation.

Remark 2. The parameter choice in Eq. (7) balances a bias and a variance term in a tight error bound (Marteau-Ferey et al., 2019b) and is thus rate optimal, see, e.g., Lu & Pereverzev (2013, Section 4.3). Eq. (7) is *a priori* and requires the knowledge of the regularity index r. This issue can be resolved using an *a posteriori* parameter choice rule, e.g., De Vito et al. (2010); Zellinger et al. (2023).

Remark 3. For nonlinear CG algorithms and strictly convex smooth objectives, $\log(\varepsilon^{-1})$ gradient and function evaluations are required to achieve an error tolerance ε, see e.g. Neumaier et al. (2022); Chan-Renous-Legoubin & Royer (2022). Combining these findings with arguments from Beugnot et al. (2021, Section 3.2), we see that ε influences Theorem 1 in an additive way, i.e. with probability at least $1-\delta$,

$$B_F(\beta, g(\bar{f}_{\mathbf{z}}^{\lambda,t})) - B_F(\beta, g(f_{\mathcal{H}})) \leq \tilde{C}\Big((m+n)^{-\frac{2s\alpha}{2s\alpha+1}} + \varepsilon\Big) \quad (8)$$

for a numerical approximation $\bar{f}_{\mathbf{z}}^{\lambda,t}$ of $f_{\mathbf{z}}^{\lambda,t}$ computed as described in Subsection 4.1 and $\tilde{C} > 0$ independent of n, m and ε, see Appendix B for details.

However, as it is typically the case in learning theory (Vapnik, 2013; Zhang et al., 2021), the constant C in Theorem 1 can be quite large. Therefore, we also test the performance of the proposed iterated regularization on benchmark datasets.

6. Empirical Evaluations

We investigate the performance of iterated density ratio estimation regarding three aspects below. We include additional experiments conducted during the rebuttal period that helped to improve our work. More precisely, we compare to a SOTA domain adaptation method (Dinu et al., 2023), combine our approach with telescoping from Rhodes et al. (2020), and extend our approach to Deep Learning methods. Please find further details in Appendix E.

Sample convergence in highly regular problems. To investigate the sample convergence of iterated approaches compared to their non-iterated regularized versions, i.e., Theorem 1), we follow the study of Beugnot et al. (2021). From their dataset, density ratios with known smoothness can be extracted, see dataset description below.

Accuracy improvement for known density ratios. To study the accuracy of the iteratively regularized estimations of density ratios, we follow investigations of Kanamori et al. (2012b) and construct high dimensional data with exact known density ratios. Unfortunately, for real-world problems, exact density ratios are not available and the question arises whether synthetic data is too special. In a small ablation study, we therefore also indicate performance improvements of iteration on a synthetically constructed density ratio.

Importance weighted ensembling in deep domain adaptation. To test the effect of iterated regularization in a real real world setting we rely on large-scale (over 9000 trained neural networks) state-of-the-art experiments for re-solving parameter choice issues in unsupervised domain adaptation. There, we are given two datasets, a source dataset $\mathbf{x}' \subset \mathcal{X}$ drawn from Q with labels $\mathbf{y}' \subset \mathbb{R}$ and an unlabeled target dataset $\mathbf{x} \subset \mathcal{X}$ drawn from P without labels. The goal is to learn a model $f : \mathcal{X} \to \mathcal{Y}$ with low risk $\mathcal{R}(f) := \mathbb{E}_{(x,y)\sim P}[\ell(y, f(x))]$ on future target data from P. In the state-of-the-art approach of Dinu et al. (2023) the key problem of choosing algorithm parameters in this setting is approached by ensemble learning. More precisely, for a given domain adaptation algorithm $A((\mathbf{x}', \mathbf{y}'), \mathbf{x}) = f^{\alpha_1,\dots,\alpha_b}$ with b different parameters $\alpha_1, \dots, \alpha_b$, they compute several model candidates $f_1, \dots, f_l$ with l different parameter settings, and then compute a simple ensemble model $A_{c_1,\dots,c_l}(x) = \sum_{i=1}^{l} c_i^* f_i(x)$ out of $f_1, \dots, f_l$ using the minimizers $c_1^*, \dots, c_l^*$ of

$$\min_{c_1,\dots,c_l \in \mathbb{R}} \frac{1}{|\mathbf{x}'|} \sum_{(x',y') \in (\mathbf{x},\mathbf{y})} \widehat{\beta}(x') \cdot \ell(y', A_{c_1,\dots,c_l}(x')) \quad (9)$$

with an estimate $\widehat{\beta}$ of the density ratio $\beta = \frac{\mathrm{d}P}{\mathrm{d}Q}$. We refer to Sugiyama et al. (2007) as a motivation for using β as importance weight in domain adaptation.

6.1. Datasets

Known regularity To illustrate Theorem 1, we adapt an example of Beugnot et al. (2021, Section 4) to den-

sity ratio estimation. In this example, the regularity index r and the capacity parameter α are known by design. Let therefore $\mathcal{X} = [0,1]$ and the kernel of $\mathcal{H}$ given by $k(x,y) = h_\alpha(x,y) = 1 + \sum_{l\in\mathbb{Z}\setminus 0} \frac{e^{2il\pi(x-y)}}{|l|^\alpha}$, for which an explicit formula is known for even α (Wahba, 1990, Page 22), which is essentially determined by Bernoulli polynomials (Olver et al., 2010, Section 24.2). If we fix $\rho_\mathcal{X}$ as the uniform distribution, ℓ the logistic loss and $f_\mathcal{H}(x) = h_{(r+\frac{1}{2})\alpha+\frac{1}{2}}(0,x)$, then $f_\mathcal{H} \in \mathcal{H}$ and Assumptions 1 and 2 are satisfied (Rudi & Rosasco, 2017). Moreover, it is shown in Ciliberto et al. (2020, Lemma A1) that

$$f_\mathcal{H}(x) = \arg\min_z \mathbb{E}_{y\sim\rho(y|x)}[\ell(y,z)]. \tag{10}$$

From Eq. (10) we obtain (Beugnot et al., 2021, Appendix E3) $\rho(y|x) = \frac{1}{1+e^{-yf_\mathcal{H}(x)}}$. To obtain a formula for P and Q, we apply Bayes' Theorem: $\rho_\mathcal{Y}(y) = \int_\mathcal{X} \frac{1}{1+e^{-yf_\mathcal{H}(x)}} d\rho(x)$, $P(x) = \rho(x|y=1) = \frac{\rho(y=1|x)}{\rho(y=1)}$ and $Q(x) = \rho(x|y=-1) = \frac{\rho(y=-1|x)}{\rho(y=-1)}$. However, the inverse link in Example 2 changes from $\Psi^{-1}(v) = \frac{1}{1+e^{-v}}$ to the new link function (Menon & Ong, 2016, Lemma 5)

$$\widetilde{\Psi}^{-1}(v) = \frac{\Psi^{-1}(v)\pi}{\Psi^{-1}(v)/\pi(1-2\pi)+1-\pi}, \pi = \rho_\mathcal{Y}(y=1),$$

resulting in the density ratio $\beta(x) = \frac{1-\pi}{\pi}\frac{\rho(y=1|x)}{1-\rho(y=1|x)}$ and a different B_F, see Menon & Ong (2016, Section 4.2). We investigate the sample convergence in the distribution-independent error $\left\|\beta - g(f_\mathbf{z}^{\lambda,t})\right\|_{L^1([0,1])}$.

In addition to the dataset adapted from Beugnot et al. (2021), we also construct density ratios between one-dimensional Gaussian mixtures P and a Gaussian distribution Q, with the goal of approximation in a Gaussian RKHS, see Figure 1 (right). A greater number of components allows for rougher density ratios, according to Assumption 1. This may be observed, e.g., from the Fourier decomposition of $f_\mathcal{H}$ with respect to the eigenbasis associated to $\mathbf{H}(f_\mathcal{H})$. A higher number of Gaussian components in P intuitively yields heavier oscillations, and therefore emphasizes higher order Fourier modes. This, in turn, gives a smaller decay rate of the associated Fourier coefficients, implying a reduced regularity according to Assumption 1.

Known density ratios Following Kanamori et al. (2012b), we generate ten different datasets using Gaussian mixture models in the 50-dimensional space with different numbers $\{1,2,3\}$ of components for source and target distribution. Each component has different a mean in $[0,0.5]^{50}$ and accordingly sampled covariance matrices. In all datasets, we have full knowledge of the exact density ratio, as it is calculated as the fraction of the Gaussian mixtures, see Appendix D. To debate whether our method is only suited for Gaussian mixtures, we follow Kanamori et al. (2012b, Section 5.3) and utilize the breast cancer dataset (Street et al., 1993) consisting of 569 samples in the 30-dimensional space with binary labels. There, a target density ratio model β is constructed using a binary support vector machine (Platt et al., 1999) to estimate the class posterior, and, by subsequently re-labeling the miss-labeled data. Note, that the i.i.d. assumption of statistical learning is violated in this case, making the above process more an ablation study than a statistical performance benchmark.

Domain adaptation: text data The Amazon reviews dataset (Blitzer et al., 2006) consists of bag-of-words representations of text reviews from four domains (books, DVDs, kitchen, and electronics) with binary labels indicating the class of review. Twelve domain adaptation tasks are constructed using every domain, once as source domain and once as target domain.

Domain adaptation: image data The DomainNet-2019 dataset (Peng et al., 2019) consists of six different image domains (Quickdraw: Q, Real: R, Clipart: C, Sketch: S, Infograph: I, and Painting: P). We follow Zellinger et al. (2021) and rely on the reduced version of DomainNet-2019, referred to as MiniDomainNet, which reduces the number of classes to the top five largest representatives in the training set across all six domains. To further improve computation time, we rely on a ImageNet (Krizhevsky et al., 2012) pre-trained ResNet-18 (He et al., 2016) backbone. Therefore, we assume that the backbone has learned lower-level filters suitable for the "real" image category, and we only need to adapt to the remaining five domains (Clipart, Sketch,...). This results in five domain adaptation tasks.

Domain adaptation: sensory data The Heterogeneity Human Activity Recognition (Stisen et al., 2015) dataset investigate sensor-, device- and workload-specific heterogeneities using 36 smartphones and smartwatches, consisting of 13 different device models from four manufacturers. In total, we used all fiv domain adaptation scenarios from Dinu et al. (2023, Tables 25 and 26).

6.2. Methods

We test our suggested iterated regularization strategy for the four different baseline approaches in Example 1: KuLSIF (Kanamori et al., 2009), LR (Bickel et al., 2009), Exp (Menon & Ong, 2016) and SQ (Menon & Ong, 2016). Except for KuLSIF, which has a closed form solution, we rely on the CG method implemented in Python Scipy (Virtanen et al., 2020) with Polak-Ribiere line search scheme (Hager & Zhang, 2006).

In domain adaptation experiments we follow the evaluation protocol of Dinu et al. (2023). That is, we compute

	Geometric Figures							
	No Iteration				Iteration			
Dataset	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
c3,d1.70	8.616(±0.011)	8.322(±0.009)	8.840(±0.021)	9.170(±0.011)	**8.605(±0.005)**	**8.827(±0.014)**	8.840(±0.021)	**8.840(±0.010)**
c2,d1.72	13.031(±0.005)	12.994(±0.015)	13.255(±0.013)	13.537(±0.027)	**13.026(±0.009)**	**12.855(±0.017)**	13.255(±0.011)	**13.255(±0.032)**
c2,d1.59	12.625(±0.005)	19.748(±0.037)	12.829(±0.014)	13.056(±0.015)	**12.622(±0.007)**	**12.521(±0.051)**	12.829(±0.014)	**12.829(±0.012)**
c1,d1.55	11.813(±0.007)	14.477(±0.103)	12.001(±0.023)	12.179(±0.013)	**11.805(±0.006)**	**11.552(±0.109)**	12.001(±0.023)	**12.001(±0.019)**
c2,d1.78	9.632(±0.003)	18.008(±0.069)	9.802(±0.035)	9.990(±0.006)	**9.626(±0.019)**	19.895(±0.021)	9.802(±0.037)	**9.802(±0.017)**
c2,d1.55	10.371(±0.007)	9.774(±0.019)	10.555(±0.059)	10.757(±0.023)	**10.363(±0.003)**	**9.646(±0.017)**	10.555(±0.059)	**10.555(±0.025)**
c3,d1.57	12.014(±0.003)	70.827(±0.926)	12.214(±0.037)	14.048(±0.029)	**12.006(±0.007)**	**12.082(±0.014)**	12.214(±0.019)	**12.214(±0.046)**
c2,d1.61	11.614(±0.004)	11.282(±0.034)	11.800(±0.008)	12.242(±0.007)	**11.609(±0.004)**	**15.275(±0.044)**	11.800(±0.009)	**11.800(±0.008)**
c3,d1.46	12.803(±0.009)	12.616(±0.008)	12.971(±0.007)	13.159(±0.006)	**12.795(±0.005)**	**12.218(±0.013)**	12.971(±0.004)	**12.971(±0.004)**
c1,d1.63	9.527(±0.006)	9.704(±0.009)	9.732(±0.014)	9.965(±0.015)	**9.517(±0.010)**	14.026(±0.016)	9.732(±0.010)	**9.732(±0.008)**
Avg	11.205(±0.006)	18.775(±0.123)	11.400(±0.023)	11.810(±0.015)	**11.198(±0.008)**	**12.890(±0.032)**	11.400(±0.021)	**11.400(±0.018)**

Table 1. Mean and standard deviation (after ±) of twice the Bregman divergence error on the geometrically constructed datasets following Kanamori et al. (2012b) over ten different sample draws from P and Q.

an ensemble of different deep neural networks using the importance weighted functional regression approach. As importance weight, we use the estimated density ratio of the iterated and non-iterated versions of KuLSIF, LR, Exp, and SQ.

To get the full picture of the impact on different deep domain adaptation methods, we compute ensemble model candidates for 11 domain adaptation algorithms: Adversarial Spectral Kernel Matching (AdvSKM) (Liu & Xue, 2021), Deep Domain Confusion (DDC) (Tzeng et al., 2014), Correlation Alignment via Deep Neural Networks (Deep-Coral) (Sun et al., 2017), Central Moment Discrepancy (CMD) (Zellinger et al., 2017), Higher-order Moment Matching (HoMM) (Chen et al., 2020), Minimum Discrepancy Estimation for Deep Domain Adaptation (MMDA) (Rahman et al., 2020), Deep Subdomain Adaptation (DSAN) (Zhu et al., 2021), Domain-Adversarial Neural Networks (DANN) (Ganin et al., 2016), Conditional Adversarial Domain Adaptation (CDAN) (Long et al., 2018), A DIRT-T Approach to Unsupervised Domain Adaptation (DIRT) (Shu et al., 2018) and Convolutional deep Domain Adaptation model for Time-Series data (CoDATS) (Wilson et al., 2020).

In total, in this benchmark, we trained more than 9000 deep neural network models and we evaluated the whole aggregation benchmark of Dinu et al. (2023) for Amazon Review, MiniDomainNet, and HHAR for each of the eight (Example 1 with and without iteration) density ratio estimation methods.

6.3. Results

Summary In general, we observe a consistent improvement of iterated density ratio estimation compared to its related non-iterative methods, in all, synthetic, numerically controlled and benchmark domain adaptation experiments. That is, iterative regularization is always suggested when enough computational resources are available. One highlight is, that iteration almost never leads to worse results, i.e., overfitting, compared to the non-iterated version, when the same cross-validation procedure is used to select the regularization parameter as for selecting the number of iterations.

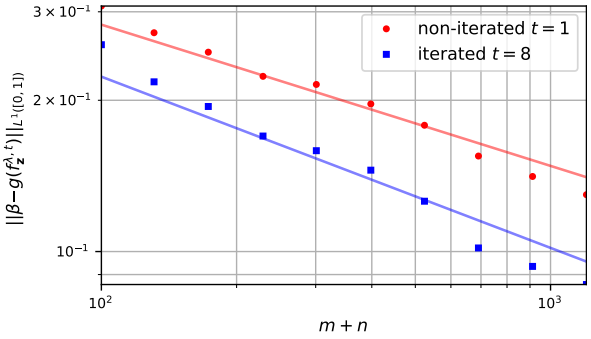

Figure 2. Error $\|\beta - g(f_{\mathbf{z}}^{\lambda,t})\|_{L^1([0,1])}$ of estimating β by $g(f_{\mathbf{z}}^{\lambda,t})$ as a function of sample size $m + n$ on dataset of Beugnot et al. (2021). Slope for iterated estimate ($t = 8$, blue) is steeper, as suggested by Theorem 1.

