ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230510

Revisiting the Robustness of the Minimum Error Entropy Criterion: A Transfer Learning Case Study

Luis Pedro Silvestrin[a;*], Shujian Yu [a;] and Mark Hoogendoorn[a]**

[a]Vrije Universiteit Amsterdam
ORCiD ID: Luis Pedro Silvestrin https://orcid.org/0000-0002-5759-1986, Shujian Yu https://orcid.org/0000-0002-6385-1705, Mark Hoogendoorn https://orcid.org/0000-0003-3356-3574

Abstract. Coping with distributional shifts is an important part of transfer learning methods in order to perform well in real-life tasks. However, most of the existing approaches in this area either focus on an ideal scenario in which the data does not contain noises or employ a complicated training paradigm or model design to deal with distributional shifts. In this paper, we revisit the robustness of the minimum error entropy (MEE) criterion, a widely used objective in statistical signal processing to deal with non-Gaussian noises, and investigate its feasibility and usefulness in real-life transfer learning regression tasks, where distributional shifts are common. Specifically, we put forward a new theoretical result showing the robustness of MEE against covariate shift. We also show that by simply replacing the mean squared error (MSE) loss with the MEE on basic transfer learning algorithms such as fine-tuning and linear probing, we can achieve competitive performance with respect to state-of-the-art transfer learning algorithms. We justify our arguments on both synthetic data and 5 real-world time-series data.

1 Introduction

Robustness is an essential quality for machine learning models to cope with the challenges of real-world scenarios. Typical challenges where robustness is desired include the distributional shift between training and test data [30], noisy data [34], and adversarial attacks [2]. Distributional shifts can result in poor generalization performance, as the model may rely its decision on spurious correlations in the training set [1], while noisy data, such as label noise or response variable noise, can further bias the resulting model.

Most of the robustness research in transfer learning focuses on covariate shift, a special case of general distributional shift in which only the distribution of input ($p(\mathbf{x})$) changes and the conditional distribution ($p(y|\mathbf{x})$) remains the same. These methods aim to learn models that are less sensitive to changes in the data distribution and can adapt to new environments. However, these approaches either focus on an ideal scenario in which the source and target domains are noise free, or are very complicated to implement, requiring extensive training or hyperparameter tuning. For example, there are approaches based on adversarial training [12, 8] which is known for being difficult to converge, and approaches based on boosting [7, 27] which also require extensive hyperparameter tuning of both the base estimators and the approach itself.

* Corresponding Author. Email: l.p.silvestrin@vu.nl
** Email: s.yu3@vu.nl

Research addressing the challenges of noisy data predominantly focuses on the classification setting, also known as label noise [34]. However, there is notably less emphasis on regression problems. For example, most of existing machine learning approaches simply use the mean-squared error (MSE) loss or the mean-absolute error (MAE) loss. The former implicitly assumes the noises (in the response variable) follow a Gaussian distribution, whereas the latter takes a Laplacian distributional assumption. Training a model using MSE or MAE is likely to negatively impact its performance on real-world data, especially when noise is non-Gaussian or non-Laplacian. Therefore, it is of paramount importance to design a transfer learning model such that it can handle a wide range of noise distributions in a non-parametric way (without distributional assumption on noises).

In this paper, we present an approach focusing on covariate shift in a realistic transfer learning regression scenario, where non-Gaussian noise is present. We do so by combining the minimum error entropy (MEE) loss [11] - a widely used learning objective in statistical signal processing to deal with non-Gaussian noises - with classic deep transfer learning methods such as fine-tuning and linear probing. MEE has received lots of attention in signal processing and information theory literature, whereas its practical usage in machine learning, especially deep neural networks, is scarcely investigated due to the difficulty of entropy estimation [29]. Our work put forward a new theoretical result on the robustness of MEE, showing that it also encourages the robustness to covariate shift.

We conduct comprehensive experiments on both synthetic data and 5 real-world time-series data, showing that the simple combination strategy outperforms existing deep neural network based transfer learning approaches with complicated model design or training paradigms. Time-series data serves as an ideal testing ground for our approach since it is commonly affected by covariate shift in many forms such as seasonal variation or sensor drift and contains measurement noises of unknown distributions. The main contributions of this paper are summarized in the following points:

- We provide a theoretical result showing that, besides its resilience to non-Gaussian noise in the response variable, MEE can also cope with covariate shift.
- We use the transfer learning regression setting to show empirically that by simply replacing the training loss with MEE the resulting model becomes more robust to both covariate shifts and response variable noise.
- We compare our approach with other state-of-the-art robust learning methods, and our method consistently outperforms them in

multiple real-life time-series transfer learning tasks.

2 Background and Related Work

In this section, we introduce the prior knowledge about the MEE loss that we will base the rest of the paper on. We also cover the previous work on robust machine learning in general.

