IMPROVED DENSITY RATIO ESTIMATION FOR EVALUATING SYNTHETIC DATA QUALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

High-quality synthetic data is essential for accurate downstream analysis. Density Ratio Estimation (DRE) has emerged as a powerful tool for evaluating synthetic data quality. However, existing DRE methods are highly sensitive to hyperparameter selection, where suboptimal choices lead to poor convergence rates and degraded empirical performance. To mitigate this, we propose a novel model aggregation algorithm for DRE that trains multiple models with diverse hyperparameter configurations and combines their outputs. Our approach achieves fast convergence without requiring prior knowledge of the unknown density ratio smoothness and is minimax optimal for the squared loss. We demonstrate that our method enhances the performance of established DRE techniques across benchmark datasets, achieving state-of-the-art results on MiniDomainNet and Amazon Reviews.

1 INTRODUCTION

In the field of synthetic data evaluation, three primary categories of utility measures have been identified by Drechsler & Haensch (2024): Fit-for-purpose measures, analysis-specific utility measures, and global utility measures. Fit-for-purpose measures primarily assess the univariate distributions of observed and synthetic data, often employing visualization techniques or goodness-of-fit tests. While these measures offer an initial assessment of the synthesis model's performance, their scope is inherently limited, as they typically examine only one or two variables at a time. To capture more complex relationships, analysis-specific utility measures evaluate whether analyses conducted on synthetic data yield results comparable to those obtained from the original data. For instance, these measures can assess the similarity of regression model coefficients (Karr et al., 2006) or compare the predictive performance of models trained on synthetic versus observed data based on relevant evaluation metrics. However, a key limitation of analysis-specific utility measures is their lack of generalizability, strong utility for one analysis does not necessarily imply high utility for another.

Global utility measures offer a potential solution to the limitations of previous approaches by assessing the divergence between the entire multivariate distributions of the observed and synthetic data. Consequently, they represent one of the most promising classes of utility measures, as the closer the multivariate distributions of the observed and synthetic data, the more similar the results of any subsequent analyses are expected to be. Global utility is typically quantified using divergence metrics, such as the Kullback-Leibler divergence, or by determining whether a classification model can effectively distinguish between observed and synthetic data, a method known as propensity score mean squared error (pMSE) (Snoke et al., 2018). Despite their advantages, global utility measures are often criticized for being overly general (Drechsler, 2022). Their outputs can be difficult to interpret and provide limited insight into specific regions where the synthetic data fail to accurately capture the characteristics of the true data.

To address the limitations associated with conventional global utility measures, Volker et al. (2024) recently proposed density ratio estimation (DRE) as an alternative approach for utility assessment. Let $\{x_m\}_{m=1}^M$ and $\{x_n\}_{n=1}^N$, $M, N \in \mathbb{N}$ be two samples that are independently and identically distributed (i.i.d.) according to two probability density functions p and q, respectively. Then the objective of DRE is to learn the density ratio $\beta(x) := \frac{p(x)}{q(x)}$ from the samples, see Figure 1 (Left) for an illustration. Conceptually, when two datasets exhibit similar multivariate distributions, the

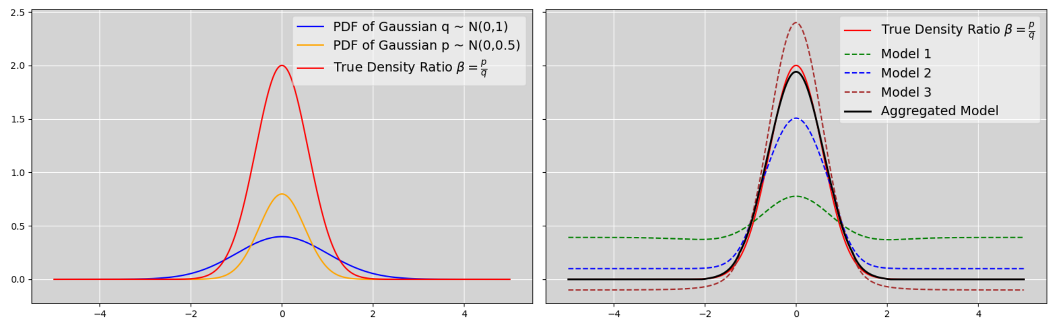

Figure 1: Left: Density ratio $\beta = \frac{p}{q}$ between Normal distribution $p = \mathcal{N}(0, 0.5)$ and $q = \mathcal{N}(0, 1)$. Right: Procedures for selecting a single Model 1–3 (dashed) cannot approximate β well. In contrast, our aggregation strategy (solid) achieves a good approximation.

density ratio remains approximately equal to 1 across the data range. Conversely, if the distributions of the observed and synthetic data diverge significantly, the density ratio deviates substantially from 1 in regions where discrepancies between the distributions are most pronounced.

However, the performance of DRE methods is highly sensitive to the choice of hyperparameters, and poorly chosen hyperparameters often result in suboptimal convergence rates and degraded empirical performance, see Figure 1 (Right) for a simplified example. In this work, we therefore study the problem of resolving hyperparameter choice issues for DRE methods.

2 Related Work

Global utility measures are commonly used to quantify the degree of distributional similarity between observed and synthetic data samples. One effective approach for assessing this similarity involves evaluating whether a classification model can reliably differentiate between samples from the two distributions. If a classification model achieves high accuracy in distinguishing between observed and synthetic data, it indicates a low level of distributional similarity, thereby implying reduced global utility. The propensity score mean squared error (pMSE) introduced by Woo et al. (2009) provides a formalized metric for capturing this relationship. Let $\{x_m\}_{m=1}^{M}$, $\{x_n\}_{n=1}^{N}$ be a real and synthetic dataset respectively, and $s(x_i)$ the predicted probability of a classifier that a datapoint x_i is synthetic. Then

$$pMSE = \frac{1}{M+N} \sum_{i=1}^{M+N} \left(s(x_i) - \frac{N}{M+N} \right).$$

Snoke et al. (2018) extended this approach by comparing the pMSE value with its expected value under the null hypothesis that the real and synthetic data are indistinguishable. In addition to the pMSE, various alternative measures can be derived from the estimated propensity scores. These include the percentage of correctly classified records (Raab et al., 2021) and the Kolmogorov-Smirnov statistic (Bowen et al., 2021), both of which exhibit a strong correlation with the pMSE. Drechsler (2022) demonstrated that the utility score is highly sensitive to the choice of the propensity score model, emphasizing that substantial improvements in synthetic data generation models may not always be reflected in the pMSE. Furthermore, selecting an appropriate propensity score model presents challenges, as common issues in model selection, such as the bias-variance trade-off, remain relevant.

An alternative approach to quantifying distributional similarity is by utilizing the Kullback-Leibler (KL) divergence which is approximated by density estimation techniques (Karr et al., 2006; Wang et al., 2009). However, density estimation is one of the most challenging tasks in statistical learning, inherently introducing estimation errors, particularly in high-dimensional settings (Sugiyama et al., 2012a). These errors become further exacerbated when computing the ratio of estimated densities, as inaccuracies in both density estimates magnify during the division process. Direct DRE mitigates this issue by modeling and estimating the density ratio directly, without the need for separate density estimations.

Recognizing this advantage, Volker et al. (2024) proposed the use of DRE for assessing data utility. Thereby, they can evaluate global utility by calculating some f-divergence between the real and synthetic data distributions. Moreover, using a density ratio estimator also allows for evaluating local utility by using point evaluations $\beta(x_i) := \frac{p(x_i)}{q(x_i)}$. It was first observed by Sugiyama et al. (2012b) that a broad class of methods for DRE use the objective

$$f_{\mathcal{H}} := \arg\min_{f \in \mathcal{H}} B_F(\beta, g(f)) \tag{1}$$

and derive the desired density ratio estimator by $\beta_{\mathcal{H}} := g(f_{\mathcal{H}})$, where $g : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function and

$$B_F\left(\beta, \widehat{\beta}\right) := \mathbb{E}_{x \sim Q}[F(\beta(x)) - F(\widehat{\beta}(x)) - F'(\widehat{\beta}(x))[\beta(x) - \widehat{\beta}(x)]]$$

denotes the Bregman divergence with prescribed generator $F : \mathbb{R} \to \mathbb{R}$, and $\mathcal{H}$ a given model class. Consider the following example (cf. Sugiyama et al. (2012b); Menon & Ong (2016); Zellinger et al. (2023)):

Example 1.

1. *As shown in Kanamori et al. (2012b) the kernel unconstrained least squares importance fitting procedure (KuLSIF) (Kanamori et al., 2009) is realized by using $F(x) = (x-1)^2/2$ and $g(x) = x$ in equation (1) such that $B_F(\beta, g(f)) = \frac{1}{2}\|\beta - g(f)\|_{L^2(Q)}$.*
2. *Menon & Ong (2016) use $F(x) = x^{3/2}$ and $g(x) = e^{2x}$ in equation (1) to obtain the exponential function approach (Exp) as applied for AdaBoost (Freund et al., 1996).*
3. *The square loss approach (SQ) in Menon & Ong (2016) uses $F(x) = \frac{1}{2x+2}$ and $g(x) = \frac{-1+2x}{2-2x}$.*
4. *To realize the logistic regression (Nelder & Wedderburn, 1972) (LR) approach as used in Bickel et al. (2009) $F(x) = x\log(x) - (1+x)\log(1+x)$ and $g(x) = e^x$ are set in equation 1.*

All these methods share hyperparameter choice issues which we approach by model aggregation as explained in the next section in more detail.

3 Method

Algorithm 1 DRE Aggregation

Input: K different hyperparameter setups, dataset $\{(x_n, y_n)\}_{n=1}^{M+N} \overset{\text{iid}}{\sim} \rho$.

Output: Aggregation $\widehat{\beta}(x) = g\left(\sum_{k=1}^K \widehat{\alpha}_k f_k(x)\right)$ with optimal coefficients $\widehat{\boldsymbol{\alpha}} := (\widehat{\alpha}_1, \ldots, \widehat{\alpha}_K) \in \mathbb{R}^K$.

Step 1: Train one model $\{\beta_k\}_{k=1}^K$ for each hyperparameter setup (e.g., by a method in Example 1) and denote by $f_k = g^{-1}(\beta_k)$ the associated binary classifiers.

Step 2: Compute aggregation weights $\widehat{\boldsymbol{\alpha}} = \widehat{\mathbf{G}}^{-1}\widehat{\mathbf{r}}$ with empirical Gram matrix $\widehat{\mathbf{G}}$ and inner product vector $\widehat{\mathbf{r}}$ by

$$\widehat{\mathbf{G}} = \left(\frac{1}{M+N}\sum_{n=1}^{M+N} h(x_n, y_n) f_k(x_n) f_j(x_n)\right)_{k,j=1}^K \quad \widehat{\mathbf{r}} = \left(\frac{1}{M+N}\sum_{n=1}^{M+N} h(x_n, y_n) f_n(x_n) y_n\right)_{k=1}^K$$

with Hessian $h(x, y)$ depending on the chosen binary loss, see Appendix B for the example of LR.

Return: Aggregation $\widehat{\beta}(x) = g\left(\sum_{k=1}^K \widehat{\alpha}_k f_k(x)\right)$.

Notation Following Menon & Ong (2016) the optimization problem (1) can be equivalently formulated as binary classification with loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ and $\mathcal{Y} := \{-1, 1\}$, such that $\ell(1, f(x))$ and $\ell(-1, f(x))$ measure the error of a classifier $f(x)$ predicting whether x is originating from P or Q. For this, we only need the assumption that the combined sample $\{x_m\}_{m=1}^M \cup \{x_n\}_{n=1}^N$ is an i.i.d. sample from the distribution ρ constructed as follows (Zellinger et al., 2023): it is defined as a probability measure on $\mathcal{X} \times \mathcal{Y}$ with conditionals $\rho(x|y=1) := p(x), \rho(x|y=-1) := q(x)$, and, marginal $\rho_{\mathcal{Y}}$ defined as Bernoulli measure such that the probability for both events $y = 1, y = -1$ is $\frac{1}{2}$. Then the optimization problem (1) is equivalent to the minimization (Menon & Ong, 2016)

$$f_{\mathcal{H}} := \arg\min_{f \in \mathcal{H}} \mathcal{R}(f) \tag{2}$$

with expected risk

$$\mathcal{R}(f) := \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, f(x)) d\rho(x, y)$$

and the density ratio estimator $\beta_{\mathcal{H}} := g(f_{\mathcal{H}})$ can be recovered from the classifier $f_{\mathcal{H}}$. For instance, all the methods discussed in Example 1 can be optimized through risk minimization (2) by using the corresponding loss functions.

Problem Given a finite number $\beta_1, \ldots, \beta_K : \mathcal{X} \to \mathbb{R}$ of density ratio models, each trained with a different hyperparameter setting, the goal is to find a model $\widehat{\beta} : \mathcal{X} \to \mathbb{R}$ with minimal error $B_F(\beta, \widehat{\beta})$.

Approach We approach the problem by a linear aggregation of models

$$\widehat{\beta} := g\left(\sum_{k=1}^K \alpha_k f_k\right) \tag{3}$$

with $f_k := g^{-1}(\beta_k)$ being the binary classifier associated to to the density ratio estimator β_k and the aggregation weights $\alpha_1, \ldots, \alpha_K$ which we compute as follows.

First, we follow Menon & Ong (2016) and Marteau-Ferey et al. (2019) to bound the Bregman divergence in equation 1 by a norm (cf. Zellinger et al. (2023) for this strategy)

$$B_F\left(\beta, \widehat{\beta}\right) - B_F(\beta, g(f_{\mathcal{H}})) = 2\big(\mathcal{R}(f) - \mathcal{R}(f_{\mathcal{H}})\big) \leq 2 \cdot \left\|g^{-1}(\widehat{\beta}) - f_{\mathcal{H}}\right\|^2$$

with some norm $\|.\|$ that depends on the choices F and g, see Appendix B for its derivations. Our approach is to choose the aggregation weights $\alpha_1, \ldots, \alpha_k$ that minimize this upper bound

$$\min_{\alpha_1, \ldots, \alpha_K \in \mathbb{R}} \left\|\sum_{k=1}^K \alpha_k f_k - f_{\mathcal{H}}\right\|^2 \tag{4}$$

which has two advantages: (a) an analytic solution by functional least-squares and (b) fast provable convergence rate for estimators.

Functional least squares aggregation A necessary optimality condition for equation 4 leads to the solution (Chen et al., 2015; Dinu et al., 2023)

$$\begin{aligned}
L(\alpha) &:= \left\|\sum_{k=1}^K \alpha_k f_k - f_{\mathcal{H}}\right\|^2 = \left\langle \sum_{k=1}^K \alpha_k f_k - f_{\mathcal{H}}, \sum_{k=1}^K \alpha_k f_k - f_{\mathcal{H}} \right\rangle \\
&= \sum_{k,j=1}^K \alpha_k \alpha_j \langle f_k, f_j \rangle - 2 \sum_{k=1}^K \alpha_k \langle f_k, f_{\mathcal{H}} \rangle + \langle f_{\mathcal{H}}, f_{\mathcal{H}} \rangle \\
\frac{\partial L(\alpha)}{\partial \alpha_k} &= 2\left(\sum_{k=1}^K \alpha_k \langle f_k, f_j \rangle - \langle f_k, f_{\mathcal{H}} \rangle\right) = 0 \\
\boldsymbol{\alpha} &= \mathbf{G}^{-1}\mathbf{r}
\end{aligned}$$

with $\mathbf{G} = (\langle f_k, f_j \rangle)_{k,j=1}^K, \mathbf{r} = (\langle f_k, f_{\mathcal{H}} \rangle)_{i=1}^K$. After discretization, we arrive at Algorithm 1.

Relation to model selection methods In the continuous case (i.e., with access to p, q), our aggregation strategy is at least as good as model selection:

$$\min_{\alpha_1,\ldots,\alpha_K\in\mathbb{R}}\left\|\sum_{k=1}^{K}\alpha_k f_k - f_{\mathcal{H}}\right\|^2 \le \min_{\alpha_1,\ldots,\alpha_k\in\{0,1\}}\left\|\sum_{k=1}^{K}\alpha_k f_k - f_{\mathcal{H}}\right\|^2 \le \min_{k\in\{1,\ldots,K\}}\left\|f_k - f_{\mathcal{H}}\right\|^2 . \tag{5}$$

Our next result shows that this also holds in the practical case where we only use finite samples from p, q.

Theorem 1. *Let assumptions 1–4 and technical assumptions from Appendix A be satisfied. Consider $K > 1$, $\delta > 0$, $\{\lambda_k\}_{k=1}^K$ as defined in Appendix A and associated $\widehat{f}^{\lambda_k}$ as in Equation 10. Then we have that for $\widehat{\beta}$ of Algorithm 1 applied with $\beta_k := g(\widehat{f}^{\lambda_k})$:*

$$B_F\left(\beta,\widehat{\beta}\right) - B_F(\beta, g(f_{\mathcal{H}})) \le C(M+N)^{-\frac{2r\alpha+\alpha}{2r\alpha+\alpha+1}}, \tag{6}$$

with probability at least $1-(9+2K)\delta$ for sufficiently large $M+N$ and $C>0$ independent of M, N.

Remark 1. *To the best of our knowledge Theorem 1 provides the first provable principled way of achieving minimax optimal convergence rates for a parameter choice procedure in DRE, a full proof can be found in Appendix A.*

In the following, we go beyond theoretical convergence results and conduct a comprehensive empirical evaluation of our algorithm across multiple benchmark datasets.

4 Experiments

We investigate the performance of our aggregation approach for resolving parameter choice issues within the following areas of application. For all experiments, the real and synthetic data distribution are represented by p and q respectively.

Datasets with known density ratios Building upon Kanamori et al. (2012b) we create ten distinct datasets using Gaussian Mixture distributions in a 50-dimensional space. Each, real and synthetic data distribution gets assigned a Gaussian Mixture distribution, see Appendix C for more details.