Results for highly regular problems For the highly regular problems adapted from Beugnot et al. (2021), we observe, for increasing sample size, a trend as predicted by Theorem 1, see Figure 2. This result suggests that our approach successfully extends saturation prevention from supervised regression (Beugnot et al., 2021; Li et al., 2022) to density ratio estimation. In addition, in the case of a Gaussian mixture P and a Gaussian distribution Q in Figure 1 (right), we observe a clear improvement of iteratively regularized KuLSIF compared to KuLSIF. The significance of the improvement (non-intersecting confidence intervals) becomes more apparent for increasing problem regularity (lower number of components).

Results for known density ratio On all geometrically constructed learning problems, three (KuLSIF, Exp, SQ) out of four approaches are improved by iterated regularization.

	Domain Adaptation: MiniDomainNet							
	No Iteration				Iteration			
DA-Method	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
MMDA	0.527(±0.006)	0.528(±0.009)	0.528(±0.012)	0.062(±0.010)	0.527(±0.007)	0.528(±0.009)	0.528(±0.009)	0.062(±0.012)
CoDATS	0.536(±0.012)	0.532(±0.017)	0.530(±0.023)	0.061(±0.025)	**0.539(±0.012)**	0.532(±0.017)	**0.532(±0.017)**	**0.069(±0.025)**
DANN	0.531(±0.010)	0.522(±0.021)	0.520(±0.021)	**0.060(±0.026)**	**0.539(±0.006)**	0.522(±0.021)	**0.522(±0.021)**	0.059(±0.025)
CDAN	0.531(±0.012)	0.531(±0.015)	0.524(±0.020)	0.062(±0.020)	**0.533(±0.011)**	0.531(±0.015)	**0.531(±0.015)**	0.062(±0.027)
DSAN	0.539(±0.009)	0.532(±0.012)	0.527(±0.019)	0.068(±0.015)	**0.546(±0.005)**	0.532(±0.012)	**0.532(±0.012)**	**0.069(±0.015)**
DIRT	0.517(±0.010)	0.386(±0.013)	**0.520(±0.012)**	**0.082(±0.015)**	**0.515(±0.009)**	0.386(±0.013)	0.519(±0.013)	0.081(±0.014)
AdvSKM	0.516(±0.026)	0.515(±0.177)	0.512(±0.023)	0.072(±0.025)	0.516(±0.028)	0.515(±0.177)	**0.515(±0.024)**	**0.073(±0.024)**
HoMM	0.531(±0.011)	0.529(±0.015)	0.521(±0.015)	0.078(±0.025)	0.531(±0.010)	0.529(±0.015)	**0.529(±0.015)**	0.078(±0.025)
DDC	**0.517(±0.012)**	0.517(±0.013)	0.514(±0.012)	0.074(±0.017)	0.516(±0.010)	0.517(±0.013)	**0.517(±0.013)**	0.074(±0.018)
DeepCoral	0.535(±0.008)	0.528(±0.012)	0.530(±0.015)	0.070(±0.012)	**0.539(±0.006)**	0.528(±0.012)	0.528(±0.012)	0.070(±0.011)
CMD	0.529(±0.009)	0.524(±0.011)	0.521(±0.012)	0.091(±0.013)	**0.533(±0.007)**	0.524(±0.011)	**0.524(±0.011)**	0.091(±0.013)
Avg.	0.528(±0.011)	0.513(±0.029)	0.523(±0.017)	0.071(±0.018)	**0.530(±0.010)**	0.513(±0.029)	**0.525(±0.015)**	**0.072(±0.019)**

	Domain Adaptation: Amazon Reviews							
	No Iteration				Iteration			
DA-Method	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
MMDA	0.786(±0.011)	**0.584(±0.062)**	0.784(±0.010)	0.216(±0.012)	**0.789(±0.010)**	0.581(±0.062)	**0.788(±0.010)**	0.216(±0.012)
CoDATS	0.795(±0.012)	0.552(±0.091)	0.794(±0.010)	**0.209(±0.041)**	**0.798(±0.011)**	**0.562(±0.091)**	**0.796(±0.010)**	0.208(±0.011)
DANN	0.794(±0.013)	**0.555(±0.104)**	0.793(±0.012)	0.207(±0.015)	**0.800(±0.009)**	0.553(±0.101)	**0.798(±0.013)**	**0.232(±0.030)**
CDAN	0.787(±0.012)	0.568(±0.066)	0.787(±0.012)	**0.247(±0.013)**	**0.790(±0.010)**	0.568(±0.066)	**0.789(±0.010)**	0.213(±0.015)
DSAN	0.794(±0.011)	0.561(±0.050)	0.793(±0.012)	**0.207(±0.012)**	**0.800(±0.009)**	**0.566(±0.048)**	**0.796(±0.010)**	0.204(±0.060)
DIRT	0.787(±0.011)	**0.561(±0.093)**	0.788(±0.012)	0.212(±0.015)	**0.794(±0.011)**	0.559(±0.099)	**0.791(±0.012)**	**0.246(±0.011)**
AdvSKM	0.779(±0.011)	**0.541(±0.041)**	0.777(±0.012)	0.223(±0.010)	**0.780(±0.008)**	0.540(±0.039)	**0.779(±0.009)**	0.223(±0.067)
HOMM	0.777(±0.011)	0.558(±0.098)	0.774(±0.013)	**0.227(±0.015)**	0.777(±0.010)	**0.562(±0.104)**	**0.776(±0.011)**	0.224(±0.010)
DDC	0.780(±0.010)	0.568(±0.094)	0.778(±0.011)	**0.226(±0.016)**	0.780(±0.010)	**0.572(±0.100)**	**0.779(±0.011)**	0.222(±0.010)
DeepCoral	0.784(±0.009)	0.569(±0.094)	0.781(±0.011)	**0.223(±0.013)**	0.784(±0.009)	**0.573(±0.098)**	**0.783(±0.010)**	0.217(±0.009)
CMD	0.790(±0.010)	**0.577(±0.103)**	0.786(±0.010)	0.215(±0.013)	**0.794(±0.008)**	0.574(±0.100)	**0.789(±0.008)**	**0.224(±0.011)**
Avg.	0.787(±0.011)	0.563(±0.082)	0.785(±0.011)	0.219(±0.016)	**0.790(±0.010)**	**0.565(±0.083)**	**0.788(±0.011)**	**0.221(±0.023)**

	Domain Adaptation: HHAR							
	No Iteration				Iteration			
DA-Method	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
MMDA	0.780(±0.007)	0.670(±0.013)	0.773(±0.018)	0.024(±0.006)	0.780(±0.007)	**0.679(±0.018)**	0.773(±0.018)	0.024(±0.006)
CoDATS	0.723(±0.029)	0.776(±0.166)	0.779(±0.029)	0.002(±0.007)	**0.789(±0.030)**	**0.779(±0.113)**	0.779(±0.029)	0.002(±0.007)
DANN	0.697(±0.170)	0.785(±0.015)	0.795(±0.016)	0.001(±0.004)	**0.796(±0.008)**	**0.795(±0.016)**	0.795(±0.016)	0.001(±0.004)
CDAN	0.792(±0.130)	0.706(±0.021)	0.788(±0.021)	0.012(±0.003)	**0.791(±0.015)**	**0.739(±0.022)**	0.788(±0.021)	0.012(±0.003)
DSAN	0.754(±0.179)	0.528(±0.024)	0.792(±0.021)	0.002(±0.002)	**0.818(±0.010)**	0.528(±0.021)	0.792(±0.021)	0.002(±0.002)
DIRT	0.731(±0.017)	0.790(±0.215)	0.790(±0.019)	0.004(±0.012)	**0.806(±0.017)**	**0.793(±0.216)**	0.790(±0.019)	0.004(±0.012)
AdvSKM	0.752(±0.084)	0.746(±0.040)	0.746(±0.023)	0.004(±0.006)	**0.752(±0.024)**	0.746(±0.027)	0.746(±0.023)	0.004(±0.006)
HoMM	0.759(±0.128)	0.662(±0.228)	0.754(±0.015)	0.007(±0.003)	**0.759(±0.012)**	**0.665(±0.228)**	0.754(±0.015)	0.007(±0.003)
DDC	0.748(±0.115)	0.454(±0.021)	0.744(±0.015)	0.016(±0.005)	**0.748(±0.011)**	0.454(±0.016)	0.744(±0.015)	0.016(±0.005)
DeepCoral	0.701(±0.009)	0.749(±0.096)	0.758(±0.021)	0.006(±0.008)	**0.764(±0.009)**	**0.758(±0.095)**	0.758(±0.021)	0.006(±0.008)
CMD	0.671(±0.008)	0.770(±0.098)	0.770(±0.013)	0.015(±0.008)	**0.766(±0.008)**	0.770(±0.092)	0.770(±0.013)	0.015(±0.008)
Avg.	0.737(±0.080)	0.694(±0.085)	0.772(±0.019)	0.008(±0.006)	**0.779(±0.014)**	**0.700(±0.079)**	0.772(±0.019)	0.008(±0.006)

Table 2. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet, Amazon Reviews and HHAR datasets over three different random initialization of model weights and several domain adaptation tasks.

In particular, KuLSIF and SQ are always improved and Exp is improved in eight out of ten cases. The LR approach is not improved by iterated regularization. However, its performance remained the same. On average over all four approaches, iterated regularization clearly outperforms non-iterated regularization.

For our ablation study on the breast cancer data, we observe an error of 1.21 for KuLSIF, 3.47 for Exp, 8.83 for LR and 15000 for SQ. In the case of SQ, we could not obtain a convergence to lower error, suggesting a closer look with regard to its regularization parameter ranges. However, iterated KuLSIF achieves an error of 1.19, iterated Exp 3.17, iterated LR 3.94 and iterated SQ 100. That is, our iterated regularization approach always improves the results despite the fact that the data is not drawn from Gaussian mixtures.

Results for domain adaptation The detailed results are reported in Tables 4–36. In 9 out of 12 cases (averaged over all tasks and domain adaptation methods), iterated regularization improves its related non-iteratively regularized vesions. In 3 cases it has the same performance. In the same average scenario, it never reduces the performance, i.e., causes overfitting. In more detail, in the case of text data (AmazonReviews), all four density ratio methods (KuLSIF, LR, Exp, SQ) are improved on average across all domain adaptation methods and datasets, see Tables 15–25 for detailed results. In the case of image data (MiniDomainNet), three (KuLSIF, LR, SQ) out of four methods are improved in average over all datsets and tasks, while for Exp, average performance remains the same when the regularization is done iteratively. For time-series data (HHAR) KuLSIF and Exp, averaged across all domain adaptation methods and datasets, are improved while LR and SQ have, on average, the same performance.

Further observations Although LR outperforms KuLSIF in 50% of the cases on time series data (see Tables 26–36), the iteratively regularized KuLSIF method outperforms LR in 11 out of 12 cases. This additionally underpins the importance of considering iterated regularization strategies in density ratio estimation. Moreover, in all Tables 4–36 in Appendix E.2, the maximum average performance is achieved by an iterated method, either iterated KuLSIF, iterated LR, iterated Exp or iterated SQ.

7. Conclusion and Future Work

In this work, we showed that common applications of density ratio estimation methods suffer from saturation, an issue preventing learning methods from achieving fast convergence rates in theoretical error bounds.

Our study contained four main parts: Part (a) extends state-of-the-art error bounds from kernel regression to density ratio estimation, leading to optimal, i.e., improvable, error rates in the case of square loss. The error rates suffer from saturation. Part (b) introduces iterated regularization for a large class of density ratio estimation methods, proving that saturation can be prevented. Part (c) empirically tests the iterated approaches for granting improvements in numerical datasets with known density ratio. Part (d) evaluates the performance of the iterated regularization in practical problems of ensemble learning in unsupervised domain adaptation. In all four parts of our study, clear advantages of the proposed approach could be identified.

One important future aspect is parameter choice, especially under the requirement of reliable methods with theoretical error guarantees, since our approach introduces the number of iterations as a new parameter. We also plan to improve our (time-invariant) time-series domain adaptation using the time-dependent Bregman divergence approach of Kim et al. (2023).

A reference implementation of the methods presented in this paper is available at: https://github.com/lugruber/dre_iter_reg.

Acknowledgements

We thank Otmar Scherzer, Sergei Pereverzyev, Stefan Kindermann, Marius-Constantin Dinu and four anonymous reviewers for helpful comments. The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. We thank the projects Medical Cognitive Computing Center (MC3), INCONTROL-RL (FFG-881064), PRIMAL (FFG-873979), S3AI (FFG-872172), DL for GranularFlow (FFG-871302), EPILEPSIA (FFG-892171), AIRI FG 9-N (FWF-36284, FWF-36235), AI4GreenHeatingGrids (FFG- 899943), INTEGRATE (FFG-892418), ELISE (H2020-ICT-2019-3 ID: 951847), Stars4Waters (HORIZON-CL6-2021-CLIMATE-01-01). We thank Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Labs (SAL), FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Verbund AG, GLS (Univ. Waterloo), Software Competence Center Hagenberg GmbH, Borealis AG, TÜV Austria, Frauscher Sensonic, TRUMPF and the NVIDIA Corporation.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bach, F. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4(none):384 – 414, 2010.

Bauer, F., Pereverzev, S., and Rosasco, L. On regularization algorithms in learning theory. *Journal of Complexity*, 23 (1):52–72, 2007.

Beugnot, G., Mairal, J., and Rudi, A. Beyond tikhonov: faster learning with self-concordant losses, via iterative regularization. *Advances in Neural Information Processing Systems*, 34:28196–28207, 2021.

Bickel, S., Brückner, M., and Scheffer, T. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9), 2009.

Blanchard, G. and Mücke, N. Optimal rates for regulariza-

tion of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18:971–1013, 2018.

Blitzer, J., McDonald, R., and Pereira, F. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 120–128, 2006.

Bregman, L. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

Breiman, L. Bias, variance and arcing classifiers. Technical report, Statistics Department, University of California, 1996.

Buja, A., Stuetzle, W., and Shen, Y. Loss functions for binary class probability estimation and classification: Structure and applications. *Working draft, November*, 3:13, 2005.

Caponnetto, A. and De Vito, E. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Caponnetto, A., Rosasco, L., De Vito, E., and Verri, A. Empirical effective dimension and optimal rates for regularized least squares algorithm. Technical report, Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT, 2005.

Chan-Renous-Legoubin, R. and Royer, C. W. A nonlinear conjugate gradient method with complexity guarantees and its application to nonconvex regression. *EURO Journal on Computational Optimization*, 10:100044, 2022.

Chen, C., Fu, Z., Chen, Z., Jin, S., Cheng, Z., Jin, X., and Hua, X.-S. Homm: Higher-order moment matching for unsupervised domain adaptation. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

Choi, K., Liao, M., and Ermon, S. Featurized density ratio estimation. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 172–182, 2021.

Ciliberto, C., Rosasco, L., and Rudi, A. A general framework for consistent structured prediction with implicit loss embeddings. *The Journal of Machine Learning Research*, 21(1):3852–3918, 2020.

Cohen, A. I. Rate of convergence of several conjugate gradient algorithms. *SIAM Journal on Numerical Analysis*, 9(2):248–259, 1972.

De Vito, E., Pereverzyev, S. V., and Rosasco, L. Adaptive kernel methods using the balancing principle. *Foundations of Computational Mathematics*, 10(4):455–479, 2010.

Dinu, M.-C., Holzleitner, M., Beck, M., Nguyen, D. H., Huber, A., Eghbal-zadeh, H., Moser, B. A., Pereverzyev, S. V., Hochreiter, S., and Zellinger, W. Addressing parameter choice issues in unsupervised domain adaptation by aggregation. *International Conference on Learning Representations*, 2023.

Engl, H., Hanke, M., and Neubauer, A. *Regularization of Inverse Problems*, volume 375. Springer Science & Business Media, 1996a.

Engl, H. W., Hanke, M., and Neubauer, A. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996b.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Gidel, G., Bach, F., and Lacoste-Julien, S. Implicit regularization of discrete gradient dynamics in linear neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f39ae9ff3a81f499230c4126e01f421b-Paper.pdf.

Gizewski, E. R., Mayer, L., Moser, B. A., Nguyen, D. H., Pereverzyev Jr, S., Pereverzyev, S. V., Shepeleva, N., and Zellinger, W. On a regularization of unsupervised domain adaptation in RKHS. *Applied and Computational Harmonic Analysis*, 57:201–227, 2022.

Golub, G. H. and Van Loan, C. F. *Matrix computations*. JHU press, 2013.

Hager, W. W. and Zhang, H. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1): 35–58, 2006.

Hanke, M. and Groetsch, C. W. Nonstationary iterated tikhonov regularization. *Journal of Optimization Theory and Applications*, 98:37–53, 1998.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., and Kanamori, T. Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*, 26:309–336, 2011.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

Kanamori, T., Hido, S., and Sugiyama, M. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

Kanamori, T., Suzuki, T., and Sugiyama, M. f-divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE Transactions on Information Theory*, 58(2):708–720, 2011.