2.1 Minimum Error Entropy Criterion

We assume that the explanatory variable $\mathbf{x}$ takes values in a compact domain $\mathcal{X} \in \mathbb{R}^d$, the response variable y takes values in the output space $\mathcal{Y} \in \mathbb{R}$, and

$$y = g^*(\mathbf{x}) + \epsilon \tag{1}$$

where g^* is the ground-truth target function and ϵ is the noise in the regression model.

The purpose of regression is to estimate $g^*(\mathbf{x})$ according to a data set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ drawn independently from an unknown joint distribution $p(\mathbf{x}, y)$. Usually, a loss function $\mathcal{L}(g, (\mathbf{x}, y))$ is used to measure the performance of a hypothesis function $g : \mathcal{X} \to \mathcal{Y}$. For regression, the most used loss function is the mean-squared loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - g(\mathbf{x}_i))^2 = \frac{1}{N}\sum_{i=1}^{N} e_i^2 \tag{2}$$

where $e_i = y_i - g(\mathbf{x}_i)$ is the prediction error for sample $(\mathbf{x}_i, y_i)$. The MSE minimizes the variance of the prediction error. Its optimality heavily depends on the Gaussianity of the data due to the use of a second-order statistic. Hence, the MSE solution may deviate significantly from the ground truth, especially in the presence of noise.

Alternatively, one can obtain g by minimizing the entropy of the prediction error $H(e)$, which is also called the minimum error entropy criterion [11]. If we instantiate $H(e)$ by Rényi's α-order ($\alpha > 0$ and $\alpha \neq 1$) entropy functional [32], the resulting objective of MEE becomes:

$$\min H_\alpha(e) = \min \frac{1}{1-\alpha}\log \int p^\alpha(e)de, \tag{3}$$

in which $p(e)$ is the probability distribution function (PDF) of prediction error e. Entropy is a functional of the PDF and measures the average information contained in that distribution. The basic idea of MEE is to reduce the uncertainty (entropy) of the discrepancies between models and data generating systems and improve the models' predicting capability for unseen data [11]. Compared to MSE, the Rényi's entropy takes into consideration all higher moments. Hence, the MEE can deal with outliers, heavy-tailed noise, or skewed noise distributions.

Practically, $\alpha = 2$ (*a.k.a.*, quadratic Rényi entropy) is the most popular choice, as it can be elegantly estimated by the kernel density estimator (KDE) [28]. In this case, we have:

$$\min H_2(e) \iff \max \int p^2(e)de, \tag{4}$$

and

$$\hat{p}(e) = \frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(e - e_i), \tag{5}$$

where κ_σ is a Gaussian kernel function with width σ.

Hence, the empirical MEE loss can be expressed as:

$$\mathcal{L}_{\text{MEE}} = \max \int \hat{p}^2(e)de = \max \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa_{\sqrt{2}\sigma}(e_i - e_j). \tag{6}$$

Although there is a series of works on the theory and applications of MEE, such as [19, 5, 17], just to name a few, they only investigate or utilize the robustness of MEE against non-Gaussian noises in y. Distinct from these works, one of the motivations of our paper is to point out and systematically investigate the robustness of MEE against distributional shift (i.e., $p(\mathbf{x}, y)$ differs in training and test set), which, to the best of our knowledge, has not been done yet. Moreover, instead of estimating $H_\alpha(e)$ by KDE as shown in Eq. (6), we suggest the use of the matrix-based Rényi's α-order entropy functional [38, 33] to measure $H_\alpha(e)$, which avoids density estimation and thus more suitable for complex data and deep neural networks [39].

2.2 Robust Machine Learning with Covariate Shift

Literature on robust learning under distributional shift mainly focused on dealing with covariate shift, and assumes the training and test data is noise free. Here we list the most recent and relevant works on covariate shift split as those assuming the existence of a target dataset (transfer learning and domain adaptation), and those without that assumption.

Transfer Learning and Domain Adaptation: The majority of transfer learning approaches proposed in the literature categorize according to the differences in the input domains (homogeneous or heterogeneous) or according to the methodology used to bridge the gap between source and target domains [36]. Homogeneous transfer learning assumes that both source and target inputs come from the same feature space and therefore have the same dimensionality, whereas for heterogeneous transfer learning they come from distinct spaces. In this work, we are concerned with the homogeneous category. Within the homogeneous transfer learning literature, the most prominent methodologies fall into two categories: feature-based and instance-based. Feature-based approaches try to mitigate the covariate shift problem by mapping the inputs of the source domain (or both source and target domains) to a feature space where their distribution matches with the target input domain distribution. Instance-based approaches, on the other hand, propose to solve the same issue by weighting the source samples according to their relevance to the target task so, intuitively, source samples that are distant in the target distribution will become less important during training.