Domain Aaptation datasets To evaluate the effect of our algorithm in large-scale real-world scenarios we conduct experiments involving importance weighting for domain adaptation. Given a source dataset $\{x'_n\}_{n=1}^N \overset{\text{iid}}{\sim} Q$ with labels $\{y_n\}_{n=1}^N$ (which represents synthetic data) and a target dataset $\{x_i\}_{i=1}^M \overset{\text{iid}}{\sim} P$ *without labels* (which represents real data), the goal is to learn a model f' with low expected risk $\mathcal{R}(f') := \mathbb{E}_{(x,y)\sim P}[\ell(y, f'(x))]$ on the target domain without having sampled labels from the target domain. Motivated by Sugiyama et al. (2007) this task can be approached by approximating the density ratio $\frac{p}{q}$ with estimator $\widehat{\beta}$ and empirical risk minimization such that

$$f' = \arg\min_{f\in\mathcal{H}} \frac{1}{N}\sum_{n=1}^{N}\widehat{\beta}(x_n)\ell(y_n, f(x_n)).$$

We evaluate on image, text, and time series datasets, see Appendix C for more details.

Methods To test our aggregation approach we pick four popular representatives of the large class of widely used DRE methods that can be modeled as Bregman divergences as in equation (1). For this we use the DRE methods presented in Example 1 with cross-validation as hyperparameter selection procedure as a baseline and compare this for each of the four methods against our aggregation approach. In the domain adaptation task we evaluate this for 11 different domain adaption methods. We refer to Appendix C for more details.

Empirical Results Our proposed aggregation approach consistently outperforms all evaluated DRE methods in experiments involving geometric datasets with known density ratios (Table 1 and benchmark domain adaptation scenarios(Table 2 and Table 4). Notably, model aggregation consistently yields superior results compared to non-aggregated baselines, both on geometric datasets and in domain adaptation tasks, when performance is averaged across multiple domain adaptation scenarios. More detailed results, including the performance of individual DRE methods across specific domain adaptation scenarios, are presented in Tables 5–37. Furthermore, our approach achieves new state-of-the-art performance in deep domain adaptation, as demonstrated on the MiniDomainNet and Amazon Reviews benchmarks. We additionally conduct three ablation studies. One shows that our algorithm assigns higher weights to more accurate models (Figure 2), one shows that our algorithm can also be used in a heuristic setting where not all assumption of Theorem 1 hold (Table 3). The final one shows that our algorithm is better than model averaging (Table 3).

	Geometric Figures							
	Cross-Validation for Binary Classifier				**Aggregation**			
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)	**8.320**(±**0.004**)	**8.151**(±**0.011**)	**8.572**(±**0.016**)	**8.831**(±**0.011**)
c2,d1.72	13.031(±0.005)	12.994(±0.015)	13.255(±0.013)	13.537(±0.027)	**12.854**(±**0.007**)	**12.365**(±**0.011**)	**13.250**(±**0.009**)	**13.102**(±**0.019**)
c2,d1.59	12.625(±0.005)	19.748(±0.037)	12.829(±0.014)	13.056(±0.015)	**12.422**(±**0.010**)	**12.441**(±**0.042**)	**12.719**(±**0.012**)	**12.615**(±**0.011**)
c1,d1.55	11.813(±0.007)	14.477(±0.103)	12.001(±0.023)	12.179(±0.013)	**11.625**(±**0.004**)	**11.324**(±**0.018**)	**12.001**(±**0.015**)	**11.458**(±**0.010**)
c2,d1.78	9.632(±0.003)	18.008(±0.069)	9.802(±0.035)	9.990(±0.006)	**9.425**(±**0.013**)	**17.043**(±**0.015**)	**9.702**(±**0.023**)	**9.625**(±**0.007**)
c2,d1.55	10.371(±0.007)	9.774(±0.019)	10.555(±0.059)	10.757(±0.023)	**10.001**(±**0.002**)	**9.523**(±**0.010**)	**10.415**(±**0.039**)	**10.317**(±**0.019**)
c3,d1.57	12.014(±0.003)	18.995(±0.126)	12.214(±0.037)	14.048(±0.029)	**12.003**(±**0.007**)	**12.021**(±**0.008**)	**11.238**(±**0.013**)	**12.940**(±**0.018**)
c2,d1.61	11.614(±0.004)	11.282(±0.034)	11.800(±0.008)	12.242(±0.007)	**11.365**(±**0.003**)	**10.787**(±**0.013**)	**10.920**(±**0.008**)	**11.891**(±**0.007**)
c3,d1.46	12.803(±0.009)	12.616(±0.008)	12.971(±0.007)	13.159(±0.006)	**9.421**(±**0.003**)	**12.025**(±**0.010**)	**12.132**(±**0.004**)	**12.970**(±**0.004**)
c1,d1.63	9.527(±0.006)	9.704(±0.009)	9.732(±0.014)	9.965(±0.015)	**9.397**(±**0.008**)	**9.611**(±**0.011**)	**9.071**(±**0.007**)	**9.729**(±**0.009**)
Avg	11.205(±0.006)	13.392(±0.097)	11.400(±0.023)	11.810(±0.015)	**10.683**(±**0.006**)	**11.529**(±**0.015**)	**11.002**(±**0.014**)	**11.348**(±**0.012**)

Table 1: Mean and standard deviation (after $\pm$) 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.

	Domain Adaptation: MiniDomainNet							
	Cross-Validation for Binary Classifier				**Aggregation**			
DA-Method	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
MMDA	0.527(±0.009)	0.528(±0.011)	0.528(±0.012)	0.518(±0.010)	**0.536**(±**0.007**)	**0.539**(±**0.011**)	**0.536**(±**0.006**)	**0.535**(±**0.009**)
CoDATS	0.536(±0.012)	0.532(±0.015)	0.530(±0.020)	0.517(±0.018)	**0.542**(±**0.010**)	**0.543**(±**0.014**)	**0.540**(±**0.014**)	**0.533**(±**0.017**)
DANN	0.531(±0.009)	0.522(±0.012)	0.520(±0.019)	0.506(±0.016)	**0.536**(±**0.007**)	**0.535**(±**0.011**)	**0.526**(±**0.013**)	**0.519**(±**0.015**)
CDAN	0.531(±0.012)	0.531(±0.017)	0.524(±0.023)	0.512(±0.021)	**0.537**(±**0.009**)	**0.544**(±**0.017**)	**0.535**(±**0.017**)	**0.526**(±**0.020**)
DSAN	0.539(±0.011)	0.532(±0.015)	0.527(±0.015)	0.513(±0.013)	**0.544**(±**0.009**)	**0.546**(±**0.014**)	**0.535**(±**0.009**)	**0.525**(±**0.012**)
DIRT	0.517(±0.026)	0.386(±0.177)	0.520(±0.023)	0.509(±0.021)	**0.523**(±**0.025**)	**0.395**(±**0.175**)	**0.526**(±**0.017**)	**0.525**(±**0.020**)
AdvSKM	0.516(±0.006)	0.515(±0.009)	0.512(±0.012)	0.500(±0.010)	**0.522**(±**0.005**)	**0.529**(±**0.008**)	**0.521**(±**0.006**)	**0.517**(±**0.008**)
HoMM	0.531(±0.008)	0.529(±0.012)	0.521(±0.015)	0.505(±0.013)	**0.539**(±**0.007**)	**0.544**(±**0.011**)	**0.529**(±**0.009**)	**0.528**(±**0.012**)
DDC	0.517(±0.010)	0.517(±0.013)	0.514(±0.012)	0.500(±0.010)	**0.527**(±**0.008**)	**0.529**(±**0.012**)	**0.524**(±**0.008**)	**0.515**(±**0.009**)
DeepCoral	0.535(±0.012)	0.528(±0.013)	0.530(±0.012)	0.514(±0.011)	**0.543**(±**0.010**)	**0.540**(±**0.012**)	**0.536**(±**0.007**)	**0.527**(±**0.009**)
CMD	0.529(±0.010)	0.524(±0.021)	0.521(±0.021)	0.510(±0.019)	**0.536**(±**0.008**)	**0.538**(±**0.020**)	**0.527**(±**0.015**)	**0.521**(±**0.018**)
Avg.	0.528(±0.011)	0.513(±0.029)	0.523(±0.017)	0.510(±0.015)	**0.535**(±**0.010**)	**0.526**(±**0.028**)	**0.530**(±**0.011**)	**0.525**(±**0.014**)

Table 2: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet datasets over three different random initialization of model weights and several domain adaptation tasks.

5 Conclusion

In this work, we proposed an algorithm to resolve hyperparameter choice issues in DRE to improve the evaluation of synthetic data quality. We approach these issues by introducing a novel model aggregation algorithm to DRE that first trains a sequence of models with distinct hyperparameter settings. Subsequently, the respective model outputs are aggregated via a linear combination where the coefficients are optimized such that the algorithm achieves fast convergence rates without requiring prior knowledge of the unknown density ratio smoothness and is optimal for square loss. We supported this theoretical result by a comprehensive empirical evaluation on both, benchmarks for DRE and large-scale deep domain adaptation. All experimental results confirm that our algorithm clearly outperforms established DRE methods. Furthermore, on domain adaptation benchmarks, we achieved new state-of-the-art performance on the MiniDomainNet and Amazon Reviews datasets.

References

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

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

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9), 2009.

Lucien Birgé. An alternative point of view on lepski's method. *Lecture Notes-Monograph Series*, pp. 113–133, 2001.

Gilles Blanchard, Peter Mathé, and Nicole Mücke. Lepskii principle in supervised learning. *arXiv preprint arXiv:1905.10764*, 2019.

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

Claire McKay Bowen, Fang Liu, and Bingyue Su. Differentially private data release via statistical election to partition sequentially: Statistical election to partition sequentially. *Metron*, 79(1):1–31, 2021.

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

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

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

Jieyang Chen, Sergiy Pereverzyev Jr, and Yuesheng Xu. Aggregation of regularized solutions from multiple observation models. *Inverse Problems*, 31(7):075005, 2015.

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

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

Jörg Drechsler. Challenges in measuring utility for fully synthetic data. In *International Conference on Privacy in Statistical Databases*, pp. 220–233. Springer, 2022.

Jörg Drechsler and Anna-Carolina Haensch. 30 years of synthetic data. *Statistical Science*, 39(2): 221–242, 2024.

Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pp. 148–156. Citeseer, 1996.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Alexander Goldenshluger and Sergei V Pereverzev. Adaptive estimation of linear functionals in Hilbert scales from indirect white noise observations. *Probability Theory and Related Fields*, 118 (2):169–186, 2000.

Lukas Gruber, Markus Holzleitner, Johannes Lehner, Sepp Hochreiter, and Werner Zellinger. Overcoming saturation in density ratio estimation by iterated regularization. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 16502–16529, 2024.

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

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

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

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

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

Alan F Karr, Christine N Kohnen, Anna Oganian, Jerome P Reiter, and Ashish P Sanil. A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician*, 60 (3):224–232, 2006.

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

OV Lepskii. On a problem of adaptive estimation in gaussian white noise. *Theory of Probability & Its Applications*, 35(3):454–466, 1991.

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

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

Shuai Lu, Peter Mathé, and Sergei V Pereverzev. Balancing principle in supervised learning for a general regularization scheme. *Applied and Computational Harmonic Analysis*, 48(1):123–148, 2020.

Ulysse Marteau-Ferey, Dmitrii Ostrovskii, Francis Bach, and Alessandro Rudi. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. In *Conference on Learning Theory*, pp. 2294–2340. PMLR, 2019.

Peter Mathé. The Lepskii principle revisited. *Inverse problems*, 22(3):L11, 2006.

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

Nicole Mücke. Adaptivity for regularized kernel methods by Lepskii's principle. *arXiv preprint arXiv:1804.05433*, 2018.

John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 135(3):370–384, 1972.

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

Dmitrii M Ostrovskii and Francis Bach. Finite-sample analysis of m-estimators using self-concordance. 2021.

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

Gillian M Raab, Beata Nowok, and Chris Dibben. Assessing, visualizing and improving the utility of synthetic data. *arXiv preprint arXiv:2109.12717*, 2021.

Mohamed Ragab, Emadeldeen Eldele, Wee Ling Tan, Chuan-Sheng Foo, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. Adatime: A benchmarking suite for domain adaptation on time series data. *ACM Transactions on Knowledge Discovery from Data*, 17(8):1–18, 2023.

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

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

Bernhard Schölkopf and Alexander J Smola. Learning with kernels. In *Learning with kernels*. MIT Press, 2002.

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

Joshua Snoke, Gillian M Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 181(3):663–688, 2018.

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

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012a.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 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.

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

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

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Thom Benjamin Volker, Peter-Paul de Wolf, and Erik-Jan van Kesteren. A density ratio framework for evaluating the utility of synthetic data. *arXiv preprint arXiv:2408.13167*, 2024.

Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5): 2392–2405, 2009.

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

Mi-Ja Woo, Jerome P Reiter, Anna Oganian, and Alan F Karr. Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality*, 1(1), 2009.

Werner Zellinger. Binary losses for density ratio estimation. *International Conference on Learning Representations*, 2025.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *International Conference on Learning Representations*, 2017.

Werner Zellinger, Natalia Shepeleva, Marius-Constantin Dinu, Hamid Eghbal-zadeh, Hoan Duc Nguyen, Bernhard Nessler, Sergei Pereverzyev, and Bernhard A Moser. The balancing principle for parameter choice in distance-regularized domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing*, 30:8008–8018, 2021.

A THEORETICAL ANALYSIS

In the following we show that our aggregation approach (3) achieves fast convergence rates when (1) is optimized in some RKHS $\mathcal{H}$ with norm $\|.\|_{\mathcal{H}}$ and Tikhonov penalty $\|f\|_{\mathcal{H}}^2$ as follows

$$f^{\lambda} := \underset{f\in\mathcal{H}}{\arg\min}\left[B_F(\beta, g(f)) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2\right], \tag{7}$$

where $\lambda > 0$ is the regularization parameter which we aim to choose. All DRE methods in Example 1 fit into the optimization problem (7), see (Zellinger et al., 2023; Gruber et al., 2024). We fix a sequence $(\lambda_k)_{k=1}^K$ of regularization parameters and refine our aggregation (3) by

$$\widehat{\beta} := g\left(\sum_{k=1}^K \alpha_k f^{\lambda_k}\right).$$

To prove that our Algorithm 1 achieves fast convergence rates, we introduce a set of assumptions typical in learning theory, such as assumptions on data, the regularity of the underlying problem and the previously introduced loss function ℓ.

Assumption 1. *The loss function $\ell : \mathcal{Y}\times\mathbb{R}\to\mathbb{R}$ which is used for separating $\{(x_m,1)\}_{m=1}^M$ from $\{(x'_n,-1)\}_{n=1}^N$ has an associated link function $\psi : [0,1]\to\mathbb{R}$ with the following properties:*

- *ψ is invertible and the associated conditional Bayes risk $G(u) := u\ell(1,\psi(u)) + (1-u)\ell(-1,\psi(u))$ is twice differentiable,*
- *the minimizer $f_{\mathcal{H}}$ satisfies $f_{\mathcal{H}}(x) = \psi(\rho(y=1|x))$.*

Assumption 2 (source condition). *There exist some $r\in(0,\frac{1}{2}]$, $v\in\mathcal{H}$ satisfying $f_{\mathcal{H}} = \mathbf{H}(f_{\mathcal{H}})^r v$ for the expected Hessian $\mathbf{H}(f) := \mathbb{E}_{(x,y)\sim\rho}[\nabla^2\ell(y,f(x))]$.*

Assumption 3 (capacity condition). *There exist $\alpha\ge 1$ and $S>0$ such that $\mathrm{df}_\lambda \le S\lambda^{-\frac{1}{\alpha}}$ with the degrees of freedom*

$$\mathrm{df}_\lambda := \mathbb{E}_{(x,y)\sim\rho}\left[\left\|\mathbf{H}_\lambda(f_{\mathcal{H}})^{-\frac{1}{2}}\nabla\ell(y,f_{\mathcal{H}}(x))\right\|_{\mathcal{H}}^2\right]$$

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

Both, source and capacity condition, are typically used in learning theory (Caponnetto & De Vito, 2007; Bauer et al., 2007; Marteau-Ferey et al., 2019) to encode the regularity of the underlying problem. We also need the following self-concordance assumption (Bach, 2010; Ostrovskii & Bach, 2021) on our loss function, which is known to be satisfied for the examples we discussed (cf. Zellinger et al. (2023)).

Assumption 4 (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*

$$\left|\ell_y'''(\eta)\right| \le \ell_y''(\eta). \tag{8}$$

Note that by Lemma 3 in Gruber et al. (2024), Assumption 4 implies generalized self concordance, which is e.g. commonly employed in Marteau-Ferey et al. (2019); Zellinger et al. (2023). As a next step we discuss the specific norms that occur in (4) and (5) in more detail, see Appendix B for an example. From Menon & Ong (2016); Marteau-Ferey et al. (2019) we know that

$$B_F\left(\beta,\widehat{\beta}\right)-B_F(\beta,g(f_{\mathcal{H}}))\leq 2\left\|\widehat{\beta}-f_{\mathcal{H}}\right\|^2_{\mathbf{H}_\lambda(f_{\mathcal{H}})}$$

for $f_{\mathcal{H}},\mathbf{H}_\lambda(f_{\mathcal{H}})$ as in Assumption 2, and

$$\|f\|_{\mathbf{H}_\lambda(f_{\mathcal{H}})}:=\left\|\mathbf{H}_\lambda(f_{\mathcal{H}})^{1/2}f\right\|_{\mathcal{H}}. \tag{9}$$

However, the norm values $\|\cdot\|_{\mathbf{H}_\lambda(f_{\mathcal{H}})}$ are not directly accessible, as we have only observation from the measure ρ. That is, we have to estimate the norm to find an approximate solution to the optimization problem (4).

In the following, consider some small regularization parameter value $\lambda>0$ (as precisely specified in Section A) and the empirical risk minimizer and empirical Hessian

$$\begin{aligned}\widehat{\mathbf{H}}_\lambda(f)&:=\frac{1}{M+N}\sum_{n=1}^{M+N}\nabla^2\ell(y_n,f(x_n))+\lambda I,\\ \widehat{f}^\lambda&:=\operatorname*{arg\,min}_{f\in\mathcal{H}}\frac{1}{M+N}\sum_{n=1}^{M+N}\ell(y_n,f(x_n))+\frac{\lambda}{2}\|f\|^2_{\mathcal{H}}.\end{aligned}\tag{10}$$

To prove our main theorem, we will need to recall several concentration bounds on weighted norms, which were to a large extend established by Marteau-Ferey et al. (2019). We will need the following technical assumption:

Assumption 5 (technical assumption)**.**
- *The kernel k is continuous and bounded* $\sup_{x\in\mathcal{X}}\|k(x,\cdot)\|_{\mathcal{H}}\leq R$.
- *Let $z=(x,y)\in\mathcal{X}\times\mathcal{Y}$ and denote $\ell_z(f)=\ell(y,f(x))$. Then the quantities* $|\ell_z(0)|,\|\nabla\ell_z(0)\|_{\mathcal{H}},\mathrm{Tr}(\nabla^2\ell_z(0))$ *are almost surely (wrt. ρ) bounded.*

The first estimate deals with estimates for empirical risk minimizers $\widehat{f}^\lambda$ in weighted norms.