Kanamori, T., Suzuki, T., and Sugiyama, M. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86:335–367, 2012a.

Kanamori, T., Suzuki, T., and Sugiyama, M. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012b.

Kato, M. and Teshima, T. Non-negative bregman divergence minimization for deep direct density ratio estimation. In *International Conference on Machine Learning*, pp. 5320–5333. PMLR, 2021.

Kato, M., Teshima, T., and Honda, J. Learning from positive and unlabeled data with a selection bias. In *International Conference on Learning Representations*, 2019.

Keziou, A. and Leoni-Aubin, S. Test of homogeneity in semiparametric two-sample density ratio models. *Comptes Rendus Mathématique*, 340(12):905–910, 2005.

Kim, D., Kim, Y., Kwon, S. J., Kang, W., and Moon, I.-C. Refining generative process with discriminator guidance in score-based diffusion models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16567–16598. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kim23i.html.

Kim, Y., Na, B., Park, M., Jang, J., Kim, D., Kang, W., and chul Moon, I. Training unbiased diffusion models from biased dataset. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=39cPKijBed.

Kindermann, S., Resmerita, E., and Wolf, T. Multiscale hierarchical decomposition methods for ill-posed problems. *Inverse Problems*, 39(12):125013, 2023.

King, T. J. and Chillingworth, D. Approximation of generalized inverses by iterated regularization. *Numerical Functional Analysis and Optimization*, 1(5):499–513, 1979.

Kiryo, R., Niu, G., du Plessis, M. C., and Sugiyama, M. Positive-unlabeled learning with non-negative risk estimator. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7cce53cf90577442771720a370c3c723-Paper.pdf.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Li, Y., Zhang, H., and Lin, Q. On the saturation effect of kernel ridge regression. In *The Eleventh International Conference on Learning Representations*, 2022.

Lin, J., Rudi, A., Rosasco, L., and Cevher, V. Optimal rates for spectral algorithms with least-squares regression over hilbert spaces. *Applied and Computational Harmonic Analysis*, 48(3):868–890, 2020.

Liu, Q. and Xue, H. Adversarial spectral kernel matching for unsupervised time series domain adaptation. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 30, 2021.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

Lu, S. and Pereverzev, S. V. Regularization theory for ill-posed problems. In *Regularization Theory for Ill-posed Problems*. de Gruyter, 2013.

Marteau-Ferey, U., Bach, F., and Rudi, A. Globally convergent newton methods for ill-conditioned generalized self-concordant losses. *Advances in Neural Information Processing Systems*, 32, 2019a.

Marteau-Ferey, U., Ostrovskii, D., Bach, F., and Rudi, A. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. In *Conference on Learning Theory*, pp. 2294–2340. PMLR, 2019b.

Menon, A. and Ong, C. S. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pp. 304–313. PMLR, 2016.

Meunier, D., Shen, Z., Mollenhauer, M., Gretton, A., and Li, Z. Optimal rates for vector-valued spectral regularization learning algorithms. *arXiv preprint arXiv:2405.14778*, 2024.

Modin, K., Nachman, A., Rondi, L., et al. A multiscale theory for image registration and nonlinear inverse problems. *Advances in Mathematics*, 346:1009–1066, 2019.

Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Neubauer, A. On converse and saturation results for tikhonov regularization of linear ill-posed problems. *SIAM journal on numerical analysis*, 34(2):517–527, 1997.

Neumaier, A., Kimiaei, M., and Azmi, B. Globally linearly convergent nonlinear conjugate gradients without wolfe line search. *preprint*, 2022.

Nguyen, D., Zellinger, W., and Pereverzyev, S. On regularized Radon-Nikodym differentiation. *RICAM-Report 2023-13*, 2023.

Nguyen, X., Wainwright, M. J., and Jordan, M. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. *Advances in Neural Information Processing Systems*, 20, 2007.

Nguyen, X., Wainwright, M. J., and Joradan, M. I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010a. URL http://jmlr.org/papers/v10/bickel09a.html.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010b.

Olver, F. W. J., Lozier, D. W., Boisvert, R. F., and Clark, C. W. *NIST handbook of mathematical functions*. Cambridge University Press, 1 pap/cdr edition, 2010. ISBN 9780521192255.

Osher, S., Burger, M., Goldfarb, D., Xu, J., and Yin, W. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4 (2):460–489, 2005.

Ostrovskii, D. M. and Bach, F. Finite-sample analysis of M-estimators using self-concordance. *Electronic Journal of Statistics*, 15(1):326 – 391, 2021.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Que, Q. and Belkin, M. Inverse density as an inverse problem: The fredholm equation approach. *Advances in neural information processing systems*, 26, 2013.

Ragab, M., Eldele, E., Tan, W. L., Foo, C.-S., Chen, Z., Wu, M., Kwoh, C.-K., and Li, X. Adatime: A benchmarking suite for domain adaptation on time series data. *ACM Transactions on Knowledge Discovery from Data*, 17(8): 1–18, 2023.

Rahman, M. M., Fookes, C., Baktashmotlagh, M., and Sridharan, S. On minimum discrepancy estimation for deep domain adaptation. *Domain Adaptation for Visual Understanding*, 2020.

Reid, M. and Williamson, R. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 2011.

Reid, M. D. and Williamson, R. C. Composite binary losses. *The Journal of Machine Learning Research*, 11:2387–2422, 2010.

Resmerita, E. and Scherzer, O. Error estimates for non-quadratic regularization and the relation to enhancement. *Inverse Problems*, 22(3):801, 2006.

Rhodes, B., Xu, K., and Gutmann, M. U. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.

Rockafellar, R. T. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

Rudi, A. and Rosasco, L. Generalization properties of learning with random features. *Advances in Neural Information Processing Systems*, 30, 2017.

Scherzer, O. Convergence rates of iterated tikhonov regularized solutions of nonlinear iii—posed problems. *Numerische Mathematik*, 66(1):259–279, 1993.

Scherzer, O. and Groetsch, C. Inverse scale space theory for inverse problems. In *International Conference on Scale-Space Theories in Computer Vision*, pp. 317–325. Springer, 2001.

Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., and Lenzen, F. *Variational methods in imaging*, volume 167. Springer, 2009.

Schölkopf, B. and Smola, A. J. Learning with kernels. In *Learning with kernels*. MIT Press, 2002.

Schölkopf, B., Herbrich, R., and Smola, A. J. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pp. 416–426. Springer, 2001.

Schuster, I., Mollenhauer, M., Klus, S., and Muandet, K. Kernel conditional density operators. In *International Conference on Artificial Intelligence and Statistics*, pp. 993–1004. PMLR, 2020.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

Shu, R., Bui, H., Narui, H., and Ermon, S. A dirt-t approach to unsupervised domain adaptation. *International Conference on Learning Representations (ICLR)*, 2018.

Smola, A., Song, L., and Teo, C. H. Relative novelty detection. In *Artificial Intelligence and Statistics*, pp. 536–543. PMLR, 2009.

Srivastava, A., Han, S., Xu, K., Rhodes, B., and Gutmann, M. U. Estimating the density ratio between distributions with high discrepancy using multinomial logistic regression. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=jM8nzUzBWr.

Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., and Jensen, M. M. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 127–140, 2015.

Street, W. N., Wolberg, W. H., and Mangasarian, O. L. Nuclear feature extraction for breast tumor diagnosis. In *Electronic imaging*, 1993. URL https://api.semanticscholar.org/CorpusID:14922543.

Sugiyama, M., Krauledat, M., and Müller, K.-R. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012a.

Sugiyama, M., Suzuki, T., and Kanamori, T. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012b.

Sun, B., Feng, J., and Saenko, K. Correlation alignment for unsupervised domain adaptation. *Domain Adaptation in Computer Vision Applications*, pp. 153–171, 2017.

Tadmor, E., Nezzar, S., and Vese, L. A multiscale image representation using hierarchical (bv, l 2) decompositions. *Multiscale Modeling & Simulation*, 2(4):554–579, 2004.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Van der Vaart, A. W. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Vapnik, V. N. *The nature of statistical learning theory*. Springer science & business media, 2013.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Wahba, G. *Spline models for observational data*. SIAM, 1990.

Wilson, G., Doppa, J. R., and Cook, D. J. Multi-source deep domain adaptation with weak supervision for time-series sensor data. *Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, 2020.

You, K., Wang, X., Long, M., and Jordan, M. Towards accurate model selection in deep unsupervised domain adaptation. In *Proceedings of the International Conference on Machine Learning*, pp. 7124–7133, 2019.

Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., and Saminger-Platz, S. Central moment discrepancy (cmd) for domain-invariant representation learning. *International Conference on Learning Representations*, 2017.

Zellinger, W., Shepeleva, N., Dinu, M.-C., Eghbal-zadeh, H., Nguyen, H. D., Nessler, B., Pereverzyev, S., and Moser, B. A. The balancing principle for parameter choice in distance-regularized domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

Zellinger, W., Kindermann, S., and Pereverzyev, S. V. Adaptive learning of density ratios in RKHS. *Journal of Machine Learning Research*, 24:1–28, 2023.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., and He, Q. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1713–1722, 2021.

A. Proof of Theorem 1

We start with the following common technical assumption, essentially requiring the reproducing kernel Hilbert space $\mathcal{H}$ to be rich enough and regular in a weak sense (cf. Caponnetto & De Vito (2007); Marteau-Ferey et al. (2019b); Gizewski et al. (2022); Zellinger et al. (2023)):

Assumption 4 (technical assumption)**.**

- The kernel k is continuous and bounded $\sup_{x\in\mathcal{X}} \|k(x,\cdot)\| \leq R$.
- An expected risk minimizer $f_{\mathcal{H}} := \arg\min_{f\in\mathcal{H}} \mathcal{R}(f)$ exists.
- The quantities $|\ell_z(0)|, \|\nabla\ell_z(0)\|, \mathrm{Tr}(\nabla^2\ell_z(0))$ are almost surely bounded for any realization z of the random variable Z.

To obtain fast error rates, we rely on a *general self-concordance* property of loss functions (Marteau-Ferey et al., 2019b), which follows from Assumption 3. To this end, we denote by $A[k,h]$ the evaluation of an operator $A : \mathcal{H}\times\mathcal{H} \to \mathbb{R}$ at the arguments (functions) $k, h \in \mathcal{H}$.

Assumption 5 (generalized self-concordance (Marteau-Ferey et al., 2019b))**.** For any $z = (x,y) \in \mathcal{X}\times\mathcal{Y}$, $\ell_z(f)$ is convex in f, three times Fréchet differentiable, and it exists a set $\varphi(z) \subset \mathcal{H}$ such that

$$\forall f \in \mathcal{H}, \forall h,p \in \mathcal{H} : |\nabla^3\ell_z(f)[p,h,h]| \leq \nabla^2\ell_z(f)[h,h] \cdot \sup_{q\in\varphi(z)} |\langle p,q\rangle| \,. \tag{11}$$

Under Assumption 5, an upper bound on the excess risk of iteratively regularized kernel regression can be provided (Beugnot et al., 2021), which reads in our notation as follows.

Lemma 2 (Beugnot et al. (2021, Theorem 1, first part))**.** *Let Assumptions 1, 4 and 5 be satisfied for some loss function ℓ with corresponding set $\varphi(z)$ satisfying $\sup_{g\in\varphi(z)} \|g\| \leq R$ for some $R > 0$. Let further $\delta \in (0,1]$ and fix some $\lambda \in (0, L_0)$, $m+n \geq N$. Then, the following bound on the excess risk of a minimizer of Eq. (4) holds with probability greater than $1-\delta$:*

$$\mathcal{R}(f_{\mathbf{z}}^{\lambda,t}) - \mathcal{R}(f_{\mathcal{H}}) \leq C_{\mathrm{bias}}\lambda^{2s} + C_{\mathrm{var}}\frac{\mathrm{df}_\lambda}{m+n}, \text{ with } s = \min\{r+\frac{1}{2}, t\}. \tag{12}$$

We refer to the supplementary material of Beugnot et al. (2021) for the exact values of $L_0, N, C_{\mathrm{var}}, C_{\mathrm{bias}}$, which depend on $r, \alpha, S, R, t, \delta$ and the distribution ρ but not on the sample size $m+n$.

We will need the following statement.

Lemma 3. *Let $\ell : \mathcal{Y}\times\mathbb{R} \to \mathbb{R}$ satisfy Assumption 3. Then, for any $z = (x,y) \in \mathcal{X}\times\mathcal{Y}$, the functional $\ell_z : \mathcal{H} \to \mathbb{R}$ defined by $\ell_z(f) := \ell(y, f(x))$, $f \in \mathcal{H}$ satisfies Assumption 5.*

Proof. Let $\ell(y,\eta)$ be convex in η, then, for all $z = (x,y) \in \mathcal{X}\times\mathcal{Y}$, $f_1, f_2 \in \mathcal{H}$, $t \in [0,1]$, it holds that

$$\begin{aligned}\ell_z(tf_1 + (1-t)f_2) &= \ell(y, tf_1(x) + (1-t)f_2(x)) \\ &\leq t\ell(y, f_1(x)) + (1-t)\ell(y, f_2(x)) \\ &= t\ell_z(f_1) + (1-t)\ell_z(f_2)\end{aligned}$$

and $\ell_z(f)$ is convex in f.

To show Fréchet differentiability of $\ell_z(f)$, let us introduce the notation $\ell_y(\eta) := \ell(y,\eta)$ which is Fréchet differentiable with $\nabla\ell_y(\eta) = \ell_y'(\eta)$ by definition. We also denote by $B_x(f) = f(x)$ the (linear) evaluation functional, which is bounded for $f \in \mathcal{H}$ and it's Fréchet derivative ∇B_x therefore exists and is again the evaluation functional B_x. Then, by definition, all derivatives in the composition $\nabla\ell_y(B_x(f)) \circ \nabla B_x(f)$ exist and are bounded, and therefore it equals the desired quantity $\nabla\ell_z(f) = \nabla(\ell_y \circ B_x)(f)$ by the chain rule. The second and third Fréchet derivative can be obtained analogously.

Now note that, for all $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\begin{aligned}\left|\nabla^3 \ell_z(f)[p,h,h]\right| &= \left|p(x)h(x)h(x)\ell_y'''(f(x))\right| \\ &= \left|p(x)h(x)h(x)\left|\ell_y'''(f(x))\right|\right| \\ &\leq \left|p(x)h(x)h(x)\ell_y''(f(x))\right| \\ &= \left|p(x)\nabla^2\ell_z(f)[h,h]\right| \\ &= |p(x)|\,\nabla^2\ell_z(f)[h,h],\end{aligned}$$

where the last line follows from the positive semi-definiteness of the Hessian. Note that

$$|p(x)| = |y| \cdot |\langle p, k(x,\cdot)\rangle| = |\langle p, yk(x,\cdot)\rangle| \leq \sup_{q\in\varphi(z)} |\langle p, q\rangle|$$

follows from the reproducing property and $\varphi(z) := \{yk(x,\cdot)\}$. □

We are now able to combine above reasoning to prove Theorem 1.

As we note in Section 3, we neglect pathological loss functions and consider only strictly proper composite ℓ with twice differentiable Bayes risk G. As a consequence, from Lemma 1, we get

$$B_F(\beta, g(f)) - B_F(\beta, g(f_{\mathcal{H}})) = 2\left(\mathcal{R}(f) - \mathcal{R}(f^*) - \mathcal{R}(f_{\mathcal{H}}) + \mathcal{R}(f^*)\right) = 2\left(\mathcal{R}(f) - \mathcal{R}(f_{\mathcal{H}})\right). \tag{13}$$

Our goal is now to apply Lemma 2, which requires that Assumption 5 is satisfied with some $\varphi(z)$ satisfying $\sup_{g\in\varphi(z)} \|g\| \leq R$ and $R > 0$. This follows from Lemma 3 and the fact that $\sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \|yk(x,\cdot)\| \leq R$ by Assumption 4.