In the instance-based category, we have TrAdaBoostR2 (TRB) [27]: a combination of TrAdaBoost [7], a transfer learning variation of AdaBoost, and AdaBoostR2 [9], which is AdaBoost adapted for regression. It adapts both algorithms to transfer learning regression by taking into account that the base learner errors are unbounded, differently from classification, and it bridges the covariate shift by down-weighting source samples that are far from the target distribution. Another more recent instance-based approach is WANN [8], a neural network trained by minimizing an adversarial loss derived from an upper bound of the target generalization error. Its proposed training method includes an auxiliary network that predicts weights to maximize the error of the final model on the source samples, thus indicating which samples are less relevant for the target task.

Among the feature-based transfer learning approaches, there is an adversarial algorithm that learns a new feature representation to align source and target domains by minimizing the margin disparity discrepancy (MDD) proposed by the authors [40]. Domain-adversarial neural networks (DANN) [12] is a seminal method applying adversarial learning for better feature representations across domains. It trains three neural networks simultaneously: one for feature mapping (G_f), one for classification (G_y), and the last one (G_d) is trained to

predict whether the representation given by G_f was produced from a source or target sample. The adversarial aspect of it comes from the min-max game between G_f, which optimizes to fool G_d. In the end, G_f should learn a unified representation for both source and target samples which is indistinguishable to G_d and easily separable for the classifier G_y.

All the aforementioned works propose a complex algorithmic solution to the covariate shift problem in transfer learning and can be difficult to optimize while also introducing extra hyperparameters. They differ fundamentally from our approach, which focuses on replacing the MSE with MEE as the training objective. Since we only change the loss function, our approach can be easily implemented on the existing fine-tuning and linear probing algorithms. Nevertheless, we compare the original TRB and WANN algorithms with our approach in multiple time-series transfer learning regression tasks in our experiments. We choose TRB and WANN over the rest since they showed better performance in several transfer learning regression benchmarks [8].

Robust learning without target data: Another new and less investigated area of research proposes approaches for robustness to covariate shift without assuming any specific knowledge target domain (i.e., no labeled or labeled samples from test set). Note that, this setup is also different to domain generalization (e.g., [1]), in which there are multiple related source domains during training. See also supplementary material [1] for comparison with respect to domain generalization. Anchor regression [31] is a least-squares-based method that includes exogenous variables in the regression model to account for possible causal interventions in the data. Another work proposes [35] an algorithm robust to changes in the data-generating process. It requires the process to be modeled as a causal graph and the changes from source to target domains should be known so that their algorithm is able to take it into account. Although these approaches ensure robustness to changes in the inputs, they are also limited to linear problems so it is less suitable for many real-life regression tasks.

A recent work [14] shows that by training a model by minimizing the Hilbert Schmidt Independence Criterion (HSIC) the resulting model can be more robust to covariate shift. The HSIC [16] is originally studied as an independence measure for random variables, but subsequent work [24] shows that it is also suitable as a loss function by using it to minimize the statistical dependence between covariates and labels. This work motivates our approach to investigate the covariate shift robustness of the MEE criterion since, in the regression case, it is related to minimizing the mutual information between covariates and labels which, in turn, is also a statistical dependence measure.

3 Transfer Learning with MEE

In transfer learning, we assume a small target dataset $\{\mathbf{x}_{iT}, y_{iT}\}_{i=1}^{N_T}$, with inputs $\mathbf{x}_{iT}$ and labels y_{iT} drawn from the target distribution $p_T(\mathbf{x}, y)$, and a large source dataset $\{\mathbf{x}_{iS}, y_{iS}\}_{i=1}^{N_S}$ ($N_S \gg N_T$) with inputs $\mathbf{x}_S$ and labels y_S drawn from the source distribution $p_S(\mathbf{x}, y)$. The final goal is to build a model that generalizes to new unseen samples from p_T. Since the target data is limited, transfer learning proposes to use it combined with the extra source data to create such model. A common challenge practitioners face, besides the lack of target data, is bridging distribution shifts between p_T and p_S. There are three types of shifts that can occur in practice: covariate shift ($p_T(\mathbf{x}) \neq p_S(\mathbf{x})$), label distribution shift ($p_T(y) \neq p_S(y)$), and labeling function shift ($p_T(y|\mathbf{x}) \neq p_S(y|\mathbf{x})$), and there is a plethora of approaches for each of them [26].

In this section, we first provide a new perspective on the robustness of MEE to covariate shift, which motivates our study. We then describe our main algorithm that integrates MEE into two basic transfer learning paradigms. Finally, we elaborate on the implementation details, including the way to estimate entropy, the way to compensate for the model bias, and the way to estimate the kernel size.

3.1 Theoretical Justification on the Robustness of MEE to Covariate Shift

The robustness of MEE against non-Gaussian noises has been extensively investigated in previous literature. We refer interested readers to [5, 6, 4, 19] for more thorough analysis. We also summarize in the supplementary material two key points in this context.