Lemma 1 (Marteau-Ferey et al. (2019), Theorem 38)**.** *Let Assumptions 1–5 be satisfied, $\delta\in(0,\frac{1}{2}]$, and define B_1^*, B_2^* and L by*

$$B_1^*:=\sup_{z\in\mathrm{supp}(\rho)}\mathrm{Tr}(\nabla\ell_z(f_{\mathcal{H}})),\quad B_2^*:=\sup_{z\in\mathrm{supp}(\rho)}\mathrm{Tr}(\nabla^2\ell_z(f_{\mathcal{H}})),\quad L:=\|f_{\mathcal{H}}\|_{\mathbf{H}^{-2r}(f_{\mathcal{H}})}.$$

Whenever $0<\lambda\leq\min\{B_2^*,(2LR\log\frac{2}{\delta})^{-1/r},S^{2\alpha}(B_2^*)^\alpha(B_1^*)^{-2\alpha}\}$ with S as in Assumption 3 and $M+N$ is larger than

$$\max\left\{5184\frac{B_2^*}{\lambda}\log\left(\frac{8\cdot 414^2B_2^*}{\lambda\delta}\right),\frac{1296S^2}{L^2\lambda^{1+2r+1/\alpha}}\right\},$$

then a minimizer $\widehat{f}^\lambda$ of Equation 10 satisfies, with probability at least $1-2\delta$,

$$\begin{aligned}\mathcal{R}(\widehat{f}^\lambda)-\mathcal{R}(f_{\mathcal{H}})&\leq\left\|\widehat{f}^\lambda-f_{\mathcal{H}}\right\|^2_{\mathbf{H}(f_{\mathcal{H}})}\leq\left\|\widehat{f}^\lambda-f_{\mathcal{H}}\right\|^2_{\mathbf{H}_\lambda(f_{\mathcal{H}})}\\&\leq 414\frac{S^2}{(M+N)\lambda^{1/\alpha}}\log\left(\frac{2}{\delta}\right)+414L^2\lambda^{1+2r}=:S(M+N,\delta,\lambda)+A(\lambda).\end{aligned}\tag{11}$$

Next we use similar arguments as in Zellinger et al. (2023) to construct an admissible sequence of regularization parameters and associated regularized estimators, which we want to aggregate in the end. The same reasoning as used in the well known Lepskii balancing principle (Lepskii, 1991; Goldenshluger & Pereverzev, 2000; Birgé, 2001; Mathé, 2006; De Vito et al., 2010; Mücke, 2018; Blanchard et al., 2019; Lu et al., 2020; Zellinger et al., 2021) will give us, that the minimal (over all admissible values for λ) risk difference to the target even achieves the optimal rate:

Lemma 2. *Let Assumptions 1– 5 be satisfied and let*

$$\lambda_k := \lambda_0 \cdot \xi^k, k \in \{1, \dots, K\}$$

with $\xi > 1$, $\lambda_0 \leq \xi^{-K} \min\left\{B_2^*, \left(2LR\log\frac{2}{\delta}\right)^{-\frac{1}{r}}, S^{2\alpha}(B_2^*)^{\alpha}(B_1^*)^{-2\alpha}\right\}$, $K < \frac{e^{1296}}{4} - 2$ *and* $\delta \in [\frac{2}{e^{1296}}, \frac{1}{4+2K}]$. *Moreover, let* $\eta = 1296\log^{-1}\left(\frac{2}{\delta}\right)$ *and* λ^* *be the solution of* $\eta S(M+N,\delta,\lambda^*) = A(\lambda^*)$. *Then* $\lambda_1, ..., \lambda_K$ *and* λ^* *satisfy the assumptions of the previous Lemma 1 and we have for* $k = 1, ..., K$*:*

$$\left\|\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\mathbf{H}_{\lambda_k}(f_{\mathcal{H}})} \leq S(M+N,\delta,\lambda_k) + A(\lambda_k). \tag{12}$$

and even more:

$$\min_{1\leq k\leq K}\left\|\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\mathbf{H}_{\lambda_k}(f_{\mathcal{H}})} \leq C^*(S(M+N,\delta,\lambda^*) + A(\lambda^*)) = C^*(M+N)^{-\frac{2r\alpha+\alpha}{2r\alpha+\alpha+1}}. \tag{13}$$

with probability at least $1-(4+2K)\delta$*, for large enough sample size* $M+N$ *greater than*

$$\max\left\{5184\frac{B_2^*}{\lambda_0}\log\left(\frac{8\cdot 414^2 B_2^*}{\lambda_0\delta}\right), \frac{1296S^2}{L^2\lambda_0^{1+2r+1/\alpha}}\right\} \tag{14}$$

equation 14 and $C^* := 16560\xi^{\max(1+2r,1/\alpha)}L^2\left(\frac{1296S^2}{L^2}\right)^{\frac{\alpha+2r\alpha}{1+2\alpha r+\alpha}}$.

Proof. It has been shown in the proof of Zellinger et al. (2023, Theorem 1) that under the given conditions, λ_k for $i = 1, ..., K$ as well as λ^* are admissible according to Lemma 1, which establishes equation 11. Let us now show equation 13. First note that S is a decreasing and A an increasing function with respect to λ, so that λ^* is a minimizer of $S + A$. Due to the structure of $S + A$, it is also clear that there is an index $1 \leq j \leq K$ with $\lambda_j \leq \lambda^* \leq \lambda_{j+1}$ so that

$$\lambda_\dagger := \operatorname*{arg\,min}_{\lambda\in\{\lambda_1,\dots,\lambda_K\}} S(M+N,\delta,\lambda) + A(\lambda) \in \{\lambda_j, \lambda_{j+1}\}.$$

We now distinguish two cases: Let us first consider $\lambda_\dagger = \lambda_j$. Then $\lambda_\dagger \geq \lambda^*\frac{1}{\xi}$ which on the other hand implies $S(\lambda_\dagger) = \frac{B(\delta)}{\lambda_\dagger^{1/\alpha}(M+N)} \leq B(\delta)(\lambda^*)^{-1/\alpha}\xi^{1/\alpha}(M+N)^{-1} = \xi^{1/\alpha}S(\lambda^*)$ (for $B(\delta) = 414S^2\log\left(\frac{2}{\delta}\right)$) and $A(\lambda_\dagger) \leq A(\lambda^*) \leq \xi^{1/\alpha}A(\lambda^*)$.

In the second case we assume $\lambda_\dagger = \lambda_{j+1}$, which yields $\lambda_\dagger \leq \lambda^*\xi$ and therefore $A(\lambda_\dagger) \leq A(\xi\lambda^*) = A(\lambda^*)\xi^{1+2r}$ and also $S(\lambda_\dagger) \leq S(\lambda^*) \leq \xi^{1+2r}S(\lambda^*)$. Thus in both cases: $S(\lambda_\dagger) + A(\lambda_\dagger) \leq \xi^{\max(1+2r,1/\alpha)}(S(\lambda^*) + A(\lambda^*))$, implying equation 13. □

As a next step, let us state some concentration bounds on weighted norms, that relate on the one hand the Hessian norms $\mathbf{H}_\lambda(f)$ evaluated for different $f \in \mathcal{H}$, where on the other hand, the connections between empirical and non-empirical Hessian norms are explored. To do so, we will also need to introduce further notation: we write $\mathbf{B} \preccurlyeq \mathbf{A}$ iff $\mathbf{A} - \mathbf{B}$ is positive semi-definite. Moreover, we define $\mathbf{t}(f) := \sup_{(x,y)\in\operatorname{supp}(\rho)}\sup_{g\in\{yk(x,\cdot)\}}|f\cdot g|$.

Lemma 3 (Marteau-Ferey et al. (2019), Proposition 15)**.** *Let Assumptions 1–5 be satisfied,* $\lambda \geq 0$ *and* $f_1, f_2 \in \mathcal{H}$*. Then, we have*

$$\mathbf{H}_\lambda(f_1) \preccurlyeq e^{\mathbf{t}(f_1-f_2)}\mathbf{H}_\lambda(f_2). \tag{15}$$

Lemma 4 (Marteau-Ferey et al. (2019); Zellinger et al. (2023))**.** *Under the conditions of Lemma 1, we have, with probability at least* $1-2\delta$*,*

$$\mathbf{t}(f_{\mathcal{H}} - f^\lambda) \leq \log(2), \quad \mathbf{t}(f^\lambda - \widehat{f}^\lambda) \leq \log(2) \quad \mathbf{t}(f_{\mathcal{H}} - \widehat{f}^\lambda) \leq 2\log(2). \tag{16}$$

Lemma 5 (Marteau-Ferey et al. (2019), Rudi & Rosasco (2017))**.** *Let the conditions of Lemma 1 be satisfied. Then, it holds, with probability at least* $1-\delta$*,*

$$\mathbf{H}_\lambda(f) \preccurlyeq 2\widehat{\mathbf{H}}_\lambda(f). \tag{17}$$

If, in addition, $0 < \|\mathbf{H}(f)\|_{\mathcal{H}}$*, then it holds, for all*

$$M+N \geq \frac{16B_2^*}{\|\mathbf{H}(f)\|_{\mathcal{H}}}\log\left(\frac{2}{\delta}\right), \tag{18}$$

with probability at least $1-\delta$*,*

$$\widehat{\mathbf{H}}_\lambda(f) \preccurlyeq \frac{3}{2}\mathbf{H}_\lambda(f). \tag{19}$$

We will need another technical assumption in order to apply equation 19 to a sequence of models:

Assumption 6. *For the sequence* $(\lambda_k)_{k=1}^K$ *there exists* $b^* > 0$ *with* $b^* \leq \min_{k\in\{1,\ldots,K\}}\left\|\mathbf{H}(\widehat{f}^{\lambda_k})\right\|_{\mathcal{H}}$.

We are now in the position to formulate and prove a detailed version of our finite sample bounds for aggregation of empirical DR-estimators:

Theorem 2. *Let assumptions 1– 6 be fulfilled. Consider* $K > 1$ *and a sequence* λ_k *of regularization parameters and associated empirical risk minimizers* $\widehat{f}^{\lambda_k}$ *for* $0 \leq k \leq K$*, as defined in Lemma 2. Let moreover* $\delta \in [\frac{2}{e^{1296}}, \frac{1}{9+2K}]$*. Then we have that for* $\widehat{\beta}$ *of Algorithm equation 1 applied with* $\beta_k := g(\widehat{f}^{\lambda_k})$*:*

$$B_F\left(\beta,\widehat{\beta}\right) - B_F(\beta, g(f_{\mathcal{H}})) \leq C(M+N)^{-\frac{2r\alpha+\alpha}{2r\alpha+\alpha+1}}, \tag{20}$$

with probability at least $1-(9+2K)\delta$ *for* $M+N$ *larger than equation 23 and* C *given by equation 22.*

Proof. If the inequalities that we are going to state hold with high probability (under conditions mentioned in the theorem), we will write $\leq_\rho$. Let us start by following Menon & Ong (2016):

$$B_F\left(\beta, g\left(\sum_{k=1}^K \widehat{\alpha_k}\widehat{f}^{\lambda_k}\right)\right) - B_F(\beta, g(f_{\mathcal{H}})) = 2\left(\mathcal{R}\left(\sum_{k=1}^K \widehat{\alpha_k}\widehat{f}^{\lambda_k}\right) - \mathcal{R}\left(f_{\mathcal{H}}\right)\right).$$

Next we apply equation 11, combine it with equation 15 and equation 16 to obtain:

$$B_F\left(\beta, g\left(\sum_{k=1}^K \widehat{\alpha_k}\widehat{f}^{\lambda_k}\right)\right) - B_F(\beta, g(f_{\mathcal{H}})) \underbrace{\leq_\rho}_{(A)} 2\left\|\sum_{k=1}^K \widehat{\alpha_k}\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\mathbf{H}_{\lambda_0}(f_{\mathcal{H}})} \underbrace{\leq_\rho}_{(B)} 8\left\|\sum_{k=1}^K \widehat{\alpha_k}\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\mathbf{H}_{\lambda_0}(\widehat{f}^{\lambda_0})}.$$

An application of equation 17 then yields

$$\begin{aligned}
8\left\|\sum_{k=1}^K \widehat{\alpha_k}\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\mathbf{H}_{\lambda_0}(\widehat{f}^{\lambda_0})} &\underbrace{\leq_\rho}_{(C)} 16\left\|\sum_{k=1}^K \widehat{\alpha_k}\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\widehat{\mathbf{H}}_{\lambda_0}(\widehat{f}^{\lambda_0})} \\
&= 16\min_{\alpha_1,\ldots,\alpha_K\in\mathbb{R}}\left\|\sum_{k=1}^K \alpha_k\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\widehat{\mathbf{H}}_{\lambda_0}(\widehat{f}^{\lambda_0})} \\
&\leq 16\min_{k\in\{1,\ldots,K\}}\left\|\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\widehat{\mathbf{H}}_{\lambda_0}(\widehat{f}^{\lambda_0})},
\end{aligned} \tag{21}$$

where the last inequality follows from equation 5. Next we use that $\lambda_0 < \lambda_k$ to upper-bound equation 21 by:

$$\min_{k\in\{1,\ldots,K\}}\left\|\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\widehat{\mathbf{H}}_{\lambda_0}(\widehat{f}^{\lambda_0})} \leq \min_{k\in\{1,\ldots,K\}}\left\|\widehat{f}^{\lambda_k} - f_{\mathcal{H}}\right\|^2_{\widehat{\mathbf{H}}_{\lambda_k}(\widehat{f}^{\lambda_0})}.$$

We can further apply equation 19, which is valid due to assumption 6, and moreover, we use equation 15 and equation 16, so that we end up with:

$$16 \min_{k\in\{1,\dots,K\}} \left\| \widehat{f}^{\lambda_k} - f_{\mathcal{H}} \right\|^2_{\widehat{\mathbf{H}}_{\lambda_k}(\widehat{f}^{\lambda_0})} \underbrace{\leq_\rho}_{(D)} 24 \min_{k\in\{1,\dots,K\}} \left\| \widehat{f}^{\lambda_k} - f_{\mathcal{H}} \right\|^2_{\mathbf{H}_{\lambda_k}(\widehat{f}^{\lambda_0})} \underbrace{\leq_\rho}_{(E)} 96 \min_{k\in\{1,\dots,K\}} \left\| \widehat{f}^{\lambda_k} - f_{\mathcal{H}} \right\|^2_{\mathbf{H}_{\lambda_k}(f_{\mathcal{H}})}$$
$$\underbrace{\leq_\rho}_{(F)} C(M+N)^{-\frac{2r\alpha+\alpha}{2r\alpha+\alpha+1}},$$

where (F) follows from invoking equation 13 and the constant is given by

$$C := 96C^* = 96 \cdot 16560 \xi^{\max(1+2r,1/\alpha)} L^2 \left(\frac{1296 S^2}{L^2} \right)^{\frac{\alpha+2r\alpha}{1+2\alpha r+\alpha}}, \tag{22}$$

so that equation 20 is established. Note that by Lemma 5, (F) holds with probability $1-(4+2K)\delta$ and (A)-(E) all hold with probability $1-\delta$, so that equation 20 is valid with probability $1-(9+2K)\delta$, yielding the mentioned admissible range for δ. Moreover, taking into account the requirements equation 14 and equation 18, we obtain that $M+N$ needs to be at least as large as

$$\max\left\{ 5184 \frac{B_2^*}{\lambda_0} \log\left(\frac{8\cdot 414^2 B_2^*}{\lambda_0 \delta} \right), \frac{1296 S^2}{L^2 \lambda_0^{1+2r+1/\alpha}}, \frac{16 B_2^*}{b^*} \log\left(\frac{2}{\delta}\right) \right\}. \tag{23}$$

□

B Derivation of Loss and Hessian for LR

Starting with F, g of LR from Example 1 we have

$$F(x) := x\log(x) - (1+x)\log(1+x)$$
$$g(x) := e^x$$

From Theorem 1 from Zellinger (2025) we have

$$\psi^{-1}(x) := \frac{g(x)}{1+g(x)}$$
$$\frac{\psi^{-1}(x)}{1-\psi^{-1}(x)} = e^x$$
$$\gamma(\eta) := -F\left(\frac{\eta}{1-\eta}\right)(1-\eta) = -\left(\frac{\eta}{1-\eta}\log\left(\frac{\eta}{1-\eta}\right) - \left(1+\frac{\eta}{1-\eta}\right)\log\left(1+\frac{\eta}{1-\eta}\right)\right)(1-\eta).$$

Using these terms leads after simplification to

$$\gamma\left(\psi^{-1}(x)\right) = -F\left(e^x\right)\left(1-\psi^{-1}(x)\right) = \frac{x}{1+e^x} + \log\left(1+e^x\right) - x$$
$$\gamma'(\eta) = \frac{\partial}{\partial\eta}\left(-\left(\frac{\eta}{1-\eta}\log\left(\frac{\eta}{1-\eta}\right) - \left(1+\frac{\eta}{1-\eta}\right)\log\left(1+\frac{\eta}{1-\eta}\right)\right)(1-\eta)\right)$$
$$= -\log\left(\frac{\eta}{1-\eta}\right)$$

Taking the result for loss function $\ell(y,x)$ from Zellinger (2025) gives us further after simplifying

$$\ell(-1,x) = \gamma(\psi^{-1}(x)) - \psi^{-1}(x)\gamma'(\psi^{-1}(x))$$
$$= \frac{x}{1+e^x} + \log(1+e^x) - x - \frac{e^x}{1+e^x}(-x) = \log(1+e^x)$$
$$\ell(1,x) = \gamma(\psi^{-1}(x)) + (1-\psi^{-1}(x))\gamma'(\psi^{-1}(x))$$
$$= \frac{x}{1+e^x} + \log(1+e^x) - x + \left(1 - \frac{e^x}{1+e^x}\right)(-x) = \log(1+e^{-x})$$

which is the loss function for LR.. In the next step we derive the Hessian term $h(x, y)$ used in Algorithm 1. Again, now using model $\widehat{f}^{\lambda_0}$ we get after simplification

$$
\begin{aligned}
l'(-1, \widehat{f}^{\lambda_0}(x)) &= \frac{e^{\widehat{f}^{\lambda_0}(x)}}{1 + e^{\widehat{f}^{\lambda_0}(x)}}, \\
l''(-1, \widehat{f}^{\lambda_0}(x)) &= \frac{e^{\widehat{f}^{\lambda_0}(x)}}{(1 + e^{\widehat{f}^{\lambda_0}(x)})^2}, \\
l'(1, \widehat{f}^{\lambda_0}(x)) &= -\frac{1}{1 + e^{\widehat{f}^{\lambda_0}(x)}}, \\
l''(1, \widehat{f}^{\lambda_0}(x)) &= \frac{e^{\widehat{f}^{\lambda_0}(x)}}{(1 + e^{\widehat{f}^{\lambda_0}(x)})^2}.
\end{aligned}
$$

with the second order derivatives being $h(x, y)$. Other loss functions with respective Hessians can be derived analogously.