Let us choose

$$\lambda_* := L_0(m+n)^{-\frac{\alpha}{2s\alpha+1}}$$

which balances the bias and the variance term in Lemma 2 up to a constant factor (that ensures $\lambda_* \leq L_0$). Then, Lemma 2 can be applied and it holds that

$$B_F(\beta, g(f)) - B_F(\beta, g(f_{\mathcal{H}})) \leq 2\left(C_{\text{bias}}\lambda_*^{2s} + C_{\text{var}}\frac{S\lambda_*^{-\frac{1}{\alpha}}}{m+n}\right) = 2(C_{\text{bias}} + C_{\text{var}}S)(m+n)^{-\frac{2s\alpha}{2s\alpha+1}}$$

with probability at least $1 - 2\delta$ and $m + n \geq N$, for an N independent of $m + n$. Theorem 1 follows by fixing

$$C := 2(C_{\text{bias}} + C_{\text{var}}S).$$

B. Optimization Error

This Section is aimed to provide a more detailed sketch on how one can obtain the error bound Eq. (8). The first quantity we need to consider in this context is the so called Newton decrement, as developed in Marteau-Ferey et al. (2019a). Starting from $f_{\mathbf{z}}^{\lambda,0} = \bar{f}_{\mathbf{z}}^{\lambda,0} = 0$, we define the following expressions for $k > 0$:

$$f_{\mathbf{z}}^{\lambda,t+1} = \operatorname*{arg\,min}_{f\in\mathcal{H}} \frac{1}{|\mathbf{z}|}\sum_{i=1}^{m+n} \ell(y_i, f(x_i)) + \frac{\lambda}{2}\left\|f - f_{\mathbf{z}}^{\lambda,t}\right\|^2 := \operatorname*{arg\,min}_{f\in\mathcal{H}} \widehat{\mathcal{R}}_\lambda^t(f), \quad \nu_\lambda^{t+1}(f) := \left\|\mathbf{H}_\lambda(f)^{-\frac{1}{2}}\nabla\widehat{\mathcal{R}}_\lambda^t(f)\right\|, \tag{14}$$

$$\bar{f}_{\mathbf{z}}^{\lambda,t+1} \approx \operatorname*{arg\,min}_{f\in\mathcal{H}} \frac{1}{|\mathbf{z}|}\sum_{i=1}^{m+n} \ell(y_i, f(x_i)) + \frac{\lambda}{2}\left\|f - \bar{f}_{\mathbf{z}}^{\lambda,t}\right\|^2 := \operatorname*{arg\,min}_{f\in\mathcal{H}} \bar{\mathcal{R}}_\lambda^t(f), \quad \bar{\nu}_\lambda^{t+1}(f) := \left\|\mathbf{H}_\lambda(f)^{-\frac{1}{2}}\nabla\bar{\mathcal{R}}_\lambda^t(f)\right\|, \tag{15}$$

for a numerical approximation $\bar{f}_{\mathbf{z}}^{\lambda,t}$ of $f_{\mathbf{z}}^{\lambda,t}$ computed as described in Subsection 4.1. Next we need to enforce a bound on the true Newton decrement in Eq. (14) when we only have access to $\bar{\mathcal{R}}_\lambda^t$. This has been done in Beugnot et al. (2021, Proposition 1), and we repeat their result in our notation:

Lemma 4. *(Beugnot et al., 2021, Proposition 1) Let $\varepsilon > 0$ the target precision. Assume that we can solve each sub-problem with precision $\bar{\epsilon}_k$:*

$$\forall k \in \{1, \dots, t\}, \quad \bar{\nu}_\lambda^{k-1}\left(\bar{f}_{\mathbf{z}}^{\lambda,k}\right) \leq \bar{\varepsilon}_t = \varepsilon \frac{1.4^{k-t}}{t},$$

and that $\varepsilon \leq \frac{\sqrt{\lambda}}{2R}$. This suffice to achieve an error ε on the target function:

$$\nu_\lambda^{t-1}\left(\bar{f}_{\mathbf{z}}^{\lambda,t}\right) \leq \varepsilon$$

To compute this estimator in practice, one would need to solve t optimization problems with decreasing precision. As the CG-algorithm that we are applying (cf. Section 4.1) has error complexity $\log(\varepsilon)$ (see Remark (3)), the complexity of applying this algorithm iteratively would only be t times bigger than computing a single iteration, so achieving the required small magnitudes of the weighted gradient norms Eq. (15)–(14) is computationally feasible. Analyzing the computational burden in terms of the number of samples required to compute the numerical solution of Eq. (4) would be an interesting avenue for future work. To conclude our investigations on how the theoretical guarantees change by the use of inexact solvers, we can adapt Beugnot et al. (2021, Proposition 2) with the same arguments as used in our proof of Theorem 1:

Lemma 5. *Let $\delta \in (0, 1)$ and assume that the statistical assumptions of Theorem 1 hold as well as the optimization assumptions of Lemma 4. Then, the following bound on the excess risk holds with probability greater than $1 - \delta$:*

$$B_F(\beta, g(\bar{f}_{\mathbf{z}}^{\lambda,t})) - B_F(\beta, g(f_{\mathcal{H}})) \leq \tilde{C}\Big((m+n)^{-\frac{2s\alpha}{2s\alpha+1}} + \varepsilon\Big)$$

with $\tilde{C} > 0$ independent of n, m and ε.

This is all we need to obtain the bound Eq. (8).

C. Derivation of Iterated KuLSIF

Starting from objective (2)

$$f^{\lambda,t+1} := \operatorname*{arg\,min}_{f \in \mathcal{H}} B_F(\beta, g(f)) + \frac{\lambda}{2}\left\|f - f^{\lambda,t}\right\|^2$$

where $f^{\lambda,0} = 0$ we get for KuLSIF with $F(h) = \int_{\mathcal{X}} (h(x) - 1)^2/2 \, \mathrm{d}Q(x)$ and $g(f) = f$

$$\begin{aligned} f^{\lambda,t+1} &= \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{2}\int (\beta - f)^2 dQ + \frac{\lambda}{2}\left\|f - f^{\lambda,t}\right\|^2 \\ &= \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{2}\int ((\beta - f^{\lambda,t}) - f)^2 dQ + \frac{\lambda}{2}\left\|f\right\|^2. \end{aligned}$$

Setting $r_t = \beta - f^{\lambda,t}$ gives

$$\operatorname*{arg\,min}_{f \in \mathcal{H}} \int \frac{1}{2}(r_t - f)^2 dQ + \frac{\lambda}{2}\left\|f\right\|^2$$

Expanding the squares, using $\beta = \frac{dP}{dQ}$ and ignoring the summands only containing β gives:

$$\operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{2}\int f^2 dQ + \int f^{\lambda,t} f dQ - \int f dP + \frac{\lambda}{2}\left\|f\right\|^2.$$

The empirical computation of these quantities results in

$$f_{\mathbf{z}}^{\lambda,t+1} = \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(f(x_i')^2 + f(x_i')f^{\lambda,t}(x_i')) - \frac{1}{m}\sum_{l=1}^{m} f(x_l) + \frac{\lambda}{2}\left\|f\right\|^2 \tag{16}$$

By choosing

$$f_{\mathbf{z}}^{\lambda,t+1} = \sum_{i=1}^{n} \alpha_i^{t+1} k(\cdot, x_i') + \sum_{l=1}^{m} \beta_l^{t+1} k(\cdot, x_l)$$

as in Kanamori et al. (2012a) (following the representer Theorem), plugging this into Eq. (16) and taking the derivatives w.r.t. α_j^{t+1} and β_l^{t+1} and setting them to 0, we get as analytic expressions for the coefficients:

$$\begin{aligned}
\beta^{t+1} &= \frac{t+1}{m\lambda} 1_m \\
\alpha^{t+1} &= \left(\lambda + \frac{1}{n} K_{11}\right)^{-1} \left(\lambda \alpha^t - K_{12} \frac{1}{\lambda m n} 1_m\right)
\end{aligned}$$

where 1_m is the one vector, K_{11} is the gram sub-matrix for data $\mathbf{x}'$ and K_{12} with respect to $\mathbf{x}'$ and $\mathbf{x}$, and $\alpha^0 = 0$ and $\beta^0 = 0$.

D. Dataset with Known Density Ratios

To study the accuracy of the iteratively regularized estimations of density ratios, we follow investigations of Kanamori et al. (2012a) and Nguyen et al. (2010a) to construct high dimensional data with exact known density ratios. Thereby, we both extend and generalize existing settings. For example datasets such as Ringnorm or Twonorm (Breiman, 1996) are special cases of a Gaussian Mixture approach with mean and covariance parameters of the respective mixture modes set accordingly. Similarly, experiments in Nguyen et al. (2010a) can be modeled by a Gaussian Mixture distribution. We make the task more complex by randomly sampling the number, the weights, the expectations and the covariances of mixture components of the underlying distributions from a space with higher dimension (50) compared to existing work. More specifically, we sample the expectations from $[0, 0.5]^{50}$ and the weights from $[0, 1]$ with additional normalization. Each distribution (source, target) gets assigned to a different mixture model. In each experiment there is a maximum component count of 4 components. The randomly sampled $n \in \{1, 2, 3\}$ is the number of components for the source and $4 - n$ for the target distribution.

E. Detailed Empirical Evaluations

E.1. Known density ratios

For each Gaussian mixture dataset we draw 5000 samples from the underlying distributions. The selection and evaluation of the compared density ratio estimation method is done in a typical train/val/test split approach with split ratios $64/16/20$ respectively. The regularization (hyper) parameter λ is selected from $\{10^{-6}, 10^{-5}, \ldots, 10^{4}\}$ and for each experiment 10 replicates are carried out. Accordingly, the number of iteration steps is selected from $\{1, 2, \ldots, 10\}$ based on the respective loss metric. We follow (Kanamori et al., 2012a) in using the Gaussian kernel with kernel width set according to the median heuristic (Schölkopf & Smola, 2002) for all compared density ratio estimation methods.

E.2. Domain Adaptation

The computation of the results for the domain adaptation benchmark experiment is based on gradient-based training of overall 9174 models where we built parts of our results on the codebase of Dinu et al. (2023) which results in the following specifics.

AmazonReviews (text data): 11 methods $\times$ 14 parameters $\times$ 12 domain adaptation tasks $\times$ 3 seeds $=$ 5544 trained models

MiniDomainNet (image data): 11 methods $\times$ 8 parameters $\times$ 5 domain adaptation tasks $\times$ 3 seeds $=$ 1320 trained models

HHAR (sensory data): 11 methods $\times$ 14 parameters $\times$ 5 domain adaptation tasks $\times$ 3 seeds $=$ 2310 trained models

Following Dinu et al. (2023) we used 11 domain adaptation methods from the AdaTime (Ragab et al., 2023) benchmark. For each of these methods and domain adaptation modalities (text, image, sensory) we evaluate all 8 density ratio estimation on all 22 domain adaptation scenarios. For our experiments we follow the same model implementation and experimental setup as in Dinu et al. (2023) which results in the usage of fully connected networks for Amazon Reviews and a pretrained ResNet-18 backbone for MiniDomainNet. For training and selecting the density ratio estimation methods within this pipeline

we perform an additional train/val split of $80/20$ on the datasets that are used for training the domain adaption methods. The regularization (hyper) parameter λ is selected from $\{10^{-6}, 10^{-5}, \dots, 10^{4}\}$ and due to higher computational effort the number of iterations t is selected from $\{1, 5, 10\}$ based on the respective Bregman objective (1). We follow Kanamori et al. (2012a) in using the Gaussian kernel with kernel width set according to the median heuristic (Schölkopf & Smola, 2002) for all compared density ratio estimation methods. Each experiment is run 3 times. To test the performance of the compared methods the classification accuracy on the respective test sets of the target distribution is evaluated. We use the ensemble approach introduced in Dinu et al. (2023) as discussed in Section 6.

E.3. Further Ablation Studies

SOTA Domain Adaptation We compare our method to the state-of-the-art results reported in Dinu et al. (2023). Table 3 shows the results for each domain adaptation method (rows) averaged over 12 domain adaptation tasks and three seeds for the Amazon Reviews dataset. While KuLSIF without our proposed iteration only achieves 0.787 and does not outperform (Dinu et al., 2023), our proposed iterated KuLSIF improves (Dinu et al., 2023) from 0.788 to 0.790.

Domain Adaptation: Amazon Reviews			
DA-Method	Dinu et al. (2023)	KuLSIF	Iterated KuLSIF (ours)
HoMM	$0.788(\pm 0.010)$	$0.777(\pm 0.009)$	$0.777(\pm 0.010)$
AdvSKM	$0.780(\pm 0.009)$	$0.779(\pm 0.011)$	$0.780(\pm 0.008)$
DIRT	$0.787(\pm 0.008)$	$0.787(\pm 0.011)$	$0.794(\pm 0.011)$
DDC	$0.780(\pm 0.010)$	$0.780(\pm 0.010)$	$0.780(\pm 0.010)$
CMD	$0.794(\pm 0.009)$	$0.790(\pm 0.010)$	$0.794(\pm 0.008)$
MMDA	$0.787(\pm 0.011)$	$0.786(\pm 0.011)$	$0.789(\pm 0.010)$
CoDATS	$0.796(\pm 0.009)$	$0.795(\pm 0.012)$	$0.798(\pm 0.011)$
Deep-Coral	$0.785(\pm 0.009)$	$0.784(\pm 0.009)$	$0.784(\pm 0.009)$
CDAN	$0.788(\pm 0.010)$	$0.787(\pm 0.010)$	$0.790(\pm 0.010)$
DANN	$0.797(\pm 0.009)$	$0.794(\pm 0.013)$	$0.800(\pm 0.009)$
DSAN	$0.795(\pm 0.009)$	$0.794(\pm 0.011)$	$0.800(\pm 0.009)$
Avg.	$0.788(\pm 0.009)$	$0.787(\pm 0.011)$	$0.790(\pm 0.010)$

Table 3. Mean and standard deviation (after $\pm$) of target classification accuracy on Amazon Reviews over three different random initialization of model weights and 12 domain adaptation tasks.

Extension to Deep Learning We follow Section 4.1 of Rhodes et al. (2020) to analyze the sample efficiency for challenging (different) densities. More precisely, we sample from an extremely peaked Gaussian $p \sim \mathcal{N}(0, 10^{-6})$ and a broad Gaussian $q \sim \mathcal{N}(0, 1)$. The density ratio estimators are based on both standard logistic regression and multi-layer networks. In Figure 3 (left) we can see that our iteration method improves sample efficiency for logistic regression. Additionally, it can be seen that our approach can be extended to neural network based models for which sample efficiency is improved as well.

Combining Iteration with Telescoping We extend the previous experiment such that we sample from the same respective distributions and combine the telescoping framework of Rhodes et al. (2020) with our iteration method. In Figure 3 (right) it can be seen that telescoping improves the non-iterated density ratio estimator as in Rhodes et al. (2020). Additionally, the combination of telescoping and our iteration method further improves sample efficiency of both approaches.

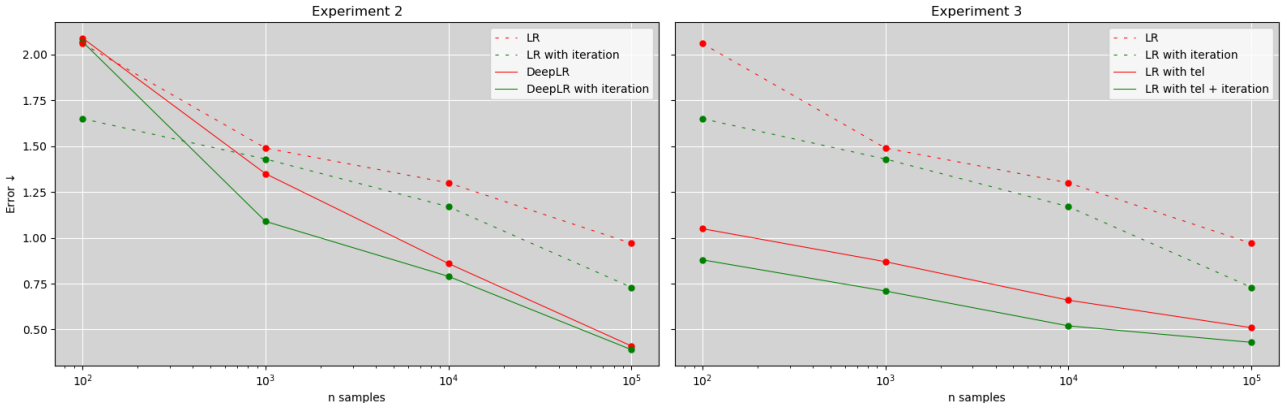

Figure 3. Sample efficiency curves for various density ratio estimators and approaches, the error is measured by L1-norm. Left: Comparison logistic regression and multi-layer network logistic regression density ratio estimators both without and with iteration. Right: Comparison of logistic regression with its iterated version, the telescoping (tel) approach from Rhodes et al. (2020) and the telescoping approach combined with our iterative method.

E.4. Detailed Empirical Results

In this section, we add all result tables for the datasets described in the main paper. Table 4-14 show all results on the MiniDomainNet image datasets, Table 15-25 show all results on the Amazon Reviews text datasets, and Table 26-36 show all results on the HHAR sensory datasets. Each table shows the results for all the respective datasets given one domain adaptation method from AdaTime. Averaging the results in these tables over the scenario datasets and combining them leads to Table 6.1.