The robustness of MEE against covariate shift is easy to understand. Note that we have:

$$\begin{aligned} \min H(e) &\iff \min H(e) - H(y|\mathbf{x}) \\ &= \min H(e) - H(e + f_\theta(\mathbf{x})|\mathbf{x}) \\ &= \min H(e) - H(e|\mathbf{x}) \\ &= \min I(\mathbf{x}; e) \end{aligned} \tag{7}$$

The first line is due to the fact that the conditional entropy $H(y|\mathbf{x})$ is a constant value that only depends on the training data; the third line is by the property that given two random variables ξ and η, then for any measurable function h, we have $H(\xi|\eta) = H(\xi + h(\eta)|\eta)$ [23].

Hence, minimizing the error entropy actually encourages the minimum dependence between $\mathbf{x}$ and e. In other words, the distribution of input variable $\mathbf{x}$ is independent of the distribution of prediction error e. Since the prediction performance is characterized by $p(e)$[2], it also suggests that the predictor performance was not impacted by the change of $p(\mathbf{x})$, i.e., covariate shift. Therefore the MEE is an ideal loss function in a transfer learning scenario where $p_S(\mathbf{x}) \neq p_T(\mathbf{x})$ and $p(y|\mathbf{x})$ is the same in both source and target domains.

Note that, Greenfeld and Shalit [14] firstly observed and rigorously proved that the HSIC between $p(e)$ and $p(\mathbf{x})$ is an upper bound of the worst-case loss in the target domain in case of covariate shift. Our simple proof in Eq. (7) generalizes the arguments in [14], showing that any independence measures can be used here (rather that just HSIC). Additionally, [14] does not discuss the close relationships between $\min I(\mathbf{x}; e)$ and MEE; and does not systematically investigate the performance and utility of MEE in a practical transfer learning scenario.

3.2 Integration of Minimum Error Entropy and Transfer Learning

In this paper, we focus on two widely used transfer learning techniques: fine-tuning and linear probing. Fine-tuning is a popular approach to transfer learning and has shown great success in several real-life tasks [25]. It consists of readjusting all the weights of a neural network pre-trained with the source dataset through gradient descent by minimizing a given loss function $\mathcal{L}$ using the target data. We assume that the pre-trained neural network architecture $g(\mathbf{x}; w, \theta) = w^\top f(\mathbf{x}; \theta)$ is split into the feature extracting layers $f(\mathbf{x}; \theta)$ with weights θ_S and the regression layer with weight matrix

[1] Supplementary material available at https://arxiv.org/abs/2307.08572.

[2] A good predictor is expected to have a concentrated error distribution with zero mean.

Algorithm 1 Fine-tuning with MEE

Input: target dataset $(\mathbf{x}_T, y_T)$, source feature extractor parameters θ_S, source regressor parameters w_S, learning rate η, number of epochs M.
$\theta_0 \leftarrow \theta_S$
$w_0 \leftarrow w_S$
$e_i \leftarrow y_{iT} - w_S^\top f(\mathbf{x}_{iT}; \theta_S), \forall i \in \{1, ...N\}$ ▷ compute residuals
$\sigma \leftarrow \text{median}(\{(e_i - e_j)^2\}_{i,j=1}^N)$ ▷ kernel size estimation
for $i \leftarrow 1$ to M **do**
    $\theta_i \leftarrow \theta_{i-1} + \eta \nabla_\theta \mathcal{L}_{\text{MEE}}(w_{i-1}, \theta_{i-1})$ ▷ MEE
    $w_i \leftarrow w_{i-1} + \eta \nabla_w \mathcal{L}_{\text{MEE}}(w_{i-1}, \theta_{i-1})$ ▷ MEE
end for
$b \leftarrow \frac{1}{N} \sum_{i=1}^N (y_i - w_M^\top f(\mathbf{x}_T; \theta_M))$ ▷ bias correction
Output: fine-tuned parameters θ_M, w_M and bias b

w_S. Since the source dataset is usually larger and more diverse, the pre-trained model could already have a better performance in the target domain than a randomly initialized model, therefore it works as a "warm start" and makes training easier since the model will need fewer data and epochs to converge. For this approach to work successfully, it requires a certain degree of similarity between the source and target tasks. If both tasks are similar enough, it is likely that fine-tuning will result in a better model for the target task, otherwise negative transfer can happen [36], meaning that the generalization performance on the target task will be hurt.

In linear probing, we freeze the parameters θ_S of the feature extracting layers f of the pre-trained source model and we update only the last layer's weights w_S. This way, we are reusing the same features learned by the source model, so we only need to adapt the last layer to the target task. The idea behind it is that we can tap into the neural network capabilities of learning to extract general meaningful features from large amounts of data. If the source dataset is large and contains a wide variety of samples, then the features extracted by the source model are likely to be relevant also for the target task. On top of that, training only the last layer can be preferable over fine-tuning all the layers since the latter option can distort the features learned from the source dataset [21].