C Details on Experiments

C.1 Details on Dataset with Known Density Ratios

To evaluate the accuracy of our aggregation algorithm, we build upon the methodologies proposed by Kanamori et al. (2012b); Nguyen et al. (2010). Specifically, we construct high-dimensional data with precisely known density ratios, enabling a systematic analysis of the estimation accuracy. Our approach extends and generalizes the settings used in prior studies. For instance, widely used datasets such as Ringnorm and Twonorm (Breiman, 1996) represent specific cases of Gaussian mixture models, where the mean and covariance parameters are set to predefined values for each mixture component. Similarly, the experiments described in Nguyen et al. (2010) can also be framed within a Gaussian mixture model structure. To introduce greater complexity, we generate distributions by randomly sampling the number, weights, and covariances of the mixture components from 50-dimensional space, which exceeds the dimensionality considered in existing studies. More concretely, the means are sampled uniformly from $[0, 0.5]^{50}$, while the weights of the mixture components are sampled from $[0, 1]$ and subsequently normalized to sum to 1. Each distribution (source and target) is assigned a distinct Gaussian Mixture distribution. In each experiment, the maximum number of mixture components is restricted to four. The number of components for the source distribution, n, is randomly selected from $\{1, 2, 3\}$, while the target distribution gets assigned $4 - n$ components, ensuring variability across datasets.

C.2 Details on Domain Adaptation Datasets

Image data for domain adaptation. The DomainNet-2019 dataset (Peng et al., 2019) includes the six distinct image domains "Real", "Clipart", "Quickdraw", "Sketch", "Painting", and "Infograph". Following the approach of Zellinger et al. (2017) we utilize a simplified version of this dataset known as MiniDomainNet. This reduced version narrows the focus to the five largest classes in the training set across all six domains. To optimize computational efficiency we use a ResNet-18 (He et al., 2016) trained on ImageNet (Krizhevsky et al., 2012). This pre-trained backbone is assumed to have already learned low-level filters effective for the "Real" image domain requiring adaptation only for the remaining five domains resulting in five domain adaptation tasks.

Time-series (sensory) data for domain adaptation. The Heterogeneity Activity Recognition dataset (Stisen et al., 2015) explores the variations specific to sensors, devices, and workloads for human activity recording. It utilizes data collected from 36 different smartphones and smartwatches comprising 13 different device models from four manufacturers. Our experimental setup incorporates all five domain adaptation scenarios analyzed in (Ragab et al., 2023).

Text data for domain adaptation. The Amazon Reviews dataset (Blitzer et al., 2006) contains bag-of-words representations of textual reviews across four categories: books, DVDs, kitchen, and electronics with binary labels indicating the class of review. Each category represents a different domain from which a total of twelve domain adaptation tasks are constructed by pairing each domain as a source domain with every other domain as a target domain.

C.3 Details on Experimental Setup

C.3.1 Methods

To test our aggregation approach we pick four popular representatives of the large class of widely used DRE methods that can be modeled as Bregman divergences as in equation (1). For this we use the methods presented in Example 1: KuLSIF (Kanamori et al., 2009), Exp (Menon & Ong, 2016), SQ (Menon & Ong, 2016), and LR (Bickel et al., 2009) with cross-validation as hyperparameter selection procedure as a baseline and compare this for each of the four methods against our aggregation approach. For KuLSIF, which offers a closed-form solution, no numerical optimization is required. For the other methods, we utilize the CG algorithm from Python's SciPY library (Virtanen et al., 2020), employing the Polak-Ribière line search strategy (Hager & Zhang, 2006). In our domain adaptation experiments, we adhere to the evaluation protocol established by Dinu et al. (2023); Gruber et al. (2024). Specifically, we calculate an ensemble of various deep neural networks using the importance-weighted functional regression approach. We use the density ratio estimates from the aggregated and non-aggregated versions of KuLSIF, Exp, SQ, and LR as importance weights. To comprehensively assess the impact on different domain adaptation techniques, we generate ensemble model candidates for 11 domain adaptation methods. These include Minimum Discrepancy Estimation for Deep Domain Adaptation (MMDA) (Rahman et al., 2020), the Convolutional deep Domanin Adaptation model for Time-Series data (CoDATS) (Wilson et al., 2020), Domain-Adversarial Neural Networks (DANN) (Ganin et al., 2016), Conditional Adversarial Domain Adaptation (CDAN) (Long et al., 2018), Deep Subdomain Adaptation (DSAN) (Zhou et al., 2021), the DIRT-T approach to Unsupervised Domain Adaptation (DIRT) (Shu et al., 2018), Adversarial Spectral Kernel Matching (AdvSKM) (Liu & Xue, 2021), Higher-order Moment Matching (HoMM) (Chen et al., 2020), Deep Domain Confusion (DDC) (Tzeng et al., 2014), Correlation Alignment via Deep Neural Networks (Deep Coral) (Sun et al., 2017), and Central Moment Discrepancy (CMD) (Zellinger et al., 2017). Altogether, this benchmark involved training over 9,000 deep neural network models and evaluating the complete ensembling benchmark proposed by Dinu et al. (2023) for Amazon Reviews, MiniDomainNet, and HHAR. The experiments on these datasets were conducted to compare 8 DRE methods (with and without aggregation).

C.3.2 Known Density Ratios

Building upon Kanamori et al. (2012b) we create ten distinct datasets using Gaussian Mixture distributions in a 50-dimensional space. These distributions feature a varying number of mixture components in $\{1, 2, 3\}$ for either distributions P and Q. Each component gets assigned a unique mean within $[0, 0.5]^{50}$ with corresponding covariance matrix. For all datasets the exact density ratio $\frac{p}{q}$ is known as it is fully determined by the ratio of the Gaussian Mixtures. For each Gaussian mixture dataset 5,000 samples are drawn from the underlying distributions. The selection and evaluation of the baseline DRE methods are conducted using a standard train/validation/test split approach with split ratios 64/16/20 respectively. The corresponding aggregated DRE methods are trained and evaluated on the same splits while the aggregation weights are computed on the respective validation sets. The regularization parameter λ is selected from the range $\{10^{-6}, 10^{-5}, \ldots, 10^{4}\}$, and each experiment is repeated 10 times to ensure statistical robustness. Consistent with Kanamori et al. (2012a), a Gaussian kernel is employed for all density ratio estimation methods, with the kernel width determined using the median heuristic (Schölkopf & Smola, 2002).

C.3.3 Ablation Studies

We additionally add an ablation study as proof of concept that our aggregation algorithm can also be used in a heuristic setting for which we can't prove fast convergence rates as in Theorem 1. For this we train several deep neural networks based logistic regression models (DeepLR) with different hyperparameter settings and optimize objective (4) as done for the other DRE methods. In Table 3 we can see that even in the absence of theoretical guarantees our aggregation approach improves experimental results on all datasets compared to its baseline model that was selected by cross-validation.

	Ablation 1: Deep Learning								
Methods	c3,d1.70	c2,d1.72	c2d1.59	c1d1.55	c2d1.78	c2d1.55	c3d1.57	c2d1.61	c3d1.46
DeepLR	7.280(±0.011)	14.539(±0.010)	13.001(±0.008)	9.963(±0.012)	9.857(±0.013)	10.992(±0.009)	13.421(±0.010)	11.892(±0.011)	9.573(±0.006)
Agg. DeepLR	**6.504**(±**0.012**)	**14.249**(±**0.013**)	**12.733**(±**0.009**)	**9.631**(±**0.014**)	**9.569**(±**0.014**)	**10.428**(±**0.008**)	**12.733**(±**0.011**)	**11.229**(±**0.010**)	**8.954**(±**0.005**)

	Ablation 2: Averaging								
Methods	c3,d1.70	c2,d1.72	c2d1.59	c1d1.55	c2d1.78	c2d1.55	c3d1.57	c2d1.61	c3d1.46
Avg. KuLSIF	8.553(±0.009)	13.151(±0.008)	12.597(±0.007)	11.803(±0.011)	9.533(±0.010)	10.292(±0.009)	12.011(±0.005)	11.524(±0.003)	10.357(±0.006)
Agg. KuLSIF	**8.320**(±**0.004**)	**12.854**(±**0.007**)	**12.422**(±**0.010**)	**11.625**(±**0.004**)	**9.425**(±**0.013**)	**10.001**(±**0.002**)	**12.003**(±**0.007**)	**11.365**(±**0.003**)	**9.421**(±**0.003**)

Table 3: Mean and standard deviation (after $\pm$) 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.

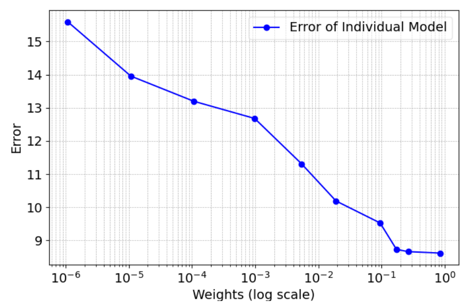

Figure 2: Plot of the value of the aggregation weights against twice the Bregman diveregence error on the geometrically constructed dataset "c3,d1.70". More accurate models get assigned higher weights.

In the second ablation study we compare our method to a naive aggregation approach where model outputs are averaged. Results in Table 3 suggest that averaging model outputs cannot compete with our aggregation algorithm. For the first ablation study we use a deep neural network based logistic regression model trained with binary cross-entropy loss and different regularization parameters in $\{10^{-6}, 10^{-5}, \ldots, 10^{4}\}$. The model is optimized by gradient descent for 125 epochs. We evaluate this model on geometrically-constructed datasets by using the exact same data splits as for the other DRE methods. In the second ablation study we compare our aggregation algorithm to a model averaging approach. We use the exact same experimental setup as for our algorithm and replace aggregation by averaging model outputs on geometrically-constructed datasets. In Figure 2 we illustrate the value of the weights assigned to KuLSIF estimators with different regularization parameter settings evaluated on dataset "c3,d1.70". It can be seen that more accurate models get assigned higher weight values.

C.3.4 Domain Adaptation

The results of the domain adaptation benchmark experiment were computed using gradient-based training across a total of 9,174 models. The implementation of certain components is based on the codebase by Dinu et al. (2023); Gruber et al. (2024). The specifics of the experimental setup are as follows:

- MiniDomainNet (image data): 11 methods $\times$ 8 parameters $\times$ 5 domain adaptation tasks $\times$ 3 seeds $=$ 1320 trained models
- AmazonReviews (text data): 11 methods $\times$ 14 parameters $\times$ 12 domain adaptation tasks $\times$ 3 seeds $=$ 5544 trained models
- HHAR (sensory data): 11 methods $\times$ 14 parameters $\times$ 5 domain adaptation tasks $\times$ 3 seeds $=$ 2310 trained models

Consistent with Dinu et al. (2023), we utilized 11 domain adaptation methods from the AdaTime benchmark (Ragab et al., 2023). For each method and domain adaptation modality (text, image, sensory), we evaluated all 8 density ratio estimation approaches across 22 domain adaptation scenarios. The model implementation for domain adaptation methods and experimental configurations also followed Dinu et al. (2023). Specifically, fully connected networks were used for Amazon Reviews, while a pretrained ResNet-18 backbone was employed for MiniDomainNet. For training and selecting/aggregating 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. For the regularization parameter λ we use $\{10^{-6}, 10^{-5}, \ldots, 10^{4}\}$ as hyperparameter space. Following Kanamori et al. (2012a), we use a Gaussian kernel with kernel width set according to the median heuristic (Schölkopf & Smola, 2002) for all DRE methods. In the results tables, we report the classification accuracy on the respective test sets of the target distribution for all compared DRE methods.

C.4 Detailed Experimental Results for Domain Adaptation

	Domain Adaptation: Amazon Reviews							
	Cross-Validation for Binary Classifier				**Aggregation**			
DA-Method	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
MMDA	0.786(±0.010)	0.771(±0.009)	0.784(±0.010)	0.780(±0.008)	**0.794(±0.008)**	**0.782(±0.008)**	**0.790(±0.010)**	**0.789(±0.006)**
CoDATS	0.795(±0.012)	0.777(±0.011)	0.794(±0.012)	0.788(±0.009)	**0.803(±0.010)**	**0.790(±0.010)**	**0.797(±0.012)**	**0.793(±0.008)**
DANN	0.794(±0.011)	0.779(±0.011)	0.793(±0.012)	0.790(±0.010)	**0.804(±0.009)**	**0.792(±0.010)**	**0.800(±0.012)**	**0.799(±0.008)**
CDAN	0.787(±0.012)	0.772(±0.010)	0.787(±0.010)	0.785(±0.009)	**0.792(±0.010)**	**0.783(±0.009)**	**0.790(±0.010)**	**0.792(±0.007)**
DSAN	0.794(±0.011)	0.778(±0.012)	0.793(±0.013)	0.790(±0.011)	**0.803(±0.009)**	**0.790(±0.011)**	**0.798(±0.013)**	**0.798(±0.009)**
DIRT	0.787(±0.011)	0.776(±0.011)	0.788(±0.012)	0.784(±0.010)	**0.797(±0.009)**	**0.787(±0.010)**	**0.790(±0.012)**	**0.794(±0.008)**
AdvSKM	0.779(±0.011)	0.761(±0.009)	0.777(±0.010)	0.772(±0.008)	**0.784(±0.009)**	**0.774(±0.008)**	**0.781(±0.010)**	**0.782(±0.006)**
HoMM	0.777(±0.009)	0.760(±0.010)	0.774(±0.011)	0.770(±0.009)	**0.786(±0.007)**	**0.772(±0.009)**	**0.779(±0.011)**	**0.779(±0.008)**
DDC	0.780(±0.011)	0.764(±0.011)	0.778(±0.012)	0.774(±0.010)	**0.787(±0.009)**	**0.777(±0.010)**	**0.782(±0.012)**	**0.783(±0.008)**
DeepCoral	0.784(±0.010)	0.767(±0.010)	0.781(±0.011)	0.776(±0.009)	**0.794(±0.008)**	**0.779(±0.009)**	**0.785(±0.011)**	**0.784(±0.007)**
CMD	0.790(±0.013)	0.773(±0.011)	0.786(±0.012)	0.781(±0.011)	**0.797(±0.011)**	**0.783(±0.011)**	**0.790(±0.012)**	**0.787(±0.008)**
Avg.	0.787(±0.011)	0.771(±0.010)	0.785(±0.011)	0.781(±0.009)	**0.795(±0.009)**	**0.783(±0.009)**	**0.789(±0.011)**	**0.789(±0.007)**

	Domain Adaptation: HHAR							
	Cross-Validation for Binary Classifier				**Aggregation**			
DA-Method	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
MMDA	0.780(±0.008)	0.670(±0.098)	0.773(±0.013)	0.736(±0.006)	**0.826(±0.023)**	**0.711(±0.085)**	**0.786(±0.007)**	**0.763(±0.005)**
CoDATS	0.723(±0.130)	0.776(±0.021)	0.779(±0.021)	0.741(±0.011)	**0.765(±0.120)**	**0.818(±0.016)**	**0.793(±0.015)**	**0.767(±0.010)**
DANN	0.697(±0.179)	0.785(±0.024)	0.795(±0.021)	0.757(±0.011)	**0.738(±0.170)**	**0.828(±0.015)**	**0.807(±0.015)**	**0.780(±0.010)**
CDAN	0.792(±0.029)	0.706(±0.166)	0.788(±0.029)	0.751(±0.022)	**0.834(±0.031)**	**0.750(±0.144)**	**0.804(±0.024)**	**0.775(±0.021)**
DSAN	0.754(±0.128)	0.528(±0.228)	0.792(±0.015)	0.754(±0.014)	**0.794(±0.133)**	**0.572(±0.219)**	**0.805(±0.013)**	**0.775(±0.013)**
DIRT	0.731(±0.084)	0.790(±0.040)	0.790(±0.023)	0.753(±0.014)	**0.774(±0.086)**	**0.828(±0.023)**	**0.801(±0.017)**	**0.779(±0.013)**
AdvSKM	0.752(±0.007)	0.746(±0.013)	0.746(±0.018)	0.708(±0.008)	**0.793(±0.024)**	**0.790(±0.019)**	**0.759(±0.012)**	**0.735(±0.006)**
HoMM	0.759(±0.009)	0.662(±0.096)	0.754(±0.021)	0.716(±0.011)	**0.803(±0.022)**	**0.707(±0.083)**	**0.768(±0.016)**	**0.749(±0.010)**
DDC	0.748(±0.017)	0.454(±0.215)	0.744(±0.019)	0.706(±0.009)	**0.794(±0.018)**	**0.496(±0.196)**	**0.759(±0.013)**	**0.732(±0.009)**
DeepCoral	0.701(±0.115)	0.749(±0.021)	0.758(±0.015)	0.720(±0.006)	**0.745(±0.118)**	**0.791(±0.013)**	**0.769(±0.011)**	**0.743(±0.005)**
CMD	0.671(±0.170)	0.770(±0.015)	0.770(±0.016)	0.732(±0.008)	**0.714(±0.164)**	**0.814(±0.017)**	**0.780(±0.010)**	**0.754(±0.007)**
Avg.	0.737(±0.080)	0.694(±0.085)	0.772(±0.019)	0.734(±0.011)	**0.780(±0.082)**	**0.737(±0.075)**	**0.785(±0.014)**	**0.759(±0.010)**

Table 4: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet, Amazon Reviews and HHAR datasets over three different random initialization of model weights and several domain adaptation tasks.