	No Iteration				**Iteration**			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.568(±0.008)	0.561(±0.013)	0.552(±0.006)	0.045(±0.008)	0.568(±0.009)	0.561(±0.013)	**0.561(±0.013)**	**0.046(±0.012)**
R → I	**0.382(±0.010)**	0.389(±0.007)	0.385(±0.026)	0.050(±0.004)	0.377(±0.013)	0.389(±0.007)	**0.389(±0.007)**	**0.053(±0.008)**
R → P	0.717(±0.005)	0.712(±0.014)	0.712(±0.010)	0.039(±0.005)	**0.721(±0.003)**	0.712(±0.014)	0.712(±0.014)	0.039(±0.003)
R → Q	0.332(±0.005)	0.325(±0.007)	0.324(±0.011)	0.146(±0.021)	**0.333(±0.005)**	0.325(±0.007)	**0.325(±0.007)**	**0.147(±0.021)**
R → S	**0.584(±0.004)**	0.586(±0.003)	**0.588(±0.005)**	0.080(±0.011)	0.582(±0.004)	0.586(±0.003)	0.586(±0.003)	0.080(±0.013)
Avg.	0.516(±0.006)	0.515(±0.009)	0.512(±0.012)	0.072(±0.010)	0.516(±0.007)	0.515(±0.009)	**0.515(±0.009)**	**0.073(±0.012)**

Table 4. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by AdvSKM (Liu & Xue, 2021).

	No Iteration				**Iteration**			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.592(±0.018)	0.595(±0.029)	0.594(±0.023)	0.031(±0.029)	**0.598(±0.022)**	0.595(±0.029)	**0.595(±0.029)**	**0.032(±0.029)**
R → I	**0.372(±0.013)**	0.381(±0.012)	0.337(±0.056)	0.083(±0.007)	0.371(±0.016)	0.381(±0.012)	**0.381(±0.012)**	0.083(±0.007)
R → P	0.729(±0.004)	0.720(±0.015)	**0.725(±0.011)**	0.053(±0.014)	**0.731(±0.004)**	0.720(±0.015)	0.720(±0.015)	**0.055(±0.017)**
R → Q	0.353(±0.018)	0.352(±0.020)	**0.354(±0.017)**	0.089(±0.058)	**0.356(±0.012)**	0.352(±0.020)	0.352(±0.020)	0.089(±0.058)
R → S	0.609(±0.008)	0.607(±0.012)	**0.611(±0.009)**	0.052(±0.015)	**0.610(±0.008)**	0.607(±0.012)	0.607(±0.012)	0.052(±0.016)
Avg.	0.531(±0.012)	0.531(±0.017)	0.524(±0.023)	0.062(±0.025)	**0.533(±0.012)**	0.531(±0.017)	**0.531(±0.017)**	0.062(±0.025)

Table 5. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by CDAN (Long et al., 2018).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.576(±0.015)	0.560(±0.026)	0.552(±0.017)	0.084(±0.021)	**0.593(±0.009)**	0.560(±0.026)	**0.560(±0.026)**	0.084(±0.021)
R → I	**0.383(±0.009)**	0.391(±0.021)	0.381(±0.033)	**0.083(±0.030)**	0.379(±0.007)	0.391(±0.021)	**0.391(±0.021)**	0.081(±0.028)
R → P	0.739(±0.008)	0.739(±0.016)	0.742(±0.011)	0.030(±0.013)	**0.740(±0.004)**	0.739(±0.016)	0.739(±0.016)	0.030(±0.011)
R → Q	0.354(±0.013)	0.336(±0.030)	0.335(±0.028)	0.179(±0.039)	**0.364(±0.007)**	0.336(±0.030)	**0.336(±0.030)**	0.179(±0.039)
R → S	**0.595(±0.005)**	0.594(±0.010)	0.597(±0.014)	0.080(±0.029)	0.592(±0.003)	0.594(±0.010)	0.594(±0.010)	**0.083(±0.028)**
Avg.	0.529(±0.010)	0.524(±0.021)	0.521(±0.021)	0.091(±0.026)	**0.533(±0.006)**	0.524(±0.021)	**0.524(±0.021)**	0.091(±0.025)

Table 6. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by CMD (Zellinger et al., 2017).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.597(±0.011)	0.577(±0.010)	**0.594(±0.010)**	**0.057(±0.020)**	**0.604(±0.013)**	0.577(±0.010)	0.577(±0.010)	0.056(±0.020)
R → I	**0.366(±0.012)**	0.368(±0.023)	0.349(±0.038)	0.095(±0.035)	0.360(±0.005)	0.368(±0.023)	**0.368(±0.023)**	**0.141(±0.071)**
R → P	0.739(±0.015)	0.733(±0.018)	0.737(±0.017)	**0.038(±0.019)**	**0.743(±0.016)**	0.733(±0.018)	0.733(±0.018)	0.037(±0.017)
R → Q	0.361(±0.013)	0.364(±0.014)	0.355(±0.027)	0.065(±0.019)	**0.366(±0.010)**	0.364(±0.014)	**0.364(±0.014)**	**0.066(±0.019)**
R → S	0.619(±0.008)	0.616(±0.008)	0.616(±0.009)	**0.049(±0.007)**	**0.622(±0.009)**	0.616(±0.008)	0.616(±0.008)	0.048(±0.006)
Avg.	0.536(±0.012)	0.532(±0.015)	0.530(±0.020)	0.061(±0.020)	**0.539(±0.011)**	0.532(±0.015)	**0.532(±0.015)**	**0.069(±0.027)**

Table 7. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by CoDATS (Wilson et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.591(±0.015)	0.576(±0.013)	**0.579(±0.012)**	**0.052(±0.026)**	**0.607(±0.008)**	0.576(±0.013)	0.576(±0.013)	0.050(±0.025)
R → I	0.374(±0.009)	0.372(±0.011)	0.369(±0.026)	**0.058(±0.010)**	0.374(±0.003)	0.372(±0.011)	**0.372(±0.011)**	0.056(±0.011)
R → P	0.720(±0.007)	0.716(±0.008)	0.699(±0.032)	**0.044(±0.012)**	**0.721(±0.008)**	0.716(±0.008)	**0.716(±0.008)**	0.044(±0.008)
R → Q	0.357(±0.009)	0.340(±0.016)	0.337(±0.016)	**0.078(±0.006)**	**0.374(±0.003)**	0.340(±0.016)	**0.340(±0.016)**	0.078(±0.006)
R → S	0.613(±0.007)	0.607(±0.010)	**0.614(±0.007)**	**0.069(±0.022)**	**0.617(±0.005)**	0.607(±0.010)	0.607(±0.010)	0.069(±0.022)
Avg.	0.531(±0.009)	0.522(±0.012)	0.520(±0.019)	**0.060(±0.015)**	**0.539(±0.005)**	0.522(±0.012)	**0.522(±0.012)**	0.059(±0.015)

Table 8. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by DANN (Ganin et al., 2016).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	**0.569(±0.027)**	0.571(±0.029)	0.563(±0.028)	0.042(±0.025)	0.568(±0.025)	0.571(±0.029)	**0.571(±0.029)**	**0.044(±0.024)**
R → I	**0.389(±0.011)**	0.391(±0.016)	0.390(±0.011)	0.047(±0.012)	0.385(±0.008)	0.391(±0.016)	**0.391(±0.016)**	**0.049(±0.014)**
R → P	**0.717(±0.002)**	0.721(±0.003)	0.715(±0.002)	0.034(±0.008)	0.715(±0.002)	0.721(±0.003)	**0.721(±0.003)**	0.034(±0.005)
R → Q	0.333(±0.004)	0.323(±0.013)	0.324(±0.013)	0.156(±0.017)	**0.335(±0.003)**	0.323(±0.013)	0.323(±0.013)	0.156(±0.018)
R → S	**0.580(±0.005)**	0.578(±0.005)	0.579(±0.004)	0.089(±0.012)	0.579(±0.007)	0.578(±0.005)	0.578(±0.005)	0.089(±0.011)
Avg.	**0.517(±0.010)**	0.517(±0.013)	0.514(±0.012)	0.074(±0.015)	0.516(±0.009)	0.517(±0.013)	**0.517(±0.013)**	0.074(±0.014)

Table 9. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by DDC (Tzeng et al., 2014).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.588(±0.021)	0.569(±0.016)	0.567(±0.012)	0.051(±0.041)	**0.590(±0.022)**	0.569(±0.016)	**0.569(±0.016)**	0.051(±0.042)
R → I	**0.369(±0.012)**	0.372(±0.008)	**0.381(±0.018)**	0.087(±0.005)	0.367(±0.012)	0.372(±0.008)	0.372(±0.008)	**0.088(±0.006)**
R → P	0.735(±0.011)	0.737(±0.010)	**0.734(±0.005)**	0.034(±0.009)	**0.738(±0.012)**	0.737(±0.010)	**0.737(±0.010)**	0.033(±0.010)
R → Q	0.363(±0.011)	0.348(±0.021)	**0.349(±0.016)**	0.134(±0.018)	**0.380(±0.002)**	0.348(±0.021)	0.348(±0.021)	**0.135(±0.019)**
R → S	0.621(±0.006)	0.616(±0.009)	0.617(±0.009)	0.041(±0.011)	**0.622(±0.001)**	0.616(±0.009)	0.616(±0.009)	**0.042(±0.013)**
Avg.	0.535(±0.012)	0.528(±0.013)	0.530(±0.012)	0.070(±0.017)	**0.539(±0.010)**	0.528(±0.013)	0.528(±0.013)	0.070(±0.018)

Table 10. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by DeepCoral (Sun et al., 2017).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.572(±0.021)	0.474(±0.176)	0.566(±0.022)	**0.081(±0.032)**	**0.577(±0.025)**	0.474(±0.176)	**0.567(±0.018)**	0.079(±0.034)
R → I	0.376(±0.047)	0.236(±0.135)	**0.406(±0.025)**	**0.083(±0.022)**	**0.377(±0.034)**	0.236(±0.135)	0.403(±0.027)	0.076(±0.018)
R → P	**0.714(±0.026)**	0.542(±0.271)	**0.713(±0.025)**	0.023(±0.013)	0.713(±0.030)	0.542(±0.271)	0.712(±0.028)	0.023(±0.010)
R → Q	**0.336(±0.018)**	0.248(±0.055)	**0.329(±0.021)**	0.133(±0.030)	0.325(±0.036)	0.248(±0.055)	0.328(±0.026)	**0.135(±0.028)**
R → S	**0.586(±0.020)**	0.431(±0.245)	**0.587(±0.021)**	0.092(±0.029)	0.583(±0.016)	0.431(±0.245)	0.584(±0.022)	0.092(±0.028)
Avg.	**0.517(±0.026)**	0.386(±0.177)	**0.520(±0.023)**	**0.082(±0.025)**	0.515(±0.028)	0.386(±0.177)	0.519(±0.024)	0.081(±0.024)

Table 11. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by DIRT (Shu et al., 2018).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.623(±0.015)	0.603(±0.026)	0.596(±0.019)	0.034(±0.028)	**0.639(±0.008)**	0.603(±0.026)	**0.603(±0.026)**	**0.036(±0.027)**
R → I	0.374(±0.010)	0.367(±0.008)	0.350(±0.014)	0.102(±0.029)	**0.376(±0.012)**	0.367(±0.008)	**0.367(±0.008)**	**0.104(±0.032)**
R → P	**0.720(±0.007)**	0.724(±0.008)	0.720(±0.008)	0.042(±0.008)	0.719(±0.007)	0.724(±0.008)	**0.724(±0.008)**	**0.044(±0.008)**
R → Q	0.364(±0.012)	0.355(±0.018)	0.354(±0.018)	0.102(±0.034)	**0.372(±0.011)**	0.355(±0.018)	**0.355(±0.018)**	0.102(±0.033)
R → S	0.616(±0.012)	0.609(±0.015)	0.615(±0.016)	0.059(±0.025)	**0.622(±0.013)**	0.609(±0.015)	0.609(±0.015)	**0.061(±0.023)**
Avg.	0.539(±0.011)	0.532(±0.015)	0.527(±0.015)	0.068(±0.025)	**0.546(±0.010)**	0.532(±0.015)	**0.532(±0.015)**	**0.069(±0.025)**

Table 12. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by DSAN (Zhu et al., 2021).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.590(±0.020)	0.588(±0.020)	0.577(±0.020)	0.041(±0.014)	**0.596(±0.009)**	0.588(±0.020)	**0.588(±0.020)**	0.041(±0.013)
R → I	**0.399(±0.006)**	0.407(±0.009)	0.378(±0.025)	0.066(±0.015)	0.392(±0.007)	0.407(±0.009)	**0.407(±0.009)**	**0.068(±0.013)**
R → P	0.729(±0.002)	0.726(±0.009)	0.726(±0.010)	0.041(±0.010)	**0.730(±0.003)**	0.726(±0.009)	0.726(±0.009)	0.041(±0.011)
R → Q	0.351(±0.005)	0.333(±0.013)	0.333(±0.013)	0.173(±0.014)	**0.353(±0.003)**	0.333(±0.013)	0.333(±0.013)	0.173(±0.014)
R → S	**0.588(±0.008)**	0.593(±0.011)	0.592(±0.008)	0.067(±0.006)	0.586(±0.008)	0.593(±0.011)	**0.593(±0.011)**	0.066(±0.007)
Avg.	0.531(±0.008)	0.529(±0.012)	0.521(±0.015)	0.078(±0.012)	0.531(±0.006)	0.529(±0.012)	**0.529(±0.012)**	0.078(±0.011)

Table 13. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by HoMM (Chen et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.585(±0.013)	0.590(±0.016)	0.589(±0.024)	0.035(±0.020)	0.585(±0.009)	0.590(±0.016)	**0.590(±0.016)**	0.035(±0.020)
R → I	**0.377(±0.009)**	0.372(±0.008)	0.378(±0.007)	0.066(±0.012)	0.373(±0.003)	0.372(±0.008)	0.372(±0.008)	**0.068(±0.012)**
R → P	0.722(±0.004)	0.729(±0.006)	0.725(±0.006)	0.031(±0.004)	**0.723(±0.004)**	0.729(±0.006)	**0.729(±0.006)**	**0.033(±0.003)**
R → Q	0.354(±0.008)	0.348(±0.016)	0.349(±0.014)	0.109(±0.019)	**0.362(±0.009)**	0.348(±0.016)	0.348(±0.016)	0.109(±0.020)
R → S	**0.596(±0.009)**	0.599(±0.010)	0.600(±0.009)	**0.067(±0.010)**	0.593(±0.009)	0.599(±0.010)	0.599(±0.010)	0.066(±0.010)
Avg.	0.527(±0.009)	0.528(±0.011)	0.528(±0.012)	0.062(±0.013)	0.527(±0.007)	0.528(±0.011)	0.528(±0.011)	0.062(±0.013)

Table 14. Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet for the aggregation method in objective (9) with models computed by MMDA (Rahman et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.794(±0.005)	0.604(±0.156)	0.786(±0.006)	**0.216(±0.010)**	0.794(±0.005)	**0.605(±0.159)**	**0.790(±0.007)**	0.208(±0.007)
B → E	0.760(±0.012)	0.504(±0.006)	0.756(±0.011)	0.239(±0.012)	**0.761(±0.012)**	0.504(±0.006)	**0.761(±0.012)**	**0.244(±0.011)**
B → K	0.782(±0.018)	0.491(±0.004)	0.781(±0.017)	0.217(±0.016)	0.782(±0.018)	0.491(±0.004)	0.781(±0.016)	**0.219(±0.016)**
D → B	0.794(±0.010)	**0.694(±0.177)**	0.794(±0.005)	0.204(±0.006)	**0.797(±0.009)**	0.693(±0.176)	**0.795(±0.005)**	**0.208(±0.006)**
D → E	0.783(±0.006)	0.498(±0.008)	0.776(±0.005)	0.222(±0.009)	**0.785(±0.005)**	0.498(±0.008)	**0.782(±0.005)**	0.222(±0.009)
D → K	0.789(±0.013)	0.499(±0.005)	0.789(±0.011)	0.210(±0.011)	0.789(±0.013)	0.499(±0.005)	**0.791(±0.012)**	**0.212(±0.013)**
E → B	**0.711(±0.021)**	0.572(±0.128)	0.706(±0.018)	0.289(±0.022)	0.710(±0.022)	**0.574(±0.130)**	**0.712(±0.021)**	0.289(±0.020)
E → D	0.732(±0.007)	0.504(±0.015)	0.731(±0.007)	**0.276(±0.019)**	**0.734(±0.006)**	0.504(±0.015)	**0.733(±0.008)**	0.266(±0.007)
E → K	0.875(±0.011)	0.506(±0.008)	0.874(±0.012)	**0.127(±0.010)**	0.875(±0.011)	0.506(±0.008)	0.874(±0.011)	0.126(±0.011)
K → B	0.723(±0.007)	0.493(±0.020)	0.723(±0.005)	0.277(±0.006)	**0.725(±0.004)**	0.493(±0.020)	0.723(±0.005)	**0.289(±0.027)**
K → D	0.748(±0.014)	0.498(±0.005)	**0.750(±0.014)**	0.249(±0.014)	0.748(±0.008)	0.498(±0.005)	0.749(±0.014)	0.249(±0.016)
K → E	0.858(±0.008)	**0.622(±0.208)**	0.854(±0.007)	0.144(±0.008)	0.858(±0.008)	0.620(±0.205)	**0.857(±0.008)**	0.144(±0.007)
Avg.	0.779(±0.011)	**0.541(±0.062)**	0.777(±0.010)	0.223(±0.012)	**0.780(±0.010)**	0.540(±0.062)	**0.779(±0.010)**	0.223(±0.012)