Fine-tuning and linear probing are generally preferred over other more complex approaches since they can be easily combined with any neural network architecture, data types, and loss functions, while also being able to give good results [22, 18]. They are commonly used for neural network transfer learning combined with the cross-entropy loss for classification or the MSE for regression. We propose to improve the regular fine-tuning and linear probing algorithms for regression tasks by combining it with the MEE loss. We do that by instantiating the MEE loss in the place of the MSE and by including two extra steps required by the MEE: the kernel size estimation and the model bias correction.

Algorithms 1 and 2 describe in pseudo-code respectively our novel approaches based on fine-tuning and linear probing. The main suggested modifications are annotated with comments. Before the training starts, we compute the RBF kernel width σ as the median of the pair-wise distance between the residuals of the source model on the target training samples. Once the network parameters are optimized using MEE, we compute the model bias b as the average of its residuals on the training target data. In the following sections we go into detail of how we compute the MEE loss, as well as the justification for the extra steps that it requires.

Algorithm 2 Linear probing with MEE

Input: target dataset $(\mathbf{x}_T, y_T)$, source feature extractor parameters θ_S, source regressor parameters w_S, learning rate η, number of epochs M.
$w_0 \leftarrow w_S$
$e_i \leftarrow y_{iT} - w_S^\top f(\mathbf{x}_{iT}; \theta_S), \forall i \in \{1, ...N\}$ ▷ compute residuals
$\sigma \leftarrow \text{median}(\{(e_i - e_j)^2\}_{i,j=1}^N)$ ▷ kernel size estimation
for $i \leftarrow 1$ to M **do**
    $w_i \leftarrow w_{i-1} + \eta \nabla_w \mathcal{L}_{\text{MEE}}(w_{i-1}, \theta_S)$ ▷ MEE
end for
$b \leftarrow \frac{1}{N} \sum_{i=1}^N (y_i - w_M^\top f(\mathbf{x}_T; \theta_M))$ ▷ bias correction
Output: fine-tuned parameters w_M and bias b

3.2.1 *Matrix-based Implementation of Minimum Error Entropy*

As mentioned in the previous section, the MEE loss has interesting robustness properties, but its KDE version described in Eq. (6) is only suitable for low-dimensional problems, and it is difficult to select the appropriate kernel function. For that reason, in this paper we implement MEE using the matrix-based version [33] of the quadratic Rényi's entropy of the model residuals e:

$$\mathcal{L}_{\text{MEE}}(w, \theta) = \frac{1}{2} \log_2 \left[\sum_{i=1}^N \lambda_i(A)^2 \right] \tag{8}$$

where A is a normalized positive definite matrix computed as $A_{ij} = \frac{1}{N} \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}}$ and $\lambda_i(A)$ is the i-th eigenvalue of A. K is the Gram matrix obtained by evaluating a positive-definite kernel κ on the model's residuals $e_i = y_i - g(\mathbf{x}_i)$ on each pair of training data point. The kernel used is the radial basis function (RBF), so κ and K are expressed as:

$$K_{ij} = \kappa_\sigma(e_i, e_j),$$
$$\kappa_\sigma(\mathbf{x}, y) = \exp\left(-\frac{\|\mathbf{x} - y\|^2}{2\sigma^2} \right).$$

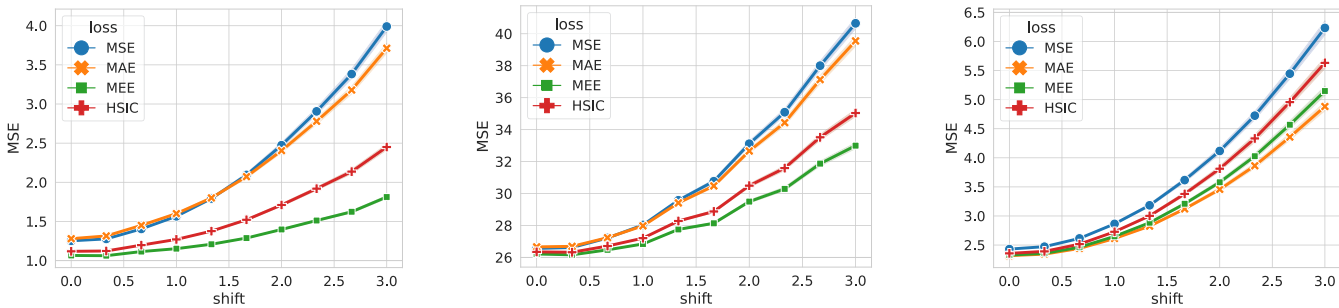
By instantiating the model $g(\mathbf{x})$ with the feature extractor f plus a linear output layer with parameter w and the target dataset $(\mathbf{x}_T, y_T)$, the Gram matrix K becomes:

$$K_{ij} = \kappa_\sigma(y_{iT} - w^\top f(\mathbf{x}_{iT}; \theta), y_{jT} - w^\top f(\mathbf{x}_{jT}; \theta))$$

This matrix-based implementation is ideal since it is differentiable and can be computed in tractable time. In this paper, for the first time, we employ it to transfer learning tasks by coupling it to the fine-tuning and linear probing frameworks. Next, we explain in detail and justify the model bias correction and the computation of the kernel size which are important steps in our algorithms.