	Cross-Validation for Binary Classifier				**Aggregation**			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R → C	0.585(±0.013)	0.590(±0.016)	0.589(±0.024)	0.578(±0.022)	**0.595(±0.011)**	**0.598(±0.015)**	**0.596(±0.017)**	**0.592(±0.021)**
R → I	0.377(±0.009)	0.372(±0.008)	0.378(±0.007)	0.365(±0.005)	**0.393(±0.007)**	**0.385(±0.008)**	**0.394(±0.001)**	**0.384(±0.004)**
R → P	0.722(±0.004)	0.729(±0.006)	**0.725(±0.006)**	0.723(±0.004)	**0.730(±0.002)**	**0.743(±0.006)**	0.721(±0.000)	**0.728(±0.003)**
R → Q	0.354(±0.008)	0.348(±0.016)	0.349(±0.014)	0.328(±0.013)	**0.356(±0.006)**	**0.356(±0.016)**	**0.358(±0.008)**	**0.351(±0.012)**
R → S	0.596(±0.009)	0.599(±0.010)	0.600(±0.009)	0.596(±0.007)	**0.606(±0.007)**	**0.611(±0.009)**	**0.610(±0.002)**	**0.618(±0.006)**
Avg.	0.527(±0.009)	0.528(±0.011)	0.528(±0.012)	0.518(±0.010)	**0.536(±0.007)**	**0.539(±0.011)**	**0.536(±0.006)**	**0.535(±0.009)**

Table 5: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet computed with domain adaptation method MMDA.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
R → C	0.597(±0.011)	0.577(±0.010)	0.594(±0.010)	0.576(±0.008)	**0.602(±0.009)**	**0.587(±0.008)**	**0.601(±0.005)**	**0.593(±0.006)**
R → I	0.366(±0.012)	0.368(±0.023)	0.349(±0.038)	0.338(±0.036)	**0.373(±0.011)**	**0.385(±0.023)**	**0.357(±0.031)**	**0.356(±0.035)**
R → P	0.739(±0.015)	0.733(±0.018)	0.737(±0.017)	0.726(±0.015)	**0.747(±0.014)**	**0.745(±0.017)**	**0.750(±0.011)**	**0.742(±0.014)**
R → Q	0.361(±0.013)	0.364(±0.014)	0.355(±0.027)	0.342(±0.025)	**0.370(±0.010)**	**0.377(±0.013)**	**0.369(±0.021)**	**0.364(±0.023)**
R → S	**0.619(±0.008)**	0.616(±0.008)	0.616(±0.009)	0.601(±0.008)	0.617(±0.006)	**0.624(±0.008)**	**0.622(±0.002)**	**0.607(±0.008)**
Avg.	0.536(±0.012)	0.532(±0.015)	0.530(±0.020)	0.517(±0.018)	**0.542(±0.010)**	**0.543(±0.014)**	**0.540(±0.014)**	**0.533(±0.017)**

Table 6: Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet computed with domain adaptation method CoDATS.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
R → C	0.591(±0.015)	0.576(±0.013)	0.579(±0.012)	0.560(±0.009)	**0.597(±0.013)**	**0.583(±0.011)**	**0.587(±0.005)**	**0.568(±0.007)**
R → I	0.374(±0.009)	0.372(±0.011)	0.369(±0.026)	0.354(±0.024)	**0.379(±0.007)**	**0.387(±0.009)**	**0.375(±0.021)**	**0.368(±0.022)**
R → P	0.720(±0.007)	0.716(±0.008)	0.699(±0.032)	0.692(±0.030)	**0.724(±0.006)**	**0.727(±0.008)**	**0.702(±0.026)**	**0.699(±0.029)**
R → Q	0.357(±0.009)	0.340(±0.016)	0.337(±0.016)	0.322(±0.014)	**0.362(±0.005)**	**0.355(±0.015)**	**0.348(±0.009)**	**0.326(±0.013)**
R → S	0.613(±0.007)	0.607(±0.010)	0.614(±0.007)	0.602(±0.005)	**0.619(±0.005)**	**0.624(±0.010)**	**0.621(±0.001)**	**0.632(±0.004)**
Avg.	0.531(±0.009)	0.522(±0.012)	0.520(±0.019)	0.506(±0.016)	**0.536(±0.007)**	**0.535(±0.011)**	**0.526(±0.013)**	**0.519(±0.015)**

Table 7: Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet computed with domain adaptation method DANN.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
R → C	0.592(±0.018)	0.595(±0.029)	0.594(±0.023)	0.578(±0.021)	**0.594(±0.015)**	**0.616(±0.028)**	**0.608(±0.017)**	**0.598(±0.019)**
R → I	0.372(±0.013)	0.381(±0.012)	0.337(±0.056)	0.324(±0.053)	**0.384(±0.011)**	**0.384(±0.011)**	**0.346(±0.050)**	**0.343(±0.052)**
R → P	0.729(±0.004)	0.720(±0.015)	0.725(±0.011)	0.713(±0.009)	**0.735(±0.001)**	**0.734(±0.014)**	**0.731(±0.005)**	**0.722(±0.008)**
R → Q	0.353(±0.018)	0.352(±0.020)	0.354(±0.017)	0.351(±0.014)	**0.354(±0.014)**	**0.369(±0.020)**	**0.364(±0.011)**	**0.363(±0.012)**
R → S	0.609(±0.008)	0.607(±0.012)	0.611(±0.009)	0.594(±0.007)	**0.618(±0.005)**	**0.618(±0.010)**	**0.623(±0.003)**	**0.603(±0.006)**
Avg.	0.531(±0.012)	0.531(±0.017)	0.524(±0.023)	0.512(±0.021)	**0.537(±0.009)**	**0.544(±0.017)**	**0.535(±0.017)**	**0.526(±0.020)**

Table 8: Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet computed with domain adaptation method CDAN.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
R → C	**0.623(±0.015)**	0.603(±0.026)	0.596(±0.019)	0.589(±0.017)	0.622(±0.013)	**0.614(±0.026)**	**0.606(±0.013)**	**0.597(±0.016)**
R → I	0.374(±0.010)	0.367(±0.008)	0.350(±0.014)	0.339(±0.012)	**0.383(±0.008)**	**0.391(±0.006)**	**0.363(±0.010)**	**0.344(±0.011)**
R → P	0.720(±0.007)	0.724(±0.008)	0.720(±0.008)	0.708(±0.005)	**0.726(±0.004)**	**0.738(±0.007)**	**0.729(±0.001)**	**0.723(±0.004)**
R → Q	0.364(±0.012)	0.355(±0.018)	0.354(±0.018)	0.334(±0.016)	**0.366(±0.010)**	**0.364(±0.016)**	**0.359(±0.012)**	**0.347(±0.015)**
R → S	0.616(±0.012)	0.609(±0.015)	0.615(±0.016)	0.598(±0.014)	**0.621(±0.011)**	**0.622(±0.014)**	**0.618(±0.010)**	**0.612(±0.013)**
Avg.	0.539(±0.011)	0.532(±0.015)	0.527(±0.015)	0.513(±0.013)	**0.544(±0.009)**	**0.546(±0.014)**	**0.535(±0.009)**	**0.525(±0.012)**

Table 9: Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet computed with domain adaptation method DSAN.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
R → C	0.572(±0.021)	0.474(±0.176)	0.566(±0.022)	0.555(±0.020)	**0.580(±0.019)**	**0.478(±0.175)**	**0.582(±0.016)**	**0.575(±0.018)**
R → I	0.376(±0.047)	0.236(±0.135)	0.406(±0.025)	0.391(±0.022)	**0.387(±0.045)**	**0.246(±0.133)**	**0.407(±0.018)**	**0.402(±0.022)**
R → P	0.714(±0.026)	0.542(±0.271)	**0.713(±0.025)**	0.700(±0.024)	**0.718(±0.024)**	**0.550(±0.270)**	0.711(±0.020)	**0.724(±0.023)**
R → Q	0.336(±0.018)	0.248(±0.055)	0.329(±0.021)	0.315(±0.019)	**0.340(±0.016)**	**0.258(±0.054)**	**0.338(±0.016)**	**0.325(±0.018)**
R → S	0.586(±0.020)	0.431(±0.245)	0.587(±0.021)	0.585(±0.019)	**0.591(±0.019)**	**0.441(±0.244)**	**0.595(±0.015)**	**0.598(±0.018)**
Avg.	0.517(±0.026)	0.386(±0.177)	0.520(±0.023)	0.509(±0.021)	**0.523(±0.025)**	**0.395(±0.175)**	**0.526(±0.017)**	**0.525(±0.020)**

Table 10: Mean and standard deviation (after ±) of target classification accuracy on MiniDomainNet computed with domain adaptation method DIRT.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R $\rightarrow$ C	0.568($\pm$0.008)	0.561($\pm$0.013)	0.552($\pm$0.006)	0.535($\pm$0.004)	**0.580($\pm$0.006)**	**0.573($\pm$0.012)**	**0.564($\pm$0.000)**	**0.548($\pm$0.003)**
R $\rightarrow$ I	**0.382($\pm$0.010)**	0.389($\pm$0.007)	**0.385($\pm$0.026)**	0.369($\pm$0.024)	0.381($\pm$0.008)	**0.415($\pm$0.006)**	0.384($\pm$0.019)	**0.384($\pm$0.023)**
R $\rightarrow$ P	0.717($\pm$0.005)	0.712($\pm$0.014)	0.712($\pm$0.010)	0.705($\pm$0.008)	0.717($\pm$0.004)	**0.725($\pm$0.013)**	**0.724($\pm$0.004)**	**0.723($\pm$0.006)**
R $\rightarrow$ Q	0.332($\pm$0.005)	0.325($\pm$0.007)	0.324($\pm$0.011)	0.315($\pm$0.008)	**0.342($\pm$0.003)**	**0.329($\pm$0.006)**	**0.331($\pm$0.004)**	**0.332($\pm$0.007)**
R $\rightarrow$ S	0.584($\pm$0.004)	0.586($\pm$0.003)	0.588($\pm$0.005)	0.575($\pm$0.004)	**0.587($\pm$0.003)**	**0.602($\pm$0.002)**	**0.602($\pm$0.001)**	**0.598($\pm$0.002)**
Avg.	0.516($\pm$0.006)	0.515($\pm$0.009)	0.512($\pm$0.012)	0.500($\pm$0.010)	**0.522($\pm$0.005)**	**0.529($\pm$0.008)**	**0.521($\pm$0.006)**	**0.517($\pm$0.008)**

Table 11: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet computed with domain adaptation method AdvSKM.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R $\rightarrow$ C	0.590($\pm$0.020)	0.588($\pm$0.020)	0.577($\pm$0.020)	0.567($\pm$0.018)	**0.604($\pm$0.018)**	**0.604($\pm$0.019)**	**0.589($\pm$0.013)**	**0.591($\pm$0.017)**
R $\rightarrow$ I	0.399($\pm$0.006)	0.407($\pm$0.009)	0.378($\pm$0.025)	0.365($\pm$0.024)	**0.400($\pm$0.004)**	**0.419($\pm$0.008)**	**0.386($\pm$0.018)**	**0.385($\pm$0.023)**
R $\rightarrow$ P	0.729($\pm$0.002)	0.726($\pm$0.009)	0.726($\pm$0.010)	0.704($\pm$0.008)	**0.736($\pm$0.000)**	**0.744($\pm$0.008)**	**0.737($\pm$0.004)**	**0.725($\pm$0.007)**
R $\rightarrow$ Q	0.351($\pm$0.005)	0.333($\pm$0.013)	0.333($\pm$0.013)	0.318($\pm$0.012)	**0.356($\pm$0.003)**	**0.345($\pm$0.011)**	**0.341($\pm$0.008)**	**0.345($\pm$0.011)**
R $\rightarrow$ S	0.588($\pm$0.008)	0.593($\pm$0.011)	0.592($\pm$0.008)	0.572($\pm$0.005)	**0.599($\pm$0.007)**	**0.610($\pm$0.009)**	**0.594($\pm$0.002)**	**0.595($\pm$0.003)**
Avg.	0.531($\pm$0.008)	0.529($\pm$0.012)	0.521($\pm$0.015)	0.505($\pm$0.013)	**0.539($\pm$0.007)**	**0.544($\pm$0.011)**	**0.529($\pm$0.009)**	**0.528($\pm$0.012)**

Table 12: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet computed with domain adaptation method HoMM.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R $\rightarrow$ C	0.569($\pm$0.027)	0.571($\pm$0.029)	0.563($\pm$0.028)	0.555($\pm$0.027)	**0.584($\pm$0.025)**	**0.590($\pm$0.029)**	**0.577($\pm$0.023)**	**0.573($\pm$0.027)**
R $\rightarrow$ I	0.389($\pm$0.011)	0.391($\pm$0.016)	0.390($\pm$0.011)	0.369($\pm$0.010)	**0.394($\pm$0.009)**	**0.399($\pm$0.015)**	**0.398($\pm$0.006)**	**0.388($\pm$0.009)**
R $\rightarrow$ P	0.717($\pm$0.002)	0.721($\pm$0.003)	0.715($\pm$0.002)	0.701($\pm$0.000)	**0.727($\pm$0.000)**	**0.726($\pm$0.002)**	**0.723($\pm$0.003)**	**0.711($\pm$0.000)**
R $\rightarrow$ Q	0.333($\pm$0.004)	0.323($\pm$0.013)	0.324($\pm$0.013)	0.313($\pm$0.010)	**0.337($\pm$0.002)**	**0.335($\pm$0.012)**	**0.336($\pm$0.006)**	**0.331($\pm$0.009)**
R $\rightarrow$ S	0.580($\pm$0.005)	0.578($\pm$0.005)	0.579($\pm$0.004)	0.562($\pm$0.003)	**0.594($\pm$0.003)**	**0.595($\pm$0.004)**	**0.586($\pm$0.002)**	**0.574($\pm$0.002)**
Avg.	0.517($\pm$0.010)	0.517($\pm$0.013)	0.514($\pm$0.012)	0.500($\pm$0.010)	**0.527($\pm$0.008)**	**0.529($\pm$0.012)**	**0.524($\pm$0.008)**	**0.515($\pm$0.009)**

Table 13: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet computed with domain adaptation method DDC.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R $\rightarrow$ C	0.588($\pm$0.021)	0.569($\pm$0.016)	0.567($\pm$0.012)	0.543($\pm$0.010)	**0.593($\pm$0.020)**	**0.578($\pm$0.015)**	**0.578($\pm$0.006)**	**0.548($\pm$0.009)**
R $\rightarrow$ I	0.369($\pm$0.012)	0.372($\pm$0.008)	**0.381($\pm$0.018)**	0.370($\pm$0.017)	**0.371($\pm$0.010)**	**0.387($\pm$0.008)**	0.380($\pm$0.013)	**0.379($\pm$0.016)**
R $\rightarrow$ P	0.735($\pm$0.011)	0.737($\pm$0.010)	0.734($\pm$0.005)	0.722($\pm$0.004)	**0.749($\pm$0.010)**	**0.749($\pm$0.010)**	**0.737($\pm$0.001)**	**0.732($\pm$0.003)**
R $\rightarrow$ Q	0.363($\pm$0.011)	0.348($\pm$0.021)	0.349($\pm$0.016)	0.334($\pm$0.014)	0.363($\pm$0.008)	**0.359($\pm$0.020)**	**0.365($\pm$0.011)**	**0.355($\pm$0.013)**
R $\rightarrow$ S	0.621($\pm$0.006)	0.616($\pm$0.009)	0.617($\pm$0.009)	0.604($\pm$0.008)	**0.637($\pm$0.003)**	**0.629($\pm$0.008)**	**0.619($\pm$0.003)**	**0.620($\pm$0.006)**
Avg.	0.535($\pm$0.012)	0.528($\pm$0.013)	0.530($\pm$0.012)	0.514($\pm$0.011)	**0.543($\pm$0.010)**	**0.540($\pm$0.012)**	**0.536($\pm$0.007)**	**0.527($\pm$0.009)**

Table 14: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet computed with domain adaptation method DeepCoral.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
R $\rightarrow$ C	0.576($\pm$0.015)	0.560($\pm$0.026)	0.552($\pm$0.017)	0.538($\pm$0.015)	**0.585($\pm$0.013)**	**0.574($\pm$0.025)**	**0.560($\pm$0.011)**	**0.541($\pm$0.014)**
R $\rightarrow$ I	0.383($\pm$0.009)	0.391($\pm$0.021)	0.381($\pm$0.033)	0.375($\pm$0.032)	**0.386($\pm$0.008)**	**0.400($\pm$0.019)**	**0.384($\pm$0.026)**	**0.397($\pm$0.031)**
R $\rightarrow$ P	0.739($\pm$0.008)	0.739($\pm$0.016)	0.742($\pm$0.011)	0.723($\pm$0.008)	**0.753($\pm$0.006)**	**0.747($\pm$0.016)**	**0.747($\pm$0.005)**	**0.732($\pm$0.006)**
R $\rightarrow$ Q	0.354($\pm$0.013)	0.336($\pm$0.030)	0.335($\pm$0.028)	0.321($\pm$0.026)	**0.362($\pm$0.011)**	**0.352($\pm$0.029)**	**0.341($\pm$0.022)**	**0.333($\pm$0.025)**
R $\rightarrow$ S	**0.595($\pm$0.005)**	0.594($\pm$0.010)	0.597($\pm$0.014)	0.590($\pm$0.013)	0.594($\pm$0.004)	**0.617($\pm$0.010)**	**0.603($\pm$0.009)**	**0.605($\pm$0.013)**
Avg.	0.529($\pm$0.010)	0.524($\pm$0.021)	0.521($\pm$0.021)	0.510($\pm$0.019)	**0.536($\pm$0.008)**	**0.538($\pm$0.020)**	**0.527($\pm$0.015)**	**0.521($\pm$0.018)**