Table 15. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by AdvSKM (Liu & Xue, 2021).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.802(±0.009)	**0.606(±0.159)**	0.800(±0.011)	**0.588(±0.338)**	0.802(±0.010)	0.605(±0.158)	0.800(±0.009)	0.199(±0.009)
B → E	0.779(±0.009)	**0.596(±0.164)**	0.777(±0.007)	**0.225(±0.018)**	**0.782(±0.009)**	0.595(±0.163)	**0.780(±0.009)**	0.224(±0.011)
B → K	0.797(±0.007)	0.491(±0.004)	0.789(±0.018)	0.206(±0.012)	**0.798(±0.011)**	0.491(±0.004)	**0.795(±0.012)**	0.206(±0.011)
D → B	0.797(±0.008)	**0.693(±0.176)**	**0.796(±0.007)**	**0.210(±0.009)**	**0.798(±0.006)**	0.692(±0.176)	0.794(±0.005)	0.206(±0.005)
D → E	0.798(±0.008)	0.498(±0.008)	0.800(±0.005)	0.202(±0.011)	**0.805(±0.009)**	0.498(±0.008)	**0.802(±0.005)**	**0.203(±0.007)**
D → K	0.804(±0.015)	0.499(±0.005)	0.802(±0.016)	0.207(±0.023)	**0.805(±0.014)**	0.499(±0.005)	**0.804(±0.013)**	**0.209(±0.017)**
E → B	0.707(±0.020)	0.573(±0.128)	0.705(±0.016)	0.285(±0.024)	**0.715(±0.027)**	0.573(±0.129)	**0.712(±0.024)**	**0.286(±0.024)**
E → D	0.738(±0.011)	0.504(±0.015)	0.737(±0.013)	**0.263(±0.007)**	**0.744(±0.008)**	0.504(±0.015)	**0.741(±0.004)**	0.261(±0.008)
E → K	0.879(±0.011)	**0.750(±0.204)**	0.875(±0.013)	**0.133(±0.016)**	0.879(±0.011)	0.746(±0.200)	**0.877(±0.012)**	0.124(±0.014)
K → B	0.727(±0.014)	0.493(±0.020)	**0.735(±0.007)**	0.265(±0.009)	**0.732(±0.008)**	0.493(±0.020)	0.733(±0.010)	0.265(±0.010)
K → D	0.754(±0.026)	0.498(±0.005)	0.765(±0.009)	**0.238(±0.009)**	**0.761(±0.015)**	0.498(±0.005)	**0.767(±0.012)**	0.232(±0.010)
K → E	0.859(±0.005)	0.619(±0.207)	0.858(±0.005)	**0.146(±0.013)**	**0.861(±0.006)**	0.619(±0.208)	**0.860(±0.006)**	0.142(±0.006)
Avg.	0.787(±0.012)	0.568(±0.091)	0.787(±0.010)	**0.247(±0.041)**	**0.790(±0.011)**	0.568(±0.091)	**0.789(±0.010)**	0.213(±0.011)

Table 16. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by CDAN (Long et al., 2018).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	**0.799(±0.005)**	0.603(±0.171)	0.794(±0.009)	**0.205(±0.005)**	0.796(±0.009)	**0.606(±0.177)**	**0.796(±0.005)**	0.203(±0.005)
B → E	0.779(±0.014)	**0.584(±0.134)**	**0.761(±0.021)**	0.218(±0.010)	**0.788(±0.005)**	0.575(±0.117)	0.758(±0.050)	**0.222(±0.013)**
B → K	0.781(±0.029)	0.589(±0.171)	0.793(±0.011)	**0.205(±0.013)**	**0.800(±0.007)**	0.589(±0.170)	**0.798(±0.009)**	0.203(±0.008)
D → B	0.798(±0.012)	**0.701(±0.175)**	0.802(±0.007)	**0.197(±0.007)**	**0.803(±0.004)**	0.696(±0.171)	**0.804(±0.009)**	0.196(±0.007)
D → E	0.795(±0.009)	0.498(±0.008)	0.802(±0.010)	0.198(±0.006)	**0.811(±0.007)**	0.498(±0.008)	**0.805(±0.007)**	**0.288(±0.156)**
D → K	0.799(±0.018)	0.499(±0.005)	**0.804(±0.015)**	0.196(±0.010)	**0.810(±0.012)**	0.499(±0.005)	0.803(±0.010)	**0.202(±0.015)**
E → B	**0.719(±0.014)**	**0.575(±0.113)**	0.721(±0.017)	0.279(±0.018)	0.718(±0.014)	0.568(±0.100)	**0.722(±0.015)**	**0.290(±0.014)**
E → D	**0.750(±0.009)**	0.504(±0.015)	0.744(±0.008)	**0.284(±0.042)**	0.739(±0.011)	0.504(±0.015)	0.744(±0.011)	0.263(±0.017)
E → K	0.875(±0.015)	0.628(±0.218)	0.875(±0.010)	**0.133(±0.014)**	**0.880(±0.011)**	**0.629(±0.220)**	0.875(±0.009)	0.124(±0.012)
K → B	0.745(±0.004)	0.493(±0.020)	0.731(±0.004)	0.273(±0.013)	**0.747(±0.002)**	0.493(±0.020)	**0.737(±0.008)**	**0.297(±0.049)**
K → D	**0.775(±0.019)**	0.498(±0.005)	0.740(±0.031)	**0.259(±0.028)**	0.762(±0.016)	0.498(±0.005)	**0.763(±0.021)**	0.246(±0.029)
K → E	0.864(±0.010)	**0.748(±0.211)**	0.867(±0.006)	0.136(±0.012)	**0.869(±0.006)**	0.735(±0.199)	0.867(±0.004)	**0.151(±0.039)**
Avg.	0.790(±0.013)	**0.577(±0.104)**	0.786(±0.012)	0.215(±0.015)	**0.794(±0.009)**	0.574(±0.101)	**0.789(±0.013)**	**0.224(±0.030)**

Table 17. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by CMD (Zellinger et al., 2017).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.804(±0.008)	0.609(±0.165)	0.806(±0.007)	**0.205(±0.016)**	0.804(±0.003)	0.609(±0.165)	0.806(±0.006)	0.195(±0.007)
B → E	0.785(±0.012)	0.504(±0.006)	0.782(±0.010)	**0.239(±0.032)**	**0.791(±0.005)**	0.504(±0.006)	**0.787(±0.006)**	0.213(±0.008)
B → K	**0.807(±0.009)**	0.491(±0.004)	**0.807(±0.007)**	0.196(±0.008)	0.806(±0.012)	0.491(±0.004)	0.806(±0.008)	**0.197(±0.008)**
D → B	0.801(±0.005)	0.594(±0.175)	0.802(±0.005)	0.197(±0.007)	**0.805(±0.005)**	**0.595(±0.177)**	**0.805(±0.006)**	**0.198(±0.006)**
D → E	0.806(±0.011)	0.498(±0.008)	0.804(±0.015)	0.191(±0.009)	**0.811(±0.008)**	0.498(±0.008)	**0.809(±0.009)**	**0.206(±0.029)**
D → K	0.819(±0.012)	0.499(±0.005)	0.819(±0.016)	0.179(±0.011)	**0.822(±0.009)**	0.499(±0.005)	**0.822(±0.011)**	0.179(±0.012)
E → B	0.719(±0.025)	**0.721(±0.029)**	**0.721(±0.031)**	**0.295(±0.025)**	**0.722(±0.026)**	0.719(±0.034)	0.719(±0.028)	0.281(±0.026)
E → D	0.748(±0.024)	0.588(±0.155)	0.743(±0.017)	**0.253(±0.020)**	**0.751(±0.015)**	0.588(±0.155)	**0.749(±0.022)**	0.251(±0.020)
E → K	0.883(±0.012)	0.506(±0.008)	0.881(±0.009)	**0.123(±0.011)**	0.883(±0.012)	0.506(±0.008)	0.881(±0.011)	0.120(±0.010)
K → B	0.741(±0.008)	0.493(±0.020)	0.734(±0.008)	**0.261(±0.004)**	**0.745(±0.004)**	0.493(±0.020)	**0.742(±0.005)**	0.256(±0.004)
K → D	0.766(±0.007)	0.498(±0.005)	0.762(±0.007)	0.236(±0.008)	**0.768(±0.009)**	0.498(±0.005)	**0.764(±0.006)**	**0.258(±0.042)**
K → E	0.866(±0.008)	0.625(±0.214)	0.865(±0.007)	0.133(±0.005)	**0.867(±0.008)**	**0.741(±0.205)**	**0.866(±0.006)**	**0.139(±0.012)**
Avg.	0.795(±0.012)	0.552(±0.066)	0.794(±0.012)	**0.209(±0.013)**	**0.798(±0.010)**	**0.562(±0.066)**	**0.796(±0.010)**	0.208(±0.015)

Table 18. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by CoDATS (Wilson et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.802(±0.006)	**0.604(±0.173)**	0.802(±0.010)	**0.198(±0.009)**	**0.804(±0.011)**	0.603(±0.172)	0.802(±0.009)	0.197(±0.005)
B → E	0.784(±0.012)	**0.594(±0.160)**	0.763(±0.013)	0.214(±0.010)	**0.791(±0.005)**	0.586(±0.147)	**0.782(±0.017)**	**0.216(±0.014)**
B → K	0.801(±0.009)	0.491(±0.004)	0.803(±0.008)	0.203(±0.016)	**0.807(±0.008)**	0.491(±0.004)	**0.805(±0.008)**	**0.378(±0.313)**
D → B	0.804(±0.003)	**0.596(±0.178)**	0.799(±0.003)	**0.202(±0.007)**	**0.806(±0.002)**	0.592(±0.171)	**0.805(±0.006)**	0.198(±0.004)
D → E	0.808(±0.007)	0.498(±0.008)	0.811(±0.009)	0.187(±0.006)	**0.814(±0.006)**	0.498(±0.008)	**0.813(±0.006)**	0.187(±0.006)
D → K	0.814(±0.013)	0.499(±0.005)	0.814(±0.018)	0.189(±0.007)	**0.818(±0.010)**	0.499(±0.005)	**0.820(±0.009)**	**0.362(±0.319)**
E → B	0.713(±0.017)	0.502(±0.014)	0.717(±0.014)	**0.285(±0.015)**	**0.727(±0.014)**	0.502(±0.014)	**0.720(±0.013)**	0.279(±0.014)
E → D	0.748(±0.009)	0.504(±0.015)	0.748(±0.006)	**0.253(±0.008)**	**0.763(±0.005)**	0.504(±0.015)	**0.751(±0.005)**	0.248(±0.006)
E → K	0.878(±0.015)	**0.876(±0.015)**	0.874(±0.015)	**0.129(±0.015)**	**0.880(±0.013)**	0.873(±0.016)	0.874(±0.021)	0.124(±0.014)
K → B	0.739(±0.014)	0.493(±0.020)	0.750(±0.003)	0.249(±0.005)	**0.748(±0.007)**	0.493(±0.020)	**0.751(±0.006)**	0.249(±0.008)
K → D	0.770(±0.020)	0.498(±0.005)	0.766(±0.039)	**0.228(±0.028)**	**0.781(±0.018)**	0.498(±0.005)	**0.782(±0.013)**	0.218(±0.013)
K → E	0.863(±0.011)	0.501(±0.003)	0.865(±0.007)	**0.148(±0.023)**	**0.867(±0.010)**	0.501(±0.003)	**0.866(±0.008)**	0.134(±0.009)
Avg.	0.794(±0.011)	**0.555(±0.050)**	0.793(±0.012)	0.207(±0.012)	**0.800(±0.009)**	0.553(±0.048)	**0.798(±0.010)**	**0.232(±0.060)**

Table 19. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by DANN (Ganin et al., 2016).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → E	0.763(±0.014)	0.584(±0.133)	0.760(±0.016)	0.237(±0.015)	0.763(±0.014)	**0.585(±0.134)**	0.760(±0.021)	**0.239(±0.018)**
B → K	0.783(±0.014)	**0.579(±0.153)**	**0.779(±0.022)**	0.217(±0.016)	**0.784(±0.014)**	0.578(±0.152)	0.778(±0.023)	0.217(±0.015)
D → B	0.795(±0.006)	**0.596(±0.176)**	0.791(±0.008)	**0.226(±0.022)**	0.795(±0.005)	0.595(±0.173)	**0.792(±0.008)**	0.208(±0.008)
D → E	0.784(±0.004)	0.498(±0.008)	0.778(±0.008)	**0.223(±0.016)**	**0.785(±0.002)**	0.498(±0.008)	**0.784(±0.002)**	0.219(±0.006)
D → K	0.789(±0.021)	0.499(±0.005)	0.790(±0.015)	0.207(±0.016)	**0.793(±0.016)**	0.499(±0.005)	**0.793(±0.017)**	**0.208(±0.015)**
E → B	0.705(±0.024)	0.502(±0.014)	0.699(±0.024)	0.299(±0.020)	0.705(±0.026)	**0.568(±0.100)**	**0.703(±0.024)**	0.299(±0.023)
E → D	**0.736(±0.007)**	0.504(±0.015)	**0.740(±0.007)**	**0.270(±0.017)**	0.735(±0.007)	0.504(±0.015)	0.736(±0.009)	0.263(±0.010)
E → K	0.877(±0.011)	**0.628(±0.219)**	0.874(±0.009)	**0.137(±0.028)**	0.877(±0.011)	0.626(±0.216)	0.874(±0.009)	0.129(±0.008)
K → B	**0.726(±0.003)**	0.493(±0.020)	0.722(±0.007)	0.276(±0.004)	0.725(±0.004)	0.493(±0.020)	**0.724(±0.005)**	0.276(±0.004)
K → D	0.751(±0.013)	0.498(±0.005)	0.752(±0.009)	0.248(±0.010)	**0.755(±0.011)**	0.498(±0.005)	**0.753(±0.012)**	0.248(±0.011)
K → E	0.856(±0.008)	**0.739(±0.203)**	0.856(±0.007)	**0.154(±0.008)**	0.856(±0.008)	0.730(±0.196)	**0.857(±0.007)**	0.144(±0.007)
Avg.	0.780(±0.011)	0.568(±0.093)	0.778(±0.012)	**0.226(±0.015)**	0.780(±0.011)	**0.572(±0.099)**	**0.779(±0.012)**	0.222(±0.011)

Table 20. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by DDC (Tzeng et al., 2014).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.801(±0.010)	**0.702(±0.175)**	0.799(±0.011)	**0.203(±0.008)**	0.801(±0.010)	0.701(±0.174)	0.799(±0.009)	0.202(±0.009)
B → E	0.773(±0.008)	**0.587(±0.137)**	0.763(±0.012)	0.234(±0.013)	0.773(±0.008)	0.586(±0.136)	0.763(±0.019)	**0.236(±0.017)**
B → K	0.785(±0.017)	**0.580(±0.155)**	0.781(±0.018)	**0.218(±0.016)**	0.785(±0.016)	0.579(±0.153)	0.781(±0.020)	0.216(±0.016)
D → B	0.797(±0.004)	**0.594(±0.172)**	0.794(±0.003)	**0.226(±0.024)**	**0.798(±0.003)**	0.591(±0.167)	**0.796(±0.003)**	0.206(±0.003)
D → E	0.785(±0.003)	0.498(±0.008)	0.779(±0.011)	**0.227(±0.024)**	**0.786(±0.005)**	0.498(±0.008)	**0.784(±0.004)**	0.217(±0.004)
D → K	0.792(±0.014)	0.499(±0.005)	0.791(±0.016)	0.206(±0.014)	**0.796(±0.014)**	0.499(±0.005)	**0.796(±0.013)**	0.206(±0.013)
E → B	0.710(±0.024)	0.502(±0.014)	0.709(±0.024)	0.293(±0.022)	0.710(±0.023)	**0.569(±0.101)**	**0.710(±0.022)**	0.293(±0.022)
E → D	0.739(±0.004)	0.504(±0.015)	0.739(±0.003)	**0.272(±0.020)**	0.739(±0.005)	0.504(±0.015)	**0.740(±0.003)**	0.261(±0.005)
E → K	0.878(±0.007)	0.629(±0.221)	0.876(±0.009)	**0.135(±0.027)**	0.878(±0.007)	0.629(±0.220)	0.876(±0.007)	0.127(±0.008)
K → B	**0.739(±0.009)**	0.493(±0.020)	0.735(±0.008)	**0.266(±0.006)**	0.738(±0.010)	0.493(±0.020)	**0.738(±0.009)**	0.261(±0.007)
K → D	**0.755(±0.015)**	0.498(±0.005)	0.749(±0.017)	**0.244(±0.010)**	0.749(±0.012)	0.498(±0.005)	**0.756(±0.012)**	0.243(±0.011)
K → E	0.857(±0.010)	**0.741(±0.205)**	0.855(±0.005)	**0.148(±0.008)**	0.857(±0.010)	0.733(±0.199)	**0.857(±0.009)**	0.143(±0.009)
Avg.	0.784(±0.010)	0.569(±0.094)	0.781(±0.011)	**0.223(±0.016)**	0.784(±0.010)	**0.573(±0.100)**	**0.783(±0.011)**	0.217(±0.010)