3.2.2 *Correcting the Model Bias*

An important difference between MEE and other loss functions such as MSE is that it give the same loss value for models with different errors. More specifically, the error entropy $H(g(\mathbf{x}) - y)$ is the same for any models $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ that differ only by a constant C. This can be easily seen by defining the errors of each model respectively as ϵ_1 and $\epsilon_2 = \epsilon_1 + C$. Therefore the error entropy of each model will then only differ by the integral of the error PDFs $p(\epsilon_1)$ and $p(\epsilon_2)$, which in fact are the same. This can be an issue because, even if the model's error entropy has converged to a minimum after training,



(a) Shifted exponential noise. (b) Mixed Gaussian noise. (c) Laplace noise.

Figure 1. Comparison of the squared error of regression models trained using different loss functions and evaluated on data with increasing covariate shift for different types of response variable noise. The bands indicate the standard deviation.

its predictions might still deviate from the ground truth by a bias constant.

To solve this problem, we estimate the model's bias b as:

$$b = \frac{1}{n}\sum_{i=1}^{n}(y_i - g(\mathbf{x}_i))$$

The bias b is then added to the output to correct the model's predictions, so the corrected model becomes $g_{\text{corrected}}(\mathbf{x}) = g(\mathbf{x}) + b$.

3.2.3 Computing the Kernel Size

The matrix-based implementation of MEE used in this paper relies on the RBF kernel to approximate the PDF of the error in a tractable way. It also introduces the kernel size (or width) σ as an extra hyperparameter to tune in order to obtain an accurate entropy estimation. It is important to have a proper value for σ since otherwise, it can lead to convergence problems during training. A too large σ will result in an all-ones Gram matrix K, thus its eigenvalues in Eq. (8) will tend to zero, while if σ is too small then K will approach the identity and the eigenvalues will approach $1/N$. Furthermore, if σ is small enough (e.g., $\sigma = 0$) the entropy of error will tend to be a fixed value ($\log_2 N$) and is maximized, which also violates the goal of minimizing the error entropy. In either case, the landscape of Eq. (8) over the parameter space will flatten and the training is likely to fail. In our empirical experiments, we calculate the σ to be the median of the pair-wise euclidean distance between the model errors prior to training:

$$\sigma = \text{median}(\{(e_i - e_j)^2\}_{i,j=1}^{N}) \tag{9}$$

This heuristic is also called the median-rule and has been used in practice by previous kernel learning literature [15, 37, 20] and also results in good training convergence in our experiments.

4 Experiments

To empirically evaluate our results about the robustness to covariate shift of the MEE as a loss function, we conduct experiments with both real and synthetic regression datasets, the former focusing on time-series data as argued before. In the first part, we use synthetic regression data to verify empirically our theoretical hypothesis about the robustness of the MEE loss to non-Gaussian noises and covariate shift. In the remaining parts, we use real-life time-series transfer learning regression datasets, which, as argued before, are highly suitable for studying distributional shift robustness of transfer learning methods. We use them to compare the performance of our approach with other state-of-the-art transfer learning algorithms. [3]

4.1 Synthetic Linear Regression Experiment

In this experiment, we evaluate the robustness of the MEE loss to covariate shift using synthetic data so we can fully control the amount of covariate shift between source and target data. We compare MEE with the popular MSE and MAE, as well as the state-of-the-art HSIC, gaining detailed insights about the performance of each loss function. We use a linear model to generate the data to ensure that the gradient descent optimization will converge to a global optimum.

We generate the data of both source and target domains using a linear model $y = \theta^\top \mathbf{x} + \epsilon$, where y is the response variable, $\mathbf{x}$ are the inputs, θ are the regression coefficients and ϵ is the additive noise. We randomly sample the coefficients θ once from $\mathcal{N}(0, 0.1)$ and keep them always fixed. The coefficients θ and the distribution of ϵ remain the same for both source and target datasets. We emulate different degrees of covariate shift by simulating a source dataset with fixed input distribution $p_S(\mathbf{x})$, then simulating multiple target datasets with increasing distribution shift in $p_T(\mathbf{x})$. The source inputs $\mathbf{x}_S$ are sampled from a uniform distribution in the real interval $[-1, 1]^{100}$, while the target inputs $\mathbf{x}_T$ are sampled from a normal distribution $\mathcal{N}(\mu_T, 1)$, and we vary μ_T from 0 up to 3. The covariate shift increases as the mean of $\mathbf{x}_T$ changes: when $\mu_T = 0$, $p_S(\mathbf{x})$ and $p_T(\mathbf{x})$ have the most overlap, and as we increase μ_T, $p_T(\mathbf{x})$ will sample more and more inputs that are outside the support of $p_S(\mathbf{x})$. In order to also compare the robustness of the losses to different noise distributions, we repeat the experiment using shifted exponential noise, mixed Gaussian noise, and Laplace noise. Further implementation details are discussed in the supplementary material.