Table 15: Mean and standard deviation (after $\pm$) of target classification accuracy on MiniDomainNet computed with domain adaptation method CMD.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
B → D	0.795(±0.003)	0.784(±0.002)	0.796(±0.003)	0.792(±0.001)	**0.799(±0.001)**	**0.794(±0.002)**	**0.802(±0.003)**	**0.796(±0.000)**
B → E	0.776(±0.018)	0.752(±0.015)	0.766(±0.016)	0.762(±0.014)	**0.784(±0.016)**	**0.764(±0.015)**	**0.770(±0.016)**	**0.773(±0.012)**
B → K	0.793(±0.019)	0.791(±0.020)	0.794(±0.021)	0.792(±0.019)	**0.806(±0.017)**	**0.797(±0.018)**	**0.804(±0.021)**	**0.800(±0.017)**
D → B	0.799(±0.007)	0.770(±0.005)	**0.793(±0.006)**	0.794(±0.004)	**0.805(±0.004)**	**0.790(±0.004)**	0.788(±0.006)	**0.805(±0.001)**
D → E	0.791(±0.008)	**0.785(±0.010)**	0.790(±0.011)	0.792(±0.010)	**0.797(±0.007)**	0.783(±0.010)	**0.800(±0.011)**	**0.805(±0.008)**
D → K	0.802(±0.010)	0.791(±0.008)	0.801(±0.009)	0.797(±0.007)	**0.810(±0.008)**	**0.803(±0.007)**	**0.808(±0.009)**	**0.812(±0.005)**
E → B	0.709(±0.015)	0.693(±0.010)	**0.709(±0.011)**	0.705(±0.010)	**0.720(±0.014)**	**0.708(±0.009)**	0.708(±0.011)	**0.718(±0.007)**
E → D	0.746(±0.003)	0.729(±0.008)	0.737(±0.009)	0.727(±0.006)	**0.761(±0.002)**	**0.731(±0.007)**	**0.744(±0.009)**	**0.731(±0.004)**
E → K	0.878(±0.009)	0.867(±0.008)	0.876(±0.008)	0.881(±0.005)	**0.883(±0.006)**	**0.878(±0.007)**	**0.879(±0.008)**	**0.896(±0.004)**
K → B	0.729(±0.005)	0.723(±0.005)	0.732(±0.006)	0.727(±0.004)	**0.740(±0.003)**	**0.734(±0.003)**	**0.742(±0.006)**	0.727(±0.002)
K → D	0.757(±0.015)	0.740(±0.011)	0.756(±0.012)	0.750(±0.010)	**0.767(±0.013)**	**0.756(±0.011)**	**0.766(±0.012)**	**0.758(±0.008)**
K → E	**0.857(±0.008)**	0.832(±0.007)	0.858(±0.007)	0.848(±0.005)	0.855(±0.005)	**0.850(±0.005)**	**0.865(±0.007)**	**0.854(±0.004)**
Avg.	0.786(±0.010)	0.771(±0.009)	0.784(±0.010)	0.780(±0.008)	**0.794(±0.008)**	**0.782(±0.008)**	**0.790(±0.010)**	**0.789(±0.006)**

Table 16: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method MMDA.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
B → D	0.804(±0.008)	0.778(±0.006)	**0.806(±0.007)**	0.796(±0.004)	**0.807(±0.006)**	**0.787(±0.004)**	0.804(±0.007)	**0.803(±0.002)**
B → E	0.785(±0.012)	0.763(±0.009)	**0.782(±0.010)**	0.773(±0.008)	**0.789(±0.011)**	**0.777(±0.007)**	0.777(±0.010)	**0.777(±0.005)**
B → K	0.807(±0.009)	0.784(±0.005)	0.807(±0.007)	0.797(±0.005)	**0.809(±0.006)**	**0.799(±0.004)**	**0.811(±0.007)**	**0.806(±0.003)**
D → B	0.801(±0.005)	0.790(±0.004)	0.802(±0.005)	0.807(±0.003)	**0.817(±0.003)**	**0.798(±0.003)**	**0.806(±0.005)**	**0.812(±0.001)**
D → E	0.806(±0.011)	0.787(±0.015)	0.804(±0.015)	0.797(±0.013)	**0.815(±0.009)**	**0.797(±0.014)**	**0.808(±0.015)**	**0.805(±0.010)**
D → K	0.819(±0.012)	0.801(±0.014)	0.819(±0.016)	0.810(±0.013)	**0.830(±0.010)**	**0.824(±0.013)**	**0.826(±0.016)**	**0.812(±0.012)**
E → B	0.719(±0.025)	0.704(±0.031)	0.721(±0.031)	0.717(±0.030)	**0.732(±0.024)**	**0.716(±0.030)**	**0.723(±0.031)**	**0.718(±0.029)**
E → D	0.748(±0.024)	0.729(±0.017)	0.743(±0.017)	0.743(±0.015)	**0.755(±0.022)**	**0.737(±0.016)**	**0.750(±0.017)**	**0.748(±0.013)**
E → K	0.883(±0.012)	0.868(±0.008)	0.881(±0.009)	0.874(±0.007)	**0.885(±0.010)**	**0.880(±0.008)**	**0.888(±0.009)**	**0.877(±0.005)**
K → B	0.741(±0.008)	0.730(±0.007)	0.734(±0.008)	0.730(±0.006)	**0.747(±0.007)**	**0.735(±0.005)**	**0.737(±0.008)**	**0.736(±0.004)**
K → D	0.766(±0.007)	0.745(±0.006)	0.762(±0.007)	0.760(±0.004)	**0.773(±0.005)**	**0.762(±0.006)**	**0.765(±0.007)**	**0.766(±0.002)**
K → E	0.866(±0.008)	0.849(±0.006)	0.865(±0.007)	0.855(±0.005)	**0.881(±0.006)**	**0.863(±0.005)**	**0.873(±0.007)**	**0.862(±0.004)**
Avg.	0.795(±0.012)	0.777(±0.011)	0.794(±0.012)	0.788(±0.009)	**0.803(±0.010)**	**0.790(±0.010)**	**0.797(±0.012)**	**0.793(±0.008)**

Table 17: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method CoDATS.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
B → D	0.802(±0.006)	0.793(±0.008)	**0.802(±0.010)**	0.805(±0.007)	**0.814(±0.004)**	**0.812(±0.007)**	0.800(±0.010)	**0.818(±0.005)**
B → E	0.784(±0.012)	0.743(±0.011)	0.763(±0.013)	0.755(±0.010)	**0.793(±0.010)**	**0.749(±0.010)**	**0.770(±0.013)**	**0.762(±0.009)**
B → K	0.801(±0.009)	0.783(±0.007)	0.803(±0.008)	0.802(±0.005)	**0.812(±0.007)**	**0.795(±0.006)**	**0.813(±0.008)**	**0.818(±0.004)**
D → B	0.804(±0.003)	0.788(±0.002)	0.799(±0.003)	0.798(±0.001)	**0.812(±0.001)**	**0.797(±0.001)**	**0.802(±0.003)**	**0.805(±0.001)**
D → E	0.808(±0.007)	0.787(±0.008)	**0.811(±0.009)**	0.807(±0.007)	**0.814(±0.004)**	**0.801(±0.007)**	0.810(±0.009)	**0.819(±0.005)**
D → K	0.814(±0.013)	0.806(±0.017)	0.814(±0.018)	0.813(±0.015)	**0.822(±0.011)**	**0.818(±0.016)**	**0.822(±0.018)**	**0.818(±0.013)**
E → B	0.713(±0.017)	0.706(±0.014)	0.717(±0.014)	0.710(±0.012)	**0.724(±0.015)**	**0.722(±0.012)**	**0.727(±0.014)**	**0.725(±0.010)**
E → D	0.748(±0.009)	0.733(±0.005)	0.748(±0.006)	0.751(±0.005)	**0.758(±0.007)**	**0.752(±0.003)**	**0.758(±0.006)**	**0.758(±0.003)**
E → K	0.878(±0.015)	0.866(±0.013)	0.874(±0.015)	0.871(±0.012)	**0.894(±0.013)**	**0.877(±0.012)**	**0.884(±0.015)**	**0.882(±0.010)**
K → B	0.739(±0.014)	0.739(±0.003)	0.750(±0.003)	0.740(±0.002)	**0.749(±0.012)**	**0.745(±0.002)**	**0.760(±0.003)**	**0.747(±0.001)**
K → D	0.770(±0.020)	0.752(±0.037)	0.766(±0.039)	0.760(±0.036)	**0.782(±0.019)**	**0.767(±0.036)**	**0.776(±0.039)**	**0.764(±0.033)**
K → E	0.863(±0.011)	0.853(±0.007)	0.865(±0.007)	0.870(±0.005)	**0.879(±0.008)**	**0.864(±0.006)**	**0.874(±0.007)**	**0.878(±0.003)**
Avg.	0.794(±0.011)	0.779(±0.011)	0.793(±0.012)	0.790(±0.010)	**0.804(±0.009)**	**0.792(±0.010)**	**0.800(±0.012)**	**0.799(±0.008)**

Table 18: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method DANN.

	Cross-Validation for Binary Classifier				Aggregation			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.802(±0.009)	0.786(±0.010)	0.800(±0.011)	0.800(±0.008)	**0.813(±0.007)**	**0.793(±0.009)**	**0.808(±0.011)**	**0.805(±0.007)**
B → E	**0.779(±0.009)**	0.759(±0.007)	0.777(±0.007)	0.777(±0.004)	0.778(±0.006)	**0.770(±0.006)**	**0.779(±0.007)**	**0.784(±0.001)**
B → K	0.797(±0.007)	0.776(±0.017)	0.789(±0.018)	0.785(±0.016)	**0.800(±0.005)**	**0.783(±0.016)**	**0.794(±0.018)**	**0.791(±0.014)**
D → B	0.797(±0.008)	0.787(±0.006)	0.796(±0.007)	0.797(±0.004)	**0.813(±0.006)**	**0.809(±0.006)**	0.796(±0.007)	**0.814(±0.003)**
D → E	0.798(±0.008)	0.791(±0.004)	0.800(±0.005)	0.798(±0.003)	**0.799(±0.006)**	**0.801(±0.003)**	**0.802(±0.005)**	**0.804(±0.002)**
D → K	0.804(±0.015)	0.794(±0.015)	0.802(±0.016)	0.803(±0.014)	**0.813(±0.013)**	**0.796(±0.014)**	**0.806(±0.016)**	**0.805(±0.012)**
E → B	0.707(±0.020)	0.681(±0.014)	0.705(±0.016)	0.695(±0.014)	**0.711(±0.018)**	**0.695(±0.014)**	**0.708(±0.016)**	**0.705(±0.012)**
E → D	0.738(±0.011)	0.724(±0.012)	0.737(±0.013)	0.740(±0.010)	**0.744(±0.008)**	**0.732(±0.010)**	**0.744(±0.013)**	0.740(±0.009)
E → K	0.879(±0.011)	0.859(±0.012)	0.875(±0.013)	0.865(±0.011)	**0.880(±0.009)**	**0.877(±0.010)**	**0.885(±0.013)**	**0.877(±0.009)**
K → B	0.727(±0.014)	0.717(±0.006)	0.735(±0.007)	0.733(±0.006)	**0.741(±0.013)**	**0.730(±0.005)**	**0.737(±0.007)**	**0.744(±0.004)**
K → D	**0.754(±0.026)**	0.755(±0.008)	0.765(±0.009)	0.765(±0.007)	0.752(±0.023)	**0.758(±0.006)**	**0.769(±0.009)**	**0.770(±0.005)**
K → E	0.859(±0.005)	0.841(±0.004)	**0.858(±0.005)**	0.859(±0.003)	**0.862(±0.003)**	**0.850(±0.004)**	0.854(±0.005)	**0.865(±0.000)**
Avg.	0.787(±0.012)	0.772(±0.010)	0.787(±0.010)	0.785(±0.009)	**0.792(±0.010)**	**0.783(±0.009)**	**0.790(±0.010)**	**0.792(±0.007)**

Table 19: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method CDAN.

	Cross-Validation for Binary Classifier				Aggregation			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.803(±0.015)	0.790(±0.011)	0.803(±0.011)	0.795(±0.009)	**0.816(±0.013)**	**0.798(±0.009)**	**0.813(±0.011)**	**0.804(±0.007)**
B → E	0.788(±0.004)	0.772(±0.015)	0.786(±0.016)	0.782(±0.014)	**0.794(±0.003)**	**0.792(±0.014)**	**0.793(±0.016)**	**0.796(±0.012)**
B → K	0.804(±0.012)	0.797(±0.014)	0.807(±0.015)	0.803(±0.013)	**0.812(±0.009)**	**0.812(±0.013)**	**0.809(±0.015)**	**0.805(±0.011)**
D → B	0.801(±0.005)	0.783(±0.003)	0.802(±0.005)	0.805(±0.003)	**0.813(±0.003)**	**0.794(±0.002)**	**0.810(±0.005)**	**0.820(±0.001)**
D → E	0.803(±0.008)	0.786(±0.008)	0.801(±0.009)	0.804(±0.007)	**0.815(±0.006)**	**0.791(±0.007)**	**0.805(±0.009)**	**0.821(±0.005)**
D → K	0.803(±0.029)	0.793(±0.015)	0.805(±0.017)	0.810(±0.014)	**0.813(±0.028)**	**0.809(±0.014)**	**0.807(±0.017)**	**0.812(±0.014)**
E → B	0.719(±0.021)	0.704(±0.021)	0.719(±0.022)	0.714(±0.020)	**0.733(±0.020)**	**0.717(±0.020)**	**0.725(±0.022)**	**0.724(±0.018)**
E → D	0.749(±0.004)	0.735(±0.008)	0.746(±0.009)	0.738(±0.007)	**0.764(±0.002)**	**0.754(±0.008)**	**0.753(±0.009)**	**0.740(±0.005)**
E → K	0.883(±0.010)	0.854(±0.007)	0.881(±0.008)	**0.873(±0.007)**	**0.898(±0.008)**	**0.862(±0.007)**	**0.890(±0.008)**	0.871(±0.004)
K → B	**0.739(±0.010)**	0.720(±0.017)	0.732(±0.018)	0.727(±0.016)	0.738(±0.008)	**0.731(±0.016)**	0.732(±0.018)	**0.732(±0.014)**
K → D	0.767(±0.008)	0.753(±0.014)	**0.770(±0.015)**	0.770(±0.012)	**0.773(±0.005)**	**0.764(±0.013)**	0.766(±0.015)	**0.782(±0.010)**
K → E	0.864(±0.006)	0.847(±0.006)	0.863(±0.007)	0.858(±0.004)	**0.870(±0.004)**	**0.858(±0.005)**	**0.870(±0.007)**	**0.870(±0.002)**
Avg.	0.794(±0.011)	0.778(±0.012)	0.793(±0.013)	0.790(±0.011)	**0.803(±0.009)**	**0.790(±0.011)**	**0.798(±0.013)**	**0.798(±0.009)**

Table 20: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method DSAN.

	Cross-Validation for Binary Classifier				Aggregation			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.812(±0.004)	0.795(±0.004)	**0.811(±0.005)**	0.804(±0.003)	**0.825(±0.003)**	**0.804(±0.003)**	0.810(±0.005)	**0.817(±0.000)**
B → E	0.786(±0.007)	0.769(±0.008)	0.779(±0.008)	0.771(±0.005)	**0.795(±0.005)**	**0.784(±0.007)**	**0.786(±0.008)**	**0.784(±0.003)**
B → K	0.809(±0.006)	0.793(±0.006)	0.797(±0.007)	0.794(±0.005)	**0.816(±0.004)**	**0.801(±0.004)**	0.797(±0.007)	**0.805(±0.002)**
D → B	0.806(±0.009)	0.801(±0.008)	**0.811(±0.009)**	0.807(±0.007)	**0.809(±0.007)**	**0.812(±0.008)**	0.809(±0.009)	**0.812(±0.005)**
D → E	0.804(±0.012)	0.789(±0.021)	0.802(±0.022)	0.800(±0.021)	**0.810(±0.011)**	**0.813(±0.020)**	**0.806(±0.022)**	**0.805(±0.020)**
D → K	0.830(±0.012)	0.809(±0.017)	0.823(±0.018)	0.817(±0.016)	**0.844(±0.010)**	**0.826(±0.017)**	**0.832(±0.018)**	**0.829(±0.014)**
E → B	0.686(±0.016)	0.675(±0.018)	0.685(±0.019)	0.680(±0.017)	**0.696(±0.013)**	**0.687(±0.017)**	**0.689(±0.019)**	**0.684(±0.016)**
E → D	0.711(±0.017)	0.706(±0.016)	0.715(±0.016)	0.708(±0.015)	**0.727(±0.016)**	**0.711(±0.015)**	0.715(±0.016)	**0.726(±0.013)**
E → K	0.885(±0.009)	0.868(±0.006)	0.880(±0.008)	0.876(±0.005)	**0.898(±0.008)**	**0.878(±0.005)**	**0.886(±0.008)**	**0.885(±0.002)**
K → B	0.720(±0.011)	0.717(±0.010)	**0.731(±0.011)**	0.732(±0.009)	**0.730(±0.010)**	**0.722(±0.008)**	0.729(±0.011)	**0.737(±0.007)**
K → D	0.729(±0.013)	0.732(±0.009)	**0.748(±0.010)**	0.750(±0.009)	**0.742(±0.011)**	**0.733(±0.008)**	0.746(±0.010)	**0.764(±0.007)**
K → E	0.863(±0.013)	0.857(±0.008)	0.870(±0.009)	0.867(±0.008)	**0.872(±0.011)**	**0.872(±0.008)**	**0.880(±0.009)**	**0.876(±0.005)**
Avg.	0.787(±0.011)	0.776(±0.011)	0.788(±0.012)	0.784(±0.010)	**0.797(±0.009)**	**0.787(±0.010)**	**0.790(±0.012)**	**0.794(±0.008)**

Table 21: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method DIRT.