Table 21. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by Deep Coral (Sun et al., 2017).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	**0.812**(±**0.004**)	0.509(±0.010)	**0.811**(±**0.005**)	**0.195**(±**0.004**)	0.810(±0.004)	0.509(±0.010)	0.808(±0.006)	0.193(±0.006)
B → E	**0.786**(±**0.007**)	**0.775**(±**0.009**)	0.779(±0.008)	0.224(±0.004)	0.779(±0.006)	0.768(±0.009)	0.779(±0.007)	**0.396**(±**0.274**)
B → K	**0.809**(±**0.006**)	0.491(±0.004)	0.797(±0.007)	0.198(±0.008)	0.800(±0.006)	0.491(±0.004)	**0.805**(±**0.006**)	**0.205**(±**0.021**)
D → B	0.806(±0.009)	**0.707**(±**0.183**)	0.811(±0.009)	**0.196**(±**0.012**)	**0.817**(±**0.011**)	0.696(±0.173)	**0.813**(±**0.012**)	0.191(±0.011)
D → E	**0.804**(±**0.012**)	0.498(±0.008)	0.802(±0.022)	0.199(±0.011)	0.800(±0.005)	0.498(±0.008)	**0.805**(±**0.008**)	**0.209**(±**0.008**)
D → K	0.830(±0.012)	0.499(±0.005)	0.823(±0.018)	0.173(±0.007)	0.830(±0.010)	0.499(±0.005)	**0.830**(±**0.008**)	**0.174**(±**0.011**)
E → B	0.686(±0.016)	0.502(±0.014)	0.685(±0.019)	0.306(±0.023)	**0.705**(±**0.019**)	0.502(±0.014)	**0.697**(±**0.019**)	**0.307**(±**0.018**)
E → D	0.711(±0.017)	0.504(±0.015)	0.715(±0.016)	**0.286**(±**0.016**)	**0.726**(±**0.012**)	0.504(±0.015)	**0.717**(±**0.017**)	0.282(±0.019)
E → K	0.885(±0.009)	**0.756**(±**0.209**)	0.880(±0.008)	0.121(±0.009)	0.885(±0.009)	0.744(±0.199)	**0.886**(±**0.008**)	**0.347**(±**0.412**)
K → B	0.720(±0.011)	0.493(±0.020)	0.731(±0.011)	0.269(±0.010)	**0.741**(±**0.004**)	0.493(±0.020)	**0.733**(±**0.008**)	0.269(±0.007)
K → D	0.729(±0.013)	0.498(±0.005)	0.748(±0.010)	**0.252**(±**0.007**)	**0.767**(±**0.006**)	0.498(±0.005)	**0.753**(±**0.005**)	0.247(±0.005)
K → E	0.863(±0.013)	0.501(±0.003)	0.870(±0.009)	**0.131**(±**0.007**)	**0.868**(±**0.010**)	0.501(±0.003)	0.870(±0.007)	0.130(±0.008)
Avg.	0.787(±0.011)	**0.561**(±**0.041**)	0.788(±0.012)	0.212(±0.010)	**0.794**(±**0.008**)	0.559(±0.039)	**0.791**(±**0.009**)	**0.246**(±**0.067**)

Table 22. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by DIRT (Shu et al., 2018).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.803(±0.015)	**0.707**(±**0.179**)	0.803(±0.011)	**0.206**(±**0.024**)	**0.808**(±**0.011**)	0.704(±0.176)	**0.806**(±**0.009**)	0.193(±0.010)
B → E	0.788(±0.004)	0.589(±0.142)	**0.786**(±**0.016**)	0.206(±0.006)	**0.799**(±**0.004**)	**0.592**(±**0.147**)	0.779(±0.030)	**0.207**(±**0.006**)
B → K	0.804(±0.012)	**0.593**(±**0.178**)	0.807(±0.015)	0.193(±0.009)	**0.809**(±**0.010**)	0.588(±0.169)	0.807(±0.010)	0.193(±0.009)
D → B	0.801(±0.005)	**0.596**(±**0.176**)	**0.802**(±**0.005**)	**0.202**(±**0.013**)	**0.802**(±**0.004**)	0.595(±0.175)	0.799(±0.006)	0.199(±0.007)
D → E	0.803(±0.008)	0.498(±0.008)	0.801(±0.009)	0.192(±0.006)	**0.813**(±**0.009**)	0.498(±0.008)	**0.808**(±**0.004**)	**0.196**(±**0.015**)
D → K	0.803(±0.029)	0.499(±0.005)	0.805(±0.017)	0.184(±0.012)	**0.819**(±**0.010**)	0.499(±0.005)	**0.816**(±**0.013**)	**0.185**(±**0.012**)
E → B	0.719(±0.021)	0.502(±0.014)	0.719(±0.022)	0.279(±0.019)	**0.723**(±**0.020**)	**0.572**(±**0.107**)	0.719(±0.019)	**0.280**(±**0.018**)
E → D	0.749(±0.004)	0.504(±0.015)	0.746(±0.009)	**0.264**(±**0.017**)	0.749(±0.011)	0.504(±0.015)	**0.747**(±**0.010**)	0.263(±0.015)
E → K	0.883(±0.010)	0.630(±0.223)	0.881(±0.008)	**0.127**(±**0.028**)	**0.886**(±**0.010**)	0.630(±0.223)	0.881(±0.011)	0.118(±0.010)
K → B	0.739(±0.010)	0.493(±0.020)	0.732(±0.018)	**0.258**(±**0.006**)	**0.743**(±**0.010**)	0.493(±0.020)	**0.745**(±**0.006**)	0.256(±0.006)
K → D	0.767(±0.008)	0.498(±0.005)	0.770(±0.015)	**0.241**(±**0.027**)	**0.778**(±**0.005**)	0.498(±0.005)	**0.775**(±**0.010**)	0.226(±0.009)
K → E	0.864(±0.006)	**0.625**(±**0.213**)	0.863(±0.007)	**0.139**(±**0.009**)	**0.868**(±**0.008**)	0.617(±0.199)	**0.867**(±**0.003**)	0.133(±0.006)
Avg.	0.794(±0.011)	0.561(±0.098)	0.793(±0.013)	**0.207**(±**0.015**)	**0.800**(±**0.010**)	**0.566**(±**0.104**)	**0.796**(±**0.011**)	0.204(±0.010)

Table 23. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by DSAN (Zhu et al., 2021).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.794(±0.007)	**0.700**(±**0.172**)	0.792(±0.010)	**0.217**(±**0.016**)	0.794(±0.007)	0.698(±0.171)	**0.793**(±**0.007**)	0.208(±0.006)
B → E	0.761(±0.012)	0.583(±0.131)	0.756(±0.018)	0.239(±0.011)	0.761(±0.012)	0.583(±0.131)	0.756(±0.020)	**0.243**(±**0.018**)
B → K	**0.781**(±**0.017**)	**0.580**(±**0.155**)	0.776(±0.022)	**0.222**(±**0.015**)	0.780(±0.019)	0.576(±0.149)	0.776(±0.022)	0.220(±0.013)
D → B	0.790(±0.002)	**0.594**(±**0.172**)	0.789(±0.003)	**0.217**(±**0.015**)	**0.791**(±**0.004**)	0.593(±0.171)	0.789(±0.004)	0.210(±0.004)
D → E	0.781(±0.004)	0.498(±0.008)	0.772(±0.004)	0.221(±0.003)	0.781(±0.003)	0.498(±0.008)	**0.779**(±**0.003**)	**0.224**(±**0.006**)
D → K	0.789(±0.015)	0.499(±0.005)	0.788(±0.015)	**0.212**(±**0.014**)	**0.790**(±**0.016**)	0.499(±0.005)	0.788(±0.017)	0.211(±0.015)
E → B	**0.699**(±**0.012**)	0.502(±0.014)	0.693(±0.014)	**0.304**(±**0.017**)	0.698(±0.011)	**0.566**(±**0.097**)	**0.698**(±**0.015**)	0.302(±0.015)
E → D	**0.735**(±**0.011**)	0.504(±0.015)	0.732(±0.010)	**0.275**(±**0.020**)	0.734(±0.009)	0.504(±0.015)	**0.734**(±**0.008**)	0.270(±0.008)
E → K	0.873(±0.009)	0.626(±0.216)	**0.872**(±**0.008**)	**0.136**(±**0.023**)	0.873(±0.009)	0.626(±0.216)	0.871(±0.008)	0.129(±0.007)
K → B	0.722(±0.006)	0.493(±0.020)	0.717(±0.013)	**0.279**(±**0.006**)	0.722(±0.007)	0.493(±0.020)	**0.722**(±**0.007**)	0.276(±0.007)
K → D	0.748(±0.008)	0.498(±0.005)	**0.752**(±**0.009**)	**0.257**(±**0.014**)	**0.749**(±**0.009**)	0.498(±0.005)	0.749(±0.004)	0.252(±0.007)
K → E	0.856(±0.006)	**0.622**(±**0.209**)	0.854(±0.007)	**0.147**(±**0.007**)	0.856(±0.006)	0.613(±0.193)	**0.855**(±**0.007**)	0.145(±0.006)
Avg.	0.777(±0.009)	0.558(±0.094)	0.774(±0.011)	**0.227**(±**0.013**)	0.777(±0.009)	**0.562**(±**0.098**)	**0.776**(±**0.010**)	0.224(±0.009)

Table 24. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by HoMM (Chen et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.795(±0.003)	**0.699(±0.172)**	0.796(±0.003)	**0.206(±0.007)**	**0.797(±0.003)**	0.695(±0.168)	**0.798(±0.003)**	0.203(±0.004)
B → E	**0.776(±0.018)**	**0.590(±0.143)**	0.766(±0.016)	**0.228(±0.020)**	0.773(±0.022)	0.580(±0.126)	**0.772(±0.020)**	0.227(±0.015)
B → K	0.793(±0.019)	0.587(±0.167)	0.794(±0.021)	**0.209(±0.024)**	**0.800(±0.013)**	0.587(±0.168)	**0.798(±0.013)**	0.203(±0.012)
D → B	**0.799(±0.007)**	**0.696(±0.170)**	0.793(±0.006)	**0.205(±0.013)**	0.798(±0.004)	0.694(±0.168)	**0.796(±0.006)**	0.203(±0.007)
D → E	0.791(±0.008)	0.498(±0.008)	0.790(±0.011)	0.208(±0.006)	**0.793(±0.005)**	0.498(±0.008)	**0.792(±0.005)**	**0.224(±0.029)**
D → K	0.802(±0.010)	0.499(±0.005)	0.801(±0.009)	0.194(±0.009)	**0.808(±0.002)**	0.499(±0.005)	**0.806(±0.008)**	**0.199(±0.010)**
E → B	0.709(±0.015)	0.573(±0.109)	0.709(±0.011)	0.286(±0.015)	**0.711(±0.015)**	**0.575(±0.112)**	**0.714(±0.015)**	**0.296(±0.014)**
E → D	**0.746(±0.003)**	0.504(±0.015)	0.737(±0.009)	**0.265(±0.016)**	0.740(±0.001)	0.504(±0.015)	**0.741(±0.002)**	0.263(±0.007)
E → K	0.878(±0.009)	**0.628(±0.220)**	0.876(±0.008)	**0.127(±0.011)**	0.878(±0.009)	0.627(±0.218)	**0.878(±0.006)**	0.121(±0.008)
K → B	0.729(±0.005)	0.493(±0.020)	0.732(±0.006)	0.268(±0.007)	**0.739(±0.002)**	0.493(±0.020)	**0.738(±0.007)**	**0.270(±0.012)**
K → D	0.757(±0.015)	0.498(±0.005)	0.756(±0.012)	**0.247(±0.014)**	**0.768(±0.012)**	0.498(±0.005)	**0.764(±0.011)**	0.239(±0.008)
K → E	0.857(±0.008)	**0.741(±0.205)**	**0.858(±0.007)**	0.143(±0.009)	**0.859(±0.006)**	0.724(±0.190)	0.857(±0.005)	**0.146(±0.012)**
Avg.	0.786(±0.010)	**0.584(±0.103)**	0.784(±0.010)	0.216(±0.013)	**0.789(±0.008)**	0.581(±0.100)	**0.788(±0.008)**	0.216(±0.011)

Table 25. Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews for the aggregation method in objective (9) with models computed by MMDA (Rahman et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
0 → 6	0.718(±0.009)	**0.714(±0.010)**	0.705(±0.020)	0.012(±0.019)	0.718(±0.009)	0.705(±0.020)	0.705(±0.020)	0.012(±0.019)
1 → 6	0.857(±0.006)	0.835(±0.026)	0.836(±0.025)	0.000(±0.001)	0.857(±0.006)	**0.836(±0.025)**	0.836(±0.025)	0.000(±0.001)
2 → 7	0.490(±0.014)	0.497(±0.010)	0.502(±0.024)	0.004(±0.004)	0.490(±0.014)	**0.502(±0.024)**	0.502(±0.024)	0.004(±0.004)
3 → 8	0.810(±0.002)	0.806(±0.005)	0.807(±0.005)	0.003(±0.004)	0.810(±0.002)	**0.807(±0.005)**	0.807(±0.005)	0.003(±0.004)
4 → 5	0.884(±0.005)	0.875(±0.014)	0.878(±0.017)	0.000(±0.001)	0.884(±0.005)	**0.878(±0.017)**	0.878(±0.017)	0.000(±0.001)
Avg.	0.752(±0.007)	0.746(±0.013)	0.746(±0.018)	0.004(±0.006)	0.752(±0.007)	0.746(±0.018)	0.746(±0.018)	0.004(±0.006)

Table 26. Mean and standard deviation (after ±) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by AdvSKM (Liu & Xue, 2021).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
0 → 6	0.622(±0.009)	0.507(±0.281)	0.666(±0.014)	0.014(±0.004)	0.622(±0.009)	**0.666(±0.014)**	0.666(±0.014)	0.014(±0.004)
1 → 6	0.933(±0.000)	0.906(±0.008)	0.908(±0.006)	0.014(±0.004)	0.933(±0.000)	**0.908(±0.006)**	0.908(±0.006)	0.014(±0.004)
2 → 7	**0.550(±0.061)**	0.547(±0.061)	0.552(±0.066)	0.029(±0.024)	0.548(±0.063)	**0.552(±0.066)**	0.552(±0.066)	0.029(±0.024)
3 → 8	0.874(±0.076)	**0.863(±0.059)**	0.862(±0.056)	0.000(±0.000)	0.874(±0.076)	0.862(±0.055)	0.862(±0.056)	0.000(±0.000)
4 → 5	0.980(±0.000)	0.707(±0.423)	0.954(±0.006)	0.000(±0.000)	0.980(±0.000)	**0.708(±0.424)**	0.954(±0.006)	0.000(±0.000)
Avg.	**0.792(±0.029)**	0.706(±0.166)	0.788(±0.029)	0.012(±0.007)	0.791(±0.030)	**0.739(±0.113)**	0.788(±0.029)	0.012(±0.007)

Table 27. Mean and standard deviation (after ±) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by CDAN (Long et al., 2018).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
0 → 6	0.693(±0.012)	**0.713(±0.012)**	0.693(±0.010)	0.068(±0.018)	0.693(±0.012)	0.693(±0.010)	0.693(±0.010)	0.068(±0.018)
1 → 6	0.907(±0.006)	0.885(±0.018)	0.896(±0.008)	0.005(±0.001)	0.907(±0.006)	**0.896(±0.008)**	0.896(±0.008)	0.005(±0.001)
2 → 7	0.320(±0.277)	0.503(±0.027)	0.513(±0.040)	0.001(±0.002)	**0.482(±0.008)**	**0.513(±0.040)**	0.513(±0.040)	0.001(±0.002)
3 → 8	0.811(±0.011)	0.813(±0.009)	0.813(±0.011)	0.000(±0.000)	0.811(±0.011)	**0.814(±0.011)**	**0.814(±0.011)**	0.000(±0.000)
4 → 5	0.625(±0.541)	**0.937(±0.011)**	0.934(±0.013)	0.000(±0.001)	**0.939(±0.002)**	0.934(±0.013)	0.934(±0.013)	0.000(±0.001)
Avg.	0.671(±0.170)	0.770(±0.015)	0.770(±0.016)	0.015(±0.004)	**0.766(±0.008)**	0.770(±0.016)	0.770(±0.016)	0.015(±0.004)