The results in Figure 1a and Figure 1b show the mean-squared error of the regression models trained with different loss functions in the y-axis and the distance between the means of the input distributions of the source and target datasets in the x-axis. As the target input distribution shifts, MEE's error showed a significantly slower increase compared to other approaches. In the shifted exponential noise case (Figure 1a), where the difference is striking even when compared to the state-of-the-art HSIC. MEE is only outperformed by MAE in the Laplace noise case (Figure 1c), which is expected since the model trained with MAE is the maximum likelihood solution to this case. In all cases, MEE already has lower error than MSE

[3] Our code is available at https://github.com/lpsilvestrin/mee-finetune.

with minimal shift, showing that it can handle the change from uniformly to normally distributed inputs better. Overall, this experiment confirms our theoretical claims about the robustness to covariate shift and, additionally, shows its resilience to different types of response variable noise.

In Figure 2 we use the dataset simulated with Laplace noise to visualize the effect of the kernel size σ on the spread of the error distribution. We can see that by picking it as the median ($\sigma = 1$) the errors are more concentrated around zero[4], while the larger or smaller choices result in a more spread distribution. Additionally, when reducing the size causes substantial degradation of the model such that its performance is even inferior to that of the MSE. This confirms that the procedure explained in Section 3.2.3 eliminates the need for practitioners to manually tune σ.

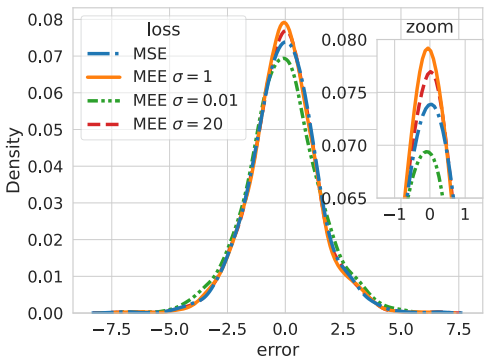


Figure 2. Comparison of error densities using MEE with different kernel sizes.

4.2 *Time-series Transfer Learning*

In this experiment, we verify the performance of our MEE version of fine-tuning and linear probing using 5 real-life time-series regression transfer learning tasks using deep neural networks. We conduct an ablation study comparing our algorithm with the versions using the MSE, MAE, and HSIC. We also compare our method with TRB and WANN which are state-of-the-art transfer learning regression algorithms.

4.2.1 *Datasets*

We compare all the training losses and transfer learning approaches using 5 real-life time-series regression tasks based on 3 datasets from the popular Monash [13] and UCI [10] repositories: the Nasa Turbofan, the Beijing air quality and the bike sharing datasets. The Nasa Turbofan data contains 4 datasets where engines operate under different conditions, providing distributional shifts between each other. We select one of them as a source dataset (NTS), and the remaining as target datasets (NT1, NT2, and NT3). For the Beijing air quality dataset, we use the first year of measurements (2013) as source data (PMS) and the early months of the last year (2017) as target training data (PMT). The models are tested in the remaining data from 2017. For the bike sharing dataset, we separate data from fall, winter, and spring as the source dataset (BKS) while the target data is from summer, and the training target dataset contains information only from the early days of summer.

[4] The error entropy is minimized if the probability of one state dominates, thus forming a highly concentrated distribution.

A summary of all datasets is listed in Table 1. Further details about the distributional shifts on each dataset are discussed in the supplementary material.

Table 1. The size of the train and test sets, the length of the time window and the amount of features of each dataset used in our experiments.

Dataset	n_{train}	n_{test}	window size	features
NTS	14,432	3,299	30	14
NT1	1,678	44,541	30	14
NT2	2,225	19,595	30	14
NT3	2,261	51,767	30	14
PMS	4,000	1,000	24	9
PMT	200	1,000	24	9
BKS	5,143	1,266	24	10
BKT	403	1,681	24	10

4.2.2 *Linear Probing and Fine-tuning Experiments*

There are two steps where MEE can be applied when using fine-tuning (or linear probing) for transfer learning: the pre-training phase and the fine-tuning (or linear probing) phase itself. The classic setup for regression is to use the MSE in both steps, therefore it is also selected as a baseline. We conduct an ablation study with three experiments where we vary the loss functions in the fine-tuning and linear probing phases. In two of them, we keep the pre-training loss fixed as the MSE and in one experiment we vary the loss on both the pre-training and fine-tuning phases:

- **Fix-vary with linear probing** We fix the pre-training loss and we vary linear probing loss.
- **Fix-vary with fine-tuning:** We fix the pre-training loss and vary the fine-tuning loss.
- **Vary-vary with fine-tuning:** We vary both the pre-training and fine-tuning loss.