	Cross-Validation for Binary Classifier				Aggregation			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.794(±0.005)	0.770(±0.005)	0.786(±0.006)	0.778(±0.004)	**0.801(±0.003)**	**0.787(±0.003)**	**0.793(±0.006)**	**0.784(±0.002)**
B → E	0.760(±0.012)	0.744(±0.010)	0.756(±0.011)	0.755(±0.009)	**0.764(±0.010)**	**0.750(±0.010)**	**0.761(±0.011)**	**0.762(±0.007)**
B → K	0.782(±0.018)	0.765(±0.015)	0.781(±0.017)	0.776(±0.014)	**0.784(±0.015)**	**0.778(±0.015)**	**0.786(±0.017)**	**0.785(±0.012)**
D → B	0.794(±0.010)	0.780(±0.004)	0.794(±0.005)	0.785(±0.003)	**0.798(±0.008)**	**0.787(±0.004)**	**0.795(±0.005)**	**0.795(±0.001)**
D → E	0.783(±0.006)	0.768(±0.004)	0.776(±0.005)	0.776(±0.004)	**0.790(±0.004)**	**0.791(±0.004)**	**0.779(±0.005)**	**0.792(±0.002)**
D → K	**0.789(±0.013)**	0.771(±0.011)	0.789(±0.011)	0.779(±0.009)	0.786(±0.011)	**0.780(±0.010)**	**0.796(±0.011)**	**0.785(±0.007)**
E → B	0.711(±0.021)	0.694(±0.017)	0.706(±0.018)	0.702(±0.016)	**0.721(±0.019)**	**0.709(±0.016)**	0.706(±0.018)	**0.717(±0.014)**
E → D	0.732(±0.007)	0.722(±0.007)	0.731(±0.007)	0.726(±0.005)	**0.746(±0.004)**	**0.729(±0.007)**	**0.734(±0.007)**	**0.728(±0.003)**
E → K	0.875(±0.011)	0.861(±0.010)	0.874(±0.012)	0.870(±0.010)	**0.882(±0.008)**	**0.885(±0.009)**	**0.880(±0.012)**	**0.876(±0.008)**
K → B	0.723(±0.007)	0.701(±0.004)	0.723(±0.005)	0.725(±0.003)	**0.726(±0.005)**	**0.706(±0.003)**	**0.729(±0.005)**	**0.742(±0.001)**
K → D	0.748(±0.014)	0.726(±0.014)	0.750(±0.014)	0.743(±0.012)	0.748(±0.013)	**0.740(±0.013)**	**0.757(±0.014)**	**0.754(±0.010)**
K → E	0.858(±0.008)	0.833(±0.006)	0.854(±0.007)	0.849(±0.005)	**0.867(±0.006)**	**0.838(±0.005)**	**0.858(±0.007)**	**0.858(±0.003)**
Avg.	0.779(±0.011)	0.761(±0.009)	0.777(±0.010)	0.772(±0.008)	**0.784(±0.009)**	**0.774(±0.008)**	**0.781(±0.010)**	**0.782(±0.006)**

Table 22: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method AdvSKM.

	Cross-Validation for Binary Classifier				Aggregation			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.794(±0.007)	0.777(±0.008)	0.792(±0.010)	0.782(±0.008)	0.794(±0.006)	**0.788(±0.007)**	**0.798(±0.010)**	**0.794(±0.006)**
B → E	0.761(±0.012)	0.746(±0.018)	0.756(±0.018)	0.755(±0.017)	**0.770(±0.010)**	**0.766(±0.017)**	0.756(±0.018)	**0.773(±0.014)**
B → K	0.781(±0.017)	0.764(±0.021)	0.776(±0.022)	0.776(±0.020)	**0.792(±0.015)**	**0.781(±0.020)**	**0.781(±0.022)**	**0.787(±0.017)**
D → B	0.790(±0.002)	0.776(±0.002)	0.789(±0.003)	0.783(±0.001)	**0.795(±0.000)**	**0.792(±0.002)**	**0.790(±0.003)**	**0.795(±0.002)**
D → E	0.781(±0.004)	0.756(±0.003)	0.772(±0.004)	0.777(±0.002)	**0.790(±0.002)**	**0.767(±0.002)**	**0.778(±0.004)**	**0.783(±0.000)**
D → K	0.789(±0.015)	0.770(±0.014)	0.788(±0.015)	0.785(±0.014)	**0.800(±0.013)**	**0.783(±0.013)**	**0.798(±0.015)**	**0.792(±0.011)**
E → B	0.699(±0.012)	0.679(±0.013)	0.693(±0.014)	0.689(±0.013)	**0.710(±0.011)**	**0.698(±0.012)**	**0.701(±0.014)**	**0.699(±0.011)**
E → D	0.735(±0.011)	0.722(±0.009)	0.732(±0.010)	0.722(±0.009)	**0.744(±0.008)**	**0.724(±0.008)**	**0.734(±0.010)**	**0.727(±0.007)**
E → K	0.873(±0.009)	0.857(±0.007)	0.872(±0.008)	0.862(±0.006)	**0.884(±0.007)**	**0.865(±0.007)**	**0.881(±0.008)**	**0.863(±0.005)**
K → B	0.722(±0.006)	0.703(±0.012)	**0.717(±0.013)**	0.711(±0.011)	**0.723(±0.003)**	**0.709(±0.011)**	0.713(±0.013)	**0.724(±0.009)**
K → D	0.748(±0.008)	0.734(±0.008)	0.752(±0.009)	0.753(±0.006)	**0.760(±0.006)**	**0.742(±0.006)**	**0.762(±0.009)**	**0.759(±0.005)**
K → E	0.856(±0.006)	0.834(±0.007)	0.854(±0.007)	0.847(±0.006)	**0.864(±0.004)**	**0.849(±0.006)**	**0.860(±0.007)**	**0.857(±0.003)**
Avg.	0.777(±0.009)	0.760(±0.010)	0.774(±0.011)	0.770(±0.009)	**0.786(±0.007)**	**0.772(±0.009)**	**0.779(±0.011)**	**0.779(±0.008)**

Table 23: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method HoMM.

	Cross-Validation for Binary Classifier				Aggregation			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.790(±0.009)	0.775(±0.006)	0.790(±0.008)	0.783(±0.006)	**0.794(±0.007)**	**0.797(±0.005)**	**0.796(±0.008)**	**0.789(±0.004)**
B → E	0.763(±0.014)	0.749(±0.015)	0.760(±0.016)	0.750(±0.014)	**0.774(±0.012)**	**0.759(±0.013)**	**0.763(±0.016)**	**0.752(±0.012)**
B → K	0.783(±0.014)	0.774(±0.022)	0.779(±0.022)	0.772(±0.020)	**0.793(±0.012)**	**0.787(±0.021)**	**0.786(±0.022)**	**0.780(±0.018)**
D → B	0.795(±0.006)	0.780(±0.007)	0.791(±0.008)	0.783(±0.006)	**0.796(±0.004)**	**0.799(±0.006)**	**0.793(±0.008)**	**0.790(±0.005)**
D → E	0.784(±0.004)	0.757(±0.007)	0.778(±0.008)	0.782(±0.005)	**0.790(±0.001)**	**0.776(±0.007)**	**0.784(±0.008)**	**0.792(±0.004)**
D → K	0.789(±0.021)	0.780(±0.014)	0.790(±0.015)	0.781(±0.014)	**0.796(±0.018)**	**0.787(±0.014)**	**0.792(±0.015)**	**0.789(±0.011)**
E → B	0.705(±0.024)	0.680(±0.022)	0.699(±0.024)	0.702(±0.022)	**0.713(±0.022)**	**0.692(±0.021)**	**0.700(±0.024)**	**0.713(±0.020)**
E → D	0.736(±0.007)	0.727(±0.007)	**0.740(±0.007)**	0.739(±0.005)	**0.746(±0.005)**	**0.744(±0.005)**	0.737(±0.007)	**0.742(±0.003)**
E → K	0.877(±0.011)	0.862(±0.007)	0.874(±0.009)	0.879(±0.007)	**0.885(±0.009)**	**0.880(±0.006)**	**0.884(±0.009)**	**0.890(±0.004)**
K → B	0.726(±0.003)	0.714(±0.006)	0.722(±0.007)	0.719(±0.005)	**0.734(±0.001)**	**0.721(±0.006)**	**0.731(±0.007)**	**0.737(±0.004)**
K → D	0.751(±0.013)	0.736(±0.007)	0.752(±0.009)	0.750(±0.006)	**0.756(±0.011)**	**0.750(±0.007)**	**0.760(±0.009)**	**0.759(±0.004)**
K → E	0.856(±0.008)	0.837(±0.006)	0.856(±0.007)	0.852(±0.005)	**0.862(±0.005)**	0.837(±0.004)	**0.865(±0.007)**	**0.864(±0.003)**
Avg.	0.780(±0.011)	0.764(±0.011)	0.778(±0.012)	0.774(±0.010)	**0.787(±0.009)**	**0.777(±0.010)**	**0.782(±0.012)**	**0.783(±0.008)**

Table 24: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method DDC.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.801(±0.010)	0.795(±0.009)	0.799(±0.011)	0.790(±0.009)	**0.814(±0.008)**	**0.812(±0.008)**	**0.804(±0.011)**	**0.801(±0.006)**
B → E	0.773(±0.008)	0.753(±0.012)	0.763(±0.012)	0.756(±0.010)	**0.781(±0.006)**	**0.762(±0.011)**	**0.772(±0.012)**	**0.765(±0.008)**
B → K	0.785(±0.017)	0.765(±0.017)	0.781(±0.018)	0.773(±0.016)	**0.797(±0.016)**	**0.782(±0.017)**	**0.789(±0.018)**	**0.779(±0.013)**
D → B	0.797(±0.004)	0.783(±0.002)	0.794(±0.003)	0.789(±0.001)	**0.803(±0.002)**	**0.794(±0.000)**	**0.804(±0.003)**	**0.803(±0.000)**
D → E	0.785(±0.003)	0.769(±0.009)	**0.779(±0.011)**	0.769(±0.009)	**0.793(±0.002)**	**0.778(±0.008)**	0.777(±0.011)	**0.775(±0.007)**
D → K	0.792(±0.014)	0.774(±0.016)	0.791(±0.016)	0.785(±0.013)	**0.799(±0.012)**	**0.787(±0.015)**	**0.792(±0.016)**	0.785(±0.010)
E → B	0.710(±0.024)	0.695(±0.022)	0.709(±0.024)	0.706(±0.022)	**0.713(±0.022)**	**0.704(±0.022)**	**0.717(±0.024)**	**0.709(±0.020)**
E → D	0.739(±0.004)	0.724(±0.002)	0.739(±0.003)	0.729(±0.000)	**0.747(±0.002)**	**0.731(±0.002)**	**0.748(±0.003)**	**0.738(±0.001)**
E → K	0.878(±0.007)	0.863(±0.007)	0.876(±0.009)	0.875(±0.007)	**0.894(±0.005)**	**0.882(±0.006)**	**0.878(±0.009)**	**0.886(±0.005)**
K → B	0.739(±0.009)	0.720(±0.007)	**0.735(±0.008)**	0.738(±0.005)	**0.754(±0.007)**	**0.735(±0.007)**	0.733(±0.008)	**0.756(±0.003)**
K → D	0.755(±0.015)	0.733(±0.016)	0.749(±0.017)	0.746(±0.014)	**0.770(±0.013)**	**0.743(±0.015)**	**0.754(±0.017)**	**0.753(±0.011)**
K → E	0.857(±0.010)	0.830(±0.004)	**0.855(±0.005)**	0.851(±0.004)	**0.864(±0.008)**	**0.842(±0.003)**	0.850(±0.005)	**0.863(±0.002)**
Avg.	0.784(±0.010)	0.767(±0.010)	0.781(±0.011)	0.776(±0.009)	**0.794(±0.008)**	**0.779(±0.009)**	**0.785(±0.011)**	**0.784(±0.007)**

Table 25: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method DeepCoral.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
B → D	0.799(±0.005)	0.785(±0.007)	0.794(±0.009)	**0.792(±0.007)**	**0.809(±0.002)**	**0.796(±0.007)**	**0.803(±0.009)**	0.790(±0.004)
B → E	0.779(±0.014)	0.747(±0.020)	**0.761(±0.021)**	0.756(±0.019)	**0.787(±0.012)**	**0.759(±0.018)**	0.756(±0.021)	**0.758(±0.017)**
B → K	0.781(±0.029)	0.782(±0.010)	0.793(±0.011)	0.783(±0.009)	**0.788(±0.026)**	**0.791(±0.009)**	**0.799(±0.011)**	**0.791(±0.007)**
D → B	0.798(±0.012)	0.791(±0.006)	0.802(±0.007)	0.792(±0.005)	**0.799(±0.010)**	**0.796(±0.005)**	**0.806(±0.007)**	**0.798(±0.003)**
D → E	0.795(±0.009)	0.777(±0.009)	0.802(±0.010)	0.792(±0.008)	**0.811(±0.007)**	**0.795(±0.008)**	**0.812(±0.010)**	**0.800(±0.006)**
D → K	0.799(±0.018)	0.786(±0.013)	**0.804(±0.015)**	0.796(±0.013)	**0.812(±0.016)**	**0.797(±0.013)**	0.803(±0.015)	**0.804(±0.011)**
E → B	0.719(±0.014)	0.714(±0.016)	0.721(±0.017)	0.726(±0.016)	**0.732(±0.012)**	**0.727(±0.015)**	**0.727(±0.017)**	**0.732(±0.013)**
E → D	**0.750(±0.009)**	0.730(±0.007)	0.744(±0.008)	0.742(±0.007)	0.746(±0.007)	**0.738(±0.006)**	**0.752(±0.008)**	**0.752(±0.005)**
E → K	0.875(±0.015)	0.857(±0.009)	**0.875(±0.010)**	0.867(±0.007)	**0.879(±0.013)**	**0.875(±0.008)**	0.872(±0.010)	**0.880(±0.004)**
K → B	0.745(±0.004)	0.719(±0.004)	0.731(±0.004)	0.731(±0.003)	**0.748(±0.003)**	**0.727(±0.003)**	**0.733(±0.004)**	**0.732(±0.001)**
K → D	0.775(±0.019)	0.723(±0.030)	0.740(±0.031)	0.730(±0.029)	**0.783(±0.018)**	**0.732(±0.030)**	**0.749(±0.031)**	**0.740(±0.027)**
K → E	0.864(±0.010)	0.859(±0.005)	0.867(±0.006)	0.865(±0.004)	**0.868(±0.008)**	**0.866(±0.004)**	**0.870(±0.006)**	**0.866(±0.002)**
Avg.	0.790(±0.013)	0.773(±0.011)	0.786(±0.012)	0.781(±0.011)	**0.797(±0.011)**	**0.783(±0.011)**	**0.790(±0.012)**	**0.787(±0.008)**

Table 26: Mean and standard deviation (after ±) of target classification accuracy on Amazon Reviews computed with domain adaptation method CMD.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
0 → 6	0.746(±0.008)	0.701(±0.031)	0.716(±0.026)	0.678(±0.012)	**0.793(±0.022)**	**0.740(±0.001)**	**0.728(±0.019)**	**0.702(±0.011)**
1 → 6	0.897(±0.002)	0.864(±0.009)	0.867(±0.006)	0.829(±0.008)	**0.953(±0.029)**	**0.906(±0.022)**	**0.891(±0.000)**	**0.858(±0.007)**
2 → 7	0.488(±0.010)	0.484(±0.009)	0.493(±0.009)	0.460(±0.005)	**0.532(±0.021)**	**0.528(±0.022)**	**0.494(±0.003)**	**0.475(±0.004)**
3 → 8	0.839(±0.012)	0.845(±0.032)	0.864(±0.012)	0.826(±0.002)	**0.877(±0.019)**	**0.883(±0.000)**	**0.877(±0.006)**	**0.859(±0.001)**
4 → 5	0.928(±0.006)	0.456(±0.410)	0.923(±0.012)	0.888(±0.002)	**0.975(±0.025)**	**0.497(±0.379)**	**0.938(±0.006)**	**0.920(±0.001)**
Avg.	0.780(±0.008)	0.670(±0.098)	0.773(±0.013)	0.736(±0.006)	**0.826(±0.023)**	**0.711(±0.085)**	**0.786(±0.007)**	**0.763(±0.005)**

Table 27: Mean and standard deviation (after ±) of target classification accuracy on HHAR computed with domain adaptation method MMDA.

Scenario	Cross-Validation for Binary Classifier				Aggregation			
	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
0 → 6	0.596(±0.048)	0.634(±0.045)	0.631(±0.043)	0.593(±0.029)	**0.638(±0.017)**	**0.674(±0.012)**	**0.643(±0.038)**	**0.620(±0.028)**
1 → 6	0.939(±0.006)	0.908(±0.011)	0.906(±0.013)	0.868(±0.001)	**0.982(±0.024)**	**0.955(±0.019)**	**0.919(±0.006)**	**0.896(±0.000)**
2 → 7	0.472(±0.005)	0.470(±0.008)	0.476(±0.008)	0.438(±0.006)	**0.516(±0.025)**	**0.511(±0.024)**	**0.493(±0.002)**	**0.464(±0.005)**
3 → 8	0.960(±0.030)	0.921(±0.028)	0.934(±0.030)	0.896(±0.016)	**1.005(±0.002)**	**0.964(±0.005)**	**0.954(±0.024)**	**0.928(±0.014)**
4 → 5	0.648(±0.562)	0.947(±0.013)	0.947(±0.012)	0.909(±0.002)	**0.682(±0.530)**	**0.984(±0.018)**	**0.958(±0.006)**	**0.925(±0.002)**
Avg.	0.723(±0.130)	0.776(±0.021)	0.779(±0.021)	0.741(±0.011)	**0.765(±0.120)**	**0.818(±0.016)**	**0.793(±0.015)**	**0.767(±0.010)**

Table 28: Mean and standard deviation (after ±) of target classification accuracy on HHAR computed with domain adaptation method CoDATS.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
$0 \rightarrow 6$	$0.604(\pm0.036)$	$0.652(\pm0.043)$	$0.657(\pm0.027)$	$0.619(\pm0.013)$	$\mathbf{0.647}(\pm\mathbf{0.006})$	$\mathbf{0.690}(\pm\mathbf{0.010})$	$\mathbf{0.669}(\pm\mathbf{0.021})$	$\mathbf{0.637}(\pm\mathbf{0.012})$
$1 \rightarrow 6$	$0.936(\pm0.002)$	$0.890(\pm0.038)$	$0.906(\pm0.010)$	$0.868(\pm0.004)$	$\mathbf{0.977}(\pm\mathbf{0.028})$	$\mathbf{0.935}(\pm\mathbf{0.006})$	$\mathbf{0.916}(\pm\mathbf{0.004})$	$\mathbf{0.891}(\pm\mathbf{0.002})$
$2 \rightarrow 7$	$0.327(\pm0.284)$	$0.511(\pm0.026)$	$0.533(\pm0.045)$	$0.495(\pm0.031)$	$\mathbf{0.367}(\pm\mathbf{0.253})$	$\mathbf{0.552}(\pm\mathbf{0.006})$	$\mathbf{0.541}(\pm\mathbf{0.038})$	$\mathbf{0.513}(\pm\mathbf{0.030})$
$3 \rightarrow 8$	$0.964(\pm0.006)$	$0.919(\pm0.007)$	$0.923(\pm0.015)$	$0.885(\pm0.001)$	$\mathbf{1.005}(\pm\mathbf{0.026})$	$\mathbf{0.964}(\pm\mathbf{0.026})$	$\mathbf{0.939}(\pm\mathbf{0.008})$	$\mathbf{0.900}(\pm\mathbf{0.001})$
$4 \rightarrow 5$	$0.654(\pm0.566)$	$0.951(\pm0.005)$	$0.957(\pm0.010)$	$0.919(\pm0.004)$	$\mathbf{0.696}(\pm\mathbf{0.536})$	$\mathbf{0.998}(\pm\mathbf{0.026})$	$\mathbf{0.969}(\pm\mathbf{0.004})$	$\mathbf{0.960}(\pm\mathbf{0.003})$
Avg.	$0.697(\pm0.179)$	$0.785(\pm0.024)$	$0.795(\pm0.021)$	$0.757(\pm0.011)$	$\mathbf{0.738}(\pm\mathbf{0.170})$	$\mathbf{0.828}(\pm\mathbf{0.015})$	$\mathbf{0.807}(\pm\mathbf{0.015})$	$\mathbf{0.780}(\pm\mathbf{0.010})$