Table 28. Mean and standard deviation (after ±) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by CMD (Zellinger et al., 2017).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$0.596(\pm 0.048)$	$\mathbf{0.634(\pm 0.045)}$	$0.631(\pm 0.043)$	$0.003(\pm 0.005)$	$\mathbf{0.601(\pm 0.029)}$	$0.631(\pm 0.043)$	$0.631(\pm 0.043)$	$0.003(\pm 0.005)$
$1 \to 6$	$0.939(\pm 0.006)$	$\mathbf{0.908(\pm 0.011)}$	$0.906(\pm 0.013)$	$0.001(\pm 0.001)$	$0.939(\pm 0.006)$	$0.906(\pm 0.013)$	$0.906(\pm 0.013)$	$0.001(\pm 0.001)$
$2 \to 7$	$0.472(\pm 0.005)$	$0.470(\pm 0.008)$	$0.476(\pm 0.008)$	$0.005(\pm 0.008)$	$0.472(\pm 0.005)$	$\mathbf{0.476(\pm 0.008)}$	$0.476(\pm 0.008)$	$0.005(\pm 0.008)$
$3 \to 8$	$0.960(\pm 0.030)$	$0.921(\pm 0.028)$	$0.934(\pm 0.030)$	$0.000(\pm 0.000)$	$0.960(\pm 0.030)$	$\mathbf{0.936(\pm 0.032)}$	$0.934(\pm 0.030)$	$0.000(\pm 0.000)$
$4 \to 5$	$0.648(\pm 0.562)$	$0.947(\pm 0.013)$	$0.947(\pm 0.012)$	$0.000(\pm 0.000)$	$\mathbf{0.973(\pm 0.004)}$	$0.947(\pm 0.012)$	$0.947(\pm 0.012)$	$0.000(\pm 0.000)$
Avg.	$0.723(\pm 0.130)$	$0.776(\pm 0.021)$	$0.779(\pm 0.021)$	$0.002(\pm 0.003)$	$\mathbf{0.789(\pm 0.015)}$	$\mathbf{0.779(\pm 0.022)}$	$0.779(\pm 0.021)$	$0.002(\pm 0.003)$

Table 29. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by CoDATS (Wilson et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$0.604(\pm 0.036)$	$0.652(\pm 0.043)$	$0.657(\pm 0.027)$	$0.000(\pm 0.000)$	$\mathbf{0.606(\pm 0.039)}$	$\mathbf{0.657(\pm 0.027)}$	$0.657(\pm 0.027)$	$0.000(\pm 0.000)$
$1 \to 6$	$0.936(\pm 0.002)$	$0.890(\pm 0.038)$	$0.906(\pm 0.010)$	$0.003(\pm 0.005)$	$0.936(\pm 0.002)$	$\mathbf{0.906(\pm 0.010)}$	$0.906(\pm 0.010)$	$0.003(\pm 0.005)$
$2 \to 7$	$0.327(\pm 0.284)$	$0.511(\pm 0.026)$	$0.533(\pm 0.045)$	$0.000(\pm 0.000)$	$\mathbf{0.493(\pm 0.004)}$	$\mathbf{0.533(\pm 0.045)}$	$0.533(\pm 0.045)$	$0.000(\pm 0.000)$
$3 \to 8$	$0.964(\pm 0.006)$	$0.919(\pm 0.007)$	$0.923(\pm 0.015)$	$0.000(\pm 0.000)$	$0.964(\pm 0.006)$	$\mathbf{0.923(\pm 0.015)}$	$0.923(\pm 0.015)$	$0.000(\pm 0.000)$
$4 \to 5$	$0.654(\pm 0.566)$	$0.951(\pm 0.005)$	$0.957(\pm 0.010)$	$0.002(\pm 0.004)$	$\mathbf{0.980(\pm 0.000)}$	$\mathbf{0.957(\pm 0.010)}$	$0.957(\pm 0.010)$	$0.002(\pm 0.004)$
Avg.	$0.697(\pm 0.179)$	$0.785(\pm 0.024)$	$0.795(\pm 0.021)$	$0.001(\pm 0.002)$	$\mathbf{0.796(\pm 0.010)}$	$\mathbf{0.795(\pm 0.021)}$	$0.795(\pm 0.021)$	$0.001(\pm 0.002)$

Table 30. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by DANN (Ganin et al., 2016)

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$0.697(\pm 0.013)$	$0.349(\pm 0.286)$	$0.690(\pm 0.011)$	$0.072(\pm 0.046)$	$0.697(\pm 0.013)$	$0.349(\pm 0.287)$	$0.690(\pm 0.011)$	$0.072(\pm 0.046)$
$1 \to 6$	$0.882(\pm 0.017)$	$0.639(\pm 0.413)$	$0.873(\pm 0.014)$	$0.001(\pm 0.002)$	$0.882(\pm 0.017)$	$0.639(\pm 0.413)$	$0.873(\pm 0.014)$	$0.001(\pm 0.002)$
$2 \to 7$	$0.439(\pm 0.041)$	$0.440(\pm 0.036)$	$0.439(\pm 0.037)$	$0.005(\pm 0.007)$	$0.439(\pm 0.041)$	$0.440(\pm 0.038)$	$0.439(\pm 0.037)$	$0.005(\pm 0.007)$
$3 \to 8$	$0.818(\pm 0.008)$	$0.621(\pm 0.342)$	$0.823(\pm 0.006)$	$0.001(\pm 0.002)$	$0.818(\pm 0.008)$	$\mathbf{0.622(\pm 0.343)}$	$0.823(\pm 0.006)$	$0.001(\pm 0.002)$
$4 \to 5$	$0.905(\pm 0.008)$	$0.219(\pm 0.000)$	$0.895(\pm 0.025)$	$0.001(\pm 0.002)$	$0.905(\pm 0.008)$	$0.219(\pm 0.000)$	$0.895(\pm 0.025)$	$0.001(\pm 0.002)$
Avg.	$0.748(\pm 0.017)$	$0.454(\pm 0.215)$	$0.744(\pm 0.019)$	$0.016(\pm 0.012)$	$0.748(\pm 0.017)$	$0.454(\pm 0.216)$	$0.744(\pm 0.019)$	$0.016(\pm 0.012)$

Table 31. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by DDC (Tzeng et al., 2014).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$0.728(\pm 0.019)$	$0.700(\pm 0.032)$	$0.717(\pm 0.025)$	$0.006(\pm 0.009)$	$0.728(\pm 0.019)$	$\mathbf{0.717(\pm 0.025)}$	$0.717(\pm 0.025)$	$0.006(\pm 0.009)$
$1 \to 6$	$0.886(\pm 0.009)$	$0.845(\pm 0.039)$	$0.861(\pm 0.014)$	$0.008(\pm 0.003)$	$0.886(\pm 0.009)$	$\mathbf{0.861(\pm 0.014)}$	$0.861(\pm 0.014)$	$0.008(\pm 0.003)$
$2 \to 7$	$0.460(\pm 0.012)$	$0.470(\pm 0.014)$	$0.480(\pm 0.020)$	$0.004(\pm 0.003)$	$0.460(\pm 0.012)$	$\mathbf{0.480(\pm 0.021)}$	$0.480(\pm 0.020)$	$0.004(\pm 0.003)$
$3 \to 8$	$0.812(\pm 0.000)$	$0.810(\pm 0.003)$	$0.811(\pm 0.001)$	$0.008(\pm 0.007)$	$0.812(\pm 0.000)$	$\mathbf{0.811(\pm 0.001)}$	$0.811(\pm 0.001)$	$0.008(\pm 0.007)$
$4 \to 5$	$0.618(\pm 0.536)$	$0.919(\pm 0.020)$	$0.923(\pm 0.016)$	$0.003(\pm 0.003)$	$\mathbf{0.936(\pm 0.016)}$	$\mathbf{0.923(\pm 0.016)}$	$0.923(\pm 0.016)$	$0.003(\pm 0.003)$
Avg.	$0.701(\pm 0.115)$	$0.749(\pm 0.021)$	$0.758(\pm 0.015)$	$0.006(\pm 0.005)$	$\mathbf{0.764(\pm 0.011)}$	$\mathbf{0.758(\pm 0.016)}$	$0.758(\pm 0.015)$	$0.006(\pm 0.005)$

Table 32. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by DeepCoral (Sun et al., 2017).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$\mathbf{0.693(\pm 0.072)}$	$0.694(\pm 0.050)$	$0.709(\pm 0.043)$	$0.006(\pm 0.010)$	$0.688(\pm 0.052)$	$\mathbf{0.709(\pm 0.043)}$	$0.709(\pm 0.043)$	$0.006(\pm 0.010)$
$1 \to 6$	$0.938(\pm 0.000)$	$0.884(\pm 0.029)$	$0.904(\pm 0.010)$	$0.002(\pm 0.003)$	$0.938(\pm 0.000)$	$\mathbf{0.904(\pm 0.010)}$	$0.904(\pm 0.010)$	$0.002(\pm 0.003)$
$2 \to 7$	$0.194(\pm 0.336)$	$\mathbf{0.582(\pm 0.090)}$	$0.545(\pm 0.041)$	$0.002(\pm 0.003)$	$\mathbf{0.574(\pm 0.057)}$	$0.557(\pm 0.062)$	$\mathbf{0.546(\pm 0.043)}$	$0.002(\pm 0.003)$
$3 \to 8$	$0.848(\pm 0.007)$	$0.839(\pm 0.022)$	$0.842(\pm 0.011)$	$0.007(\pm 0.012)$	$0.848(\pm 0.007)$	$\mathbf{0.841(\pm 0.010)}$	$0.842(\pm 0.011)$	$0.007(\pm 0.012)$
$4 \to 5$	$0.984(\pm 0.004)$	$0.949(\pm 0.008)$	$0.951(\pm 0.009)$	$0.002(\pm 0.002)$	$0.984(\pm 0.004)$	$\mathbf{0.951(\pm 0.009)}$	$0.951(\pm 0.009)$	$0.002(\pm 0.002)$
Avg.	$0.731(\pm 0.084)$	$0.790(\pm 0.040)$	$0.790(\pm 0.023)$	$0.004(\pm 0.006)$	$\mathbf{0.806(\pm 0.024)}$	$\mathbf{0.793(\pm 0.027)}$	$0.790(\pm 0.023)$	$0.004(\pm 0.006)$

Table 33. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by DIRT (Shu et al., 2018).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$\mathbf{0.719(\pm 0.070)}$	$0.350(\pm 0.289)$	$0.705(\pm 0.043)$	$0.006(\pm 0.011)$	$0.712(\pm 0.055)$	$0.350(\pm 0.289)$	$0.705(\pm 0.043)$	$0.006(\pm 0.011)$
$1 \to 6$	$0.929(\pm 0.000)$	$0.410(\pm 0.429)$	$0.893(\pm 0.023)$	$0.001(\pm 0.001)$	$0.929(\pm 0.000)$	$\mathbf{0.411(\pm 0.431)}$	$0.893(\pm 0.023)$	$0.001(\pm 0.001)$
$2 \to 7$	$0.496(\pm 0.000)$	$\mathbf{0.495(\pm 0.002)}$	$0.495(\pm 0.001)$	$0.001(\pm 0.001)$	$0.496(\pm 0.000)$	$0.494(\pm 0.003)$	$0.495(\pm 0.001)$	$0.001(\pm 0.001)$
$3 \to 8$	$0.971(\pm 0.005)$	$0.926(\pm 0.005)$	$0.927(\pm 0.005)$	$0.000(\pm 0.000)$	$0.971(\pm 0.005)$	$\mathbf{0.927(\pm 0.005)}$	$0.927(\pm 0.005)$	$0.000(\pm 0.000)$
$4 \to 5$	$0.654(\pm 0.566)$	$\mathbf{0.460(\pm 0.417)}$	$0.939(\pm 0.005)$	$0.003(\pm 0.003)$	$\mathbf{0.980(\pm 0.000)}$	$0.457(\pm 0.413)$	$0.939(\pm 0.005)$	$0.003(\pm 0.003)$
Avg.	$0.754(\pm 0.128)$	$0.528(\pm 0.228)$	$0.792(\pm 0.015)$	$0.002(\pm 0.003)$	$\mathbf{0.818(\pm 0.012)}$	$0.528(\pm 0.228)$	$0.792(\pm 0.015)$	$0.002(\pm 0.003)$

Table 34. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by DSAN (Zhu et al., 2021).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$0.732(\pm 0.010)$	$0.719(\pm 0.025)$	$0.725(\pm 0.024)$	$0.000(\pm 0.000)$	$0.732(\pm 0.010)$	$\mathbf{0.725(\pm 0.024)}$	$0.725(\pm 0.024)$	$0.000(\pm 0.000)$
$1 \to 6$	$0.879(\pm 0.011)$	$0.856(\pm 0.019)$	$0.853(\pm 0.024)$	$0.003(\pm 0.004)$	$0.879(\pm 0.011)$	$0.853(\pm 0.024)$	$0.853(\pm 0.024)$	$0.003(\pm 0.004)$
$2 \to 7$	$0.455(\pm 0.015)$	$0.466(\pm 0.029)$	$0.474(\pm 0.024)$	$0.024(\pm 0.021)$	$0.455(\pm 0.015)$	$\mathbf{0.474(\pm 0.024)}$	$0.474(\pm 0.024)$	$0.024(\pm 0.021)$
$3 \to 8$	$0.818(\pm 0.005)$	$0.820(\pm 0.008)$	$0.822(\pm 0.004)$	$0.008(\pm 0.009)$	$0.818(\pm 0.005)$	$\mathbf{0.822(\pm 0.004)}$	$0.822(\pm 0.004)$	$0.008(\pm 0.009)$
$4 \to 5$	$0.911(\pm 0.005)$	$0.449(\pm 0.399)$	$0.899(\pm 0.031)$	$0.003(\pm 0.005)$	$0.911(\pm 0.005)$	$\mathbf{0.450(\pm 0.400)}$	$0.899(\pm 0.031)$	$0.003(\pm 0.005)$
Avg.	$0.759(\pm 0.009)$	$0.662(\pm 0.096)$	$0.754(\pm 0.021)$	$0.007(\pm 0.008)$	$0.759(\pm 0.009)$	$\mathbf{0.665(\pm 0.095)}$	$0.754(\pm 0.021)$	$0.007(\pm 0.008)$

Table 35. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by HoMM (Chen et al., 2020).

Scenario	No Iteration				Iteration			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	$0.746(\pm 0.008)$	$0.701(\pm 0.031)$	$0.716(\pm 0.026)$	$0.094(\pm 0.032)$	$0.746(\pm 0.008)$	$\mathbf{0.716(\pm 0.026)}$	$0.716(\pm 0.026)$	$0.094(\pm 0.032)$
$1 \to 6$	$0.897(\pm 0.002)$	$0.864(\pm 0.009)$	$0.867(\pm 0.006)$	$0.001(\pm 0.002)$	$0.897(\pm 0.002)$	$\mathbf{0.867(\pm 0.006)}$	$0.867(\pm 0.006)$	$0.001(\pm 0.002)$
$2 \to 7$	$0.488(\pm 0.010)$	$0.484(\pm 0.009)$	$0.493(\pm 0.009)$	$0.024(\pm 0.007)$	$0.488(\pm 0.010)$	$\mathbf{0.493(\pm 0.009)}$	$0.493(\pm 0.009)$	$0.024(\pm 0.007)$
$3 \to 8$	$0.839(\pm 0.012)$	$0.845(\pm 0.032)$	$0.864(\pm 0.012)$	$0.000(\pm 0.000)$	$0.839(\pm 0.012)$	$\mathbf{0.863(\pm 0.013)}$	$0.864(\pm 0.012)$	$0.000(\pm 0.000)$
$4 \to 5$	$0.928(\pm 0.006)$	$\mathbf{0.456(\pm 0.410)}$	$0.923(\pm 0.012)$	$0.000(\pm 0.000)$	$0.928(\pm 0.006)$	$0.455(\pm 0.409)$	$0.923(\pm 0.012)$	$0.000(\pm 0.000)$
Avg.	$0.780(\pm 0.008)$	$0.670(\pm 0.098)$	$0.773(\pm 0.013)$	$0.024(\pm 0.008)$	$0.780(\pm 0.008)$	$\mathbf{0.679(\pm 0.092)}$	$0.773(\pm 0.013)$	$0.024(\pm 0.008)$

Table 36. Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR for the aggregation method in objective (9) with models computed by MMDA (Rahman et al., 2020).