Table 29: Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR computed with domain adaptation method DANN.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
$0 \rightarrow 6$	$0.622(\pm0.009)$	$0.507(\pm0.281)$	$0.666(\pm0.014)$	$0.628(\pm0.000)$	$\mathbf{0.660}(\pm\mathbf{0.022})$	$\mathbf{0.558}(\pm\mathbf{0.249})$	$\mathbf{0.685}(\pm\mathbf{0.008})$	$\mathbf{0.658}(\pm\mathbf{0.002})$
$1 \rightarrow 6$	$0.933(\pm0.000)$	$0.906(\pm0.008)$	$0.908(\pm0.006)$	$0.870(\pm0.008)$	$\mathbf{0.982}(\pm\mathbf{0.031})$	$\mathbf{0.940}(\pm\mathbf{0.025})$	$\mathbf{0.922}(\pm\mathbf{0.001})$	$\mathbf{0.899}(\pm\mathbf{0.007})$
$2 \rightarrow 7$	$0.550(\pm0.061)$	$0.547(\pm0.061)$	$0.552(\pm0.066)$	$0.514(\pm0.052)$	$\mathbf{0.592}(\pm\mathbf{0.030})$	$\mathbf{0.592}(\pm\mathbf{0.029})$	$\mathbf{0.562}(\pm\mathbf{0.060})$	$\mathbf{0.533}(\pm\mathbf{0.050})$
$3 \rightarrow 8$	$0.874(\pm0.076)$	$0.863(\pm0.059)$	$0.862(\pm0.056)$	$0.829(\pm0.042)$	$\mathbf{0.911}(\pm\mathbf{0.043})$	$\mathbf{0.910}(\pm\mathbf{0.028})$	$\mathbf{0.878}(\pm\mathbf{0.050})$	$\mathbf{0.850}(\pm\mathbf{0.041})$
$4 \rightarrow 5$	$0.980(\pm0.000)$	$0.707(\pm0.423)$	$0.954(\pm0.006)$	$0.916(\pm0.008)$	$\mathbf{1.025}(\pm\mathbf{0.031})$	$\mathbf{0.749}(\pm\mathbf{0.391})$	$\mathbf{0.972}(\pm\mathbf{0.001})$	$\mathbf{0.935}(\pm\mathbf{0.007})$
Avg.	$0.792(\pm0.029)$	$0.706(\pm0.166)$	$0.788(\pm0.029)$	$0.751(\pm0.022)$	$\mathbf{0.834}(\pm\mathbf{0.031})$	$\mathbf{0.750}(\pm\mathbf{0.144})$	$\mathbf{0.804}(\pm\mathbf{0.024})$	$\mathbf{0.775}(\pm\mathbf{0.021})$

Table 30: Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR computed with domain adaptation method CDAN.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
$0 \rightarrow 6$	$0.719(\pm0.070)$	$0.350(\pm0.289)$	$0.705(\pm0.043)$	$0.667(\pm0.029)$	$\mathbf{0.753}(\pm\mathbf{0.039})$	$\mathbf{0.391}(\pm\mathbf{0.257})$	$\mathbf{0.720}(\pm\mathbf{0.038})$	$\mathbf{0.686}(\pm\mathbf{0.028})$
$1 \rightarrow 6$	$0.929(\pm0.000)$	$0.410(\pm0.429)$	$0.893(\pm0.023)$	$0.855(\pm0.009)$	$\mathbf{0.975}(\pm\mathbf{0.031})$	$\mathbf{0.464}(\pm\mathbf{0.396})$	$\mathbf{0.911}(\pm\mathbf{0.018})$	$\mathbf{0.870}(\pm\mathbf{0.008})$
$2 \rightarrow 7$	$0.496(\pm0.000)$	$0.495(\pm0.002)$	$0.495(\pm0.001)$	$0.457(\pm0.013)$	$\mathbf{0.538}(\pm\mathbf{0.031})$	$\mathbf{0.538}(\pm\mathbf{0.030})$	$\mathbf{0.509}(\pm\mathbf{0.006})$	$\mathbf{0.482}(\pm\mathbf{0.012})$
$3 \rightarrow 8$	$0.971(\pm0.005)$	$0.926(\pm0.005)$	$0.927(\pm0.005)$	$0.889(\pm0.009)$	$\mathbf{1.009}(\pm\mathbf{0.027})$	$\mathbf{0.965}(\pm\mathbf{0.028})$	$\mathbf{0.937}(\pm\mathbf{0.001})$	$\mathbf{0.913}(\pm\mathbf{0.008})$
$4 \rightarrow 5$	$0.654(\pm0.566)$	$0.460(\pm0.417)$	$0.939(\pm0.005)$	$0.901(\pm0.009)$	$\mathbf{0.694}(\pm\mathbf{0.535})$	$\mathbf{0.503}(\pm\mathbf{0.385})$	$\mathbf{0.947}(\pm\mathbf{0.001})$	$\mathbf{0.925}(\pm\mathbf{0.008})$
Avg.	$0.754(\pm0.128)$	$0.528(\pm0.228)$	$0.792(\pm0.015)$	$0.754(\pm0.014)$	$\mathbf{0.794}(\pm\mathbf{0.133})$	$\mathbf{0.572}(\pm\mathbf{0.219})$	$\mathbf{0.805}(\pm\mathbf{0.013})$	$\mathbf{0.775}(\pm\mathbf{0.013})$

Table 31: Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR computed with domain adaptation method DSAN.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
$0 \rightarrow 6$	$0.693(\pm0.072)$	$0.694(\pm0.050)$	$0.709(\pm0.043)$	$0.671(\pm0.029)$	$\mathbf{0.737}(\pm\mathbf{0.041})$	$\mathbf{0.729}(\pm\mathbf{0.018})$	$\mathbf{0.729}(\pm\mathbf{0.037})$	$\mathbf{0.701}(\pm\mathbf{0.027})$
$1 \rightarrow 6$	$0.938(\pm0.000)$	$0.884(\pm0.029)$	$0.904(\pm0.010)$	$0.866(\pm0.004)$	$\mathbf{0.985}(\pm\mathbf{0.031})$	$\mathbf{0.923}(\pm\mathbf{0.004})$	$\mathbf{0.909}(\pm\mathbf{0.004})$	$\mathbf{0.887}(\pm\mathbf{0.004})$
$2 \rightarrow 7$	$0.194(\pm0.336)$	$0.582(\pm0.090)$	$0.545(\pm0.041)$	$0.507(\pm0.027)$	$\mathbf{0.234}(\pm\mathbf{0.306})$	$\mathbf{0.620}(\pm\mathbf{0.058})$	$\mathbf{0.548}(\pm\mathbf{0.036})$	$\mathbf{0.541}(\pm\mathbf{0.026})$
$3 \rightarrow 8$	$0.848(\pm0.007)$	$0.839(\pm0.022)$	$0.842(\pm0.011)$	$0.804(\pm0.003)$	$\mathbf{0.887}(\pm\mathbf{0.024})$	$\mathbf{0.879}(\pm\mathbf{0.011})$	$\mathbf{0.856}(\pm\mathbf{0.006})$	$\mathbf{0.823}(\pm\mathbf{0.002})$
$4 \rightarrow 5$	$0.984(\pm0.004)$	$0.949(\pm0.008)$	$0.951(\pm0.009)$	$0.919(\pm0.005)$	$\mathbf{1.026}(\pm\mathbf{0.026})$	$\mathbf{0.989}(\pm\mathbf{0.024})$	$\mathbf{0.964}(\pm\mathbf{0.002})$	$\mathbf{0.941}(\pm\mathbf{0.004})$
Avg.	$0.731(\pm0.084)$	$0.790(\pm0.040)$	$0.790(\pm0.023)$	$0.753(\pm0.014)$	$\mathbf{0.774}(\pm\mathbf{0.086})$	$\mathbf{0.828}(\pm\mathbf{0.023})$	$\mathbf{0.801}(\pm\mathbf{0.017})$	$\mathbf{0.779}(\pm\mathbf{0.013})$

Table 32: Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR computed with domain adaptation method DIRT.

Scenario	Cross-Validation for Binary Classifier KuLSIF	Exp	LR	SQ	Aggregation KuLSIF	Exp	LR	SQ
$0 \rightarrow 6$	$0.718(\pm0.009)$	$0.714(\pm0.010)$	$0.705(\pm0.020)$	$0.667(\pm0.006)$	$\mathbf{0.766}(\pm\mathbf{0.022})$	$\mathbf{0.755}(\pm\mathbf{0.022})$	$\mathbf{0.721}(\pm\mathbf{0.014})$	$\mathbf{0.690}(\pm\mathbf{0.005})$
$1 \rightarrow 6$	$0.857(\pm0.006)$	$0.835(\pm0.026)$	$0.836(\pm0.025)$	$0.798(\pm0.011)$	$\mathbf{0.893}(\pm\mathbf{0.025})$	$\mathbf{0.894}(\pm\mathbf{0.007})$	$\mathbf{0.840}(\pm\mathbf{0.019})$	$\mathbf{0.823}(\pm\mathbf{0.010})$
$2 \rightarrow 7$	$0.490(\pm0.014)$	$0.497(\pm0.010)$	$0.502(\pm0.024)$	$0.464(\pm0.010)$	$\mathbf{0.526}(\pm\mathbf{0.017})$	$\mathbf{0.541}(\pm\mathbf{0.022})$	$\mathbf{0.519}(\pm\mathbf{0.018})$	$\mathbf{0.492}(\pm\mathbf{0.008})$
$3 \rightarrow 8$	$0.810(\pm0.002)$	$0.806(\pm0.005)$	$0.807(\pm0.005)$	$0.769(\pm0.009)$	$\mathbf{0.856}(\pm\mathbf{0.029})$	$\mathbf{0.840}(\pm\mathbf{0.027})$	$\mathbf{0.819}(\pm\mathbf{0.001})$	$\mathbf{0.796}(\pm\mathbf{0.008})$
$4 \rightarrow 5$	$0.884(\pm0.005)$	$0.875(\pm0.014)$	$0.878(\pm0.017)$	$0.840(\pm0.003)$	$\mathbf{0.924}(\pm\mathbf{0.026})$	$\mathbf{0.921}(\pm\mathbf{0.019})$	$\mathbf{0.897}(\pm\mathbf{0.011})$	$\mathbf{0.874}(\pm\mathbf{0.002})$
Avg.	$0.752(\pm0.007)$	$0.746(\pm0.013)$	$0.746(\pm0.018)$	$0.708(\pm0.008)$	$\mathbf{0.793}(\pm\mathbf{0.024})$	$\mathbf{0.790}(\pm\mathbf{0.019})$	$\mathbf{0.759}(\pm\mathbf{0.012})$	$\mathbf{0.735}(\pm\mathbf{0.006})$

Table 33: Mean and standard deviation (after $\pm$) of target classification accuracy on HHAR computed with domain adaptation method AdvSKM.

	Cross-Validation for Binary Classifier				**Aggregation**			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	0.732(±0.010)	0.719(±0.025)	0.725(±0.024)	0.687(±0.010)	**0.782(±0.021)**	**0.764(±0.007)**	**0.742(±0.018)**	**0.721(±0.008)**
$1 \to 6$	0.879(±0.011)	0.856(±0.019)	0.853(±0.024)	0.815(±0.010)	**0.917(±0.020)**	**0.899(±0.013)**	**0.866(±0.017)**	**0.845(±0.009)**
$2 \to 7$	0.455(±0.015)	0.466(±0.029)	0.474(±0.024)	0.436(±0.010)	**0.499(±0.016)**	**0.515(±0.002)**	**0.491(±0.018)**	**0.466(±0.009)**
$3 \to 8$	0.818(±0.005)	0.820(±0.008)	0.822(±0.004)	0.784(±0.010)	**0.859(±0.026)**	**0.863(±0.024)**	**0.835(±0.002)**	**0.820(±0.009)**
$4 \to 5$	0.911(±0.005)	0.449(±0.399)	0.899(±0.031)	0.861(±0.017)	**0.958(±0.026)**	**0.496(±0.366)**	**0.905(±0.025)**	**0.894(±0.016)**
Avg.	0.759(±0.009)	0.662(±0.096)	0.754(±0.021)	0.716(±0.011)	**0.803(±0.022)**	**0.707(±0.083)**	**0.768(±0.016)**	**0.749(±0.010)**

Table 34: Mean and standard deviation (after ±) of target classification accuracy on HHAR computed with domain adaptation method HoMM.

	Cross-Validation for Binary Classifier				**Aggregation**			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	0.697(±0.013)	0.349(±0.286)	0.690(±0.011)	0.652(±0.003)	**0.748(±0.018)**	**0.397(±0.255)**	**0.709(±0.005)**	**0.680(±0.003)**
$1 \to 6$	0.882(±0.017)	0.639(±0.413)	0.873(±0.014)	0.835(±0.000)	**0.924(±0.014)**	**0.677(±0.381)**	**0.886(±0.009)**	**0.864(±0.001)**
$2 \to 7$	0.439(±0.041)	0.440(±0.036)	0.439(±0.037)	0.401(±0.023)	**0.486(±0.010)**	**0.476(±0.004)**	**0.452(±0.032)**	**0.422(±0.023)**
$3 \to 8$	0.818(±0.008)	0.621(±0.342)	0.823(±0.006)	0.785(±0.008)	**0.858(±0.022)**	**0.663(±0.310)**	**0.840(±0.001)**	**0.813(±0.007)**
$4 \to 5$	0.905(±0.008)	0.219(±0.000)	0.895(±0.025)	0.857(±0.011)	**0.955(±0.023)**	**0.266(±0.032)**	**0.908(±0.019)**	**0.879(±0.010)**
Avg.	0.748(±0.017)	0.454(±0.215)	0.744(±0.019)	0.706(±0.009)	**0.794(±0.018)**	**0.496(±0.196)**	**0.759(±0.013)**	**0.732(±0.009)**

Table 35: Mean and standard deviation (after ±) of target classification accuracy on HHAR computed with domain adaptation method DDC.

	Cross-Validation for Binary Classifier				**Aggregation**			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	0.728(±0.019)	0.700(±0.032)	0.717(±0.025)	0.679(±0.011)	**0.768(±0.012)**	**0.739(±0.001)**	**0.733(±0.019)**	**0.694(±0.010)**
$1 \to 6$	0.886(±0.009)	0.845(±0.039)	0.861(±0.014)	0.823(±0.000)	**0.924(±0.022)**	**0.890(±0.008)**	**0.865(±0.009)**	**0.842(±0.001)**
$2 \to 7$	0.460(±0.012)	0.470(±0.014)	0.480(±0.020)	0.442(±0.006)	**0.509(±0.019)**	**0.512(±0.017)**	**0.488(±0.015)**	**0.462(±0.005)**
$3 \to 8$	0.812(±0.000)	0.810(±0.003)	0.811(±0.001)	0.773(±0.013)	**0.849(±0.031)**	**0.852(±0.029)**	**0.832(±0.004)**	**0.804(±0.011)**
$4 \to 5$	0.618(±0.536)	0.919(±0.020)	0.923(±0.016)	0.885(±0.002)	**0.672(±0.504)**	**0.963(±0.012)**	**0.929(±0.009)**	**0.911(±0.000)**
Avg.	0.701(±0.115)	0.749(±0.021)	0.758(±0.015)	0.720(±0.006)	**0.745(±0.118)**	**0.791(±0.013)**	**0.769(±0.011)**	**0.743(±0.005)**

Table 36: Mean and standard deviation (after ±) of target classification accuracy on HHAR computed with domain adaptation method Deep Coral.

	Cross-Validation for Binary Classifier				**Aggregation**			
Scenario	KuLSIF	Exp	LR	SQ	KuLSIF	Exp	LR	SQ
$0 \to 6$	0.693(±0.012)	0.713(±0.012)	0.693(±0.010)	0.655(±0.004)	**0.739(±0.019)**	**0.757(±0.020)**	**0.706(±0.004)**	**0.667(±0.002)**
$1 \to 6$	0.907(±0.006)	0.885(±0.018)	0.896(±0.008)	0.859(±0.006)	**0.947(±0.024)**	**0.924(±0.015)**	**0.904(±0.001)**	**0.891(±0.005)**
$2 \to 7$	0.320(±0.277)	0.503(±0.027)	0.513(±0.040)	0.475(±0.026)	**0.370(±0.247)**	**0.542(±0.003)**	**0.523(±0.035)**	**0.493(±0.024)**
$3 \to 8$	0.811(±0.011)	0.813(±0.009)	0.813(±0.011)	0.775(±0.003)	**0.856(±0.019)**	**0.860(±0.023)**	**0.825(±0.005)**	**0.797(±0.002)**
$4 \to 5$	0.625(±0.541)	0.937(±0.011)	0.934(±0.013)	0.896(±0.001)	**0.660(±0.511)**	**0.989(±0.021)**	**0.945(±0.007)**	**0.921(±0.001)**
Avg.	0.671(±0.170)	0.770(±0.015)	0.770(±0.016)	0.732(±0.008)	**0.714(±0.164)**	**0.814(±0.017)**	**0.780(±0.010)**	**0.754(±0.007)**

Table 37: Mean and standard deviation (after ±) of target classification accuracy on HHAR computed with domain adaptation method CMD.