In the linear probing setup, we evaluate the capacity of each loss to use the features learned by the pre-trained source model for predicting the target labels. The fine-tuning-only setup compares how each loss can adapt the source model to each target task. Finally, the pre-training plus fine-tuning setup compares the loss functions in a complete transfer learning cycle, from building a general source model to tuning it down to the target prediction problem.

For the source models, we use a temporal convolutional neural network (TCN) as the architecture for modeling sequential data. This type of neural network is known for outperforming other classical time-series architectures such as the LSTM in many benchmark datasets [3]. Further details of the hyperparameter choices are discussed in the supplementary material.

The HSIC is implemented with RBF kernels, as reported by the original paper [14]. We select the kernel size the same way we do for MEE (Section 3.2.3): we compute the median of the pairwise distance matrices for inputs and labels for each training dataset. Table 2 contains the kernel sizes selected for covariates and for response variables for each dataset. We repeat all the runs with all models and loss functions 20 times with different weight initializations and different training and validation samples, and we compare their results on the target test datasets. The significance of the results is validated through a paired Wilcoxon test with an adjusted p-value of 0.05.

In the comparison using linear probing of a source model trained with the MSE (Table 3), MEE displays significantly lower error than all other approaches in 4 out of 5 datasets. Only for the NT2 dataset,

Table 2. RBF kernel size per dataset for both the covariates (σ_X) and the response variable (σ_Y).

	NT1	NT2	NT3	PMT	BKT	NS	PMS	BKS
σ_Y	0.5	1	0.5	0.3	0.8	0.3	0.5	0.3
σ_X	800	450	800	300	200	400	300	250

there is no significant difference between the performances of MSE, MAE, and MEE. This result is promising because it shows that by using MEE we are getting more from the features learned from the source dataset.

Table 3. Fix-vary with linear probing: average target squared error obtained from using different loss functions.

Data	MSE-MSE	MSE-MAE	MSE-HSIC	MSE-MEE
BKT	0.25 ± 0.029	0.26 ± 0.04	0.26 ± 0.031	0.24 ± 0.031
NT1	0.87 ± 0.016	0.9 ± 0.0092	0.89 ± 0.025	0.87 ± 0.018
NT2	0.48 ± 0.024	0.48 ± 0.029	0.54 ± 0.047	0.49 ± 0.025
NT3	0.72 ± 0.022	0.73 ± 0.018	0.73 ± 0.015	0.7 ± 0.0092
PMT	0.57 ± 0.039	0.56 ± 0.036	0.63 ± 0.038	0.54 ± 0.034

Regarding the experiments using MEE only in the fine-tuning phase (Table 4), we see that it outperforms all other losses in two datasets, and is on par with the best-performing (MSE) on two other datasets. It is only outperformed in one dataset, where MAE and HSIC have lower error, but overall it confirms the be a more robust loss than MSE. When using MEE for both pre-training and fine-tuning (Table 5), for two datasets it has lower error than the other losses. Only for PMT and NT2, it results in higher errors compared to MSE and MAE, respectively. However, MEE always has significantly better performance in more transfer learning tasks than each other loss in the individual comparison. Summing up, the results confirm that our approach has more resilience to arbitrary distribution shifts encountered in real-life datasets compared to the counterparts using other popular training losses such as MAE and MSE, and even state-of-the-art losses such as HSIC.

Table 4. Fix-vary with fine-tuning: average target squared error obtained from using different loss functions. The pre-training loss function is always the MSE. For BKT and PMT, there was no significant difference between MEE and each other loss function.

Data	MSE-MSE	MSE-MAE	MSE-HSIC	MSE-MEE
BKT	0.16 ± 0.015	0.17 ± 0.018	0.16 ± 0.018	0.16 ± 0.012
NT1	0.47 ± 0.008	0.48 ± 0.008	0.48 ± 0.008	0.46 ± 0.007
NT2	0.59 ± 0.013	0.58 ± 0.012	0.58 ± 0.012	0.59 ± 0.009
NT3	0.45 ± 0.007	0.45 ± 0.007	0.45 ± 0.007	0.44 ± 0.006
PMT	0.44 ± 0.042	0.44 ± 0.038	0.47 ± 0.034	0.46 ± 0.024

4.2.3 Comparison with State-of-the-art Approaches

Additionally, we compare our approach with two other state-of-the-art transfer learning algorithms tailored for regression: TrAdaBoost.R2 (TRB) [27] and WANN [8]. TRB is an ensemble learning method while WANN is based on adversarial learning. We describe both approaches and their hyperparameters in detail in the supplementary material.

Table 5. Vary-vary with fine-tuning: average target squared error obtained from using different loss functions.

Data	MSE-MSE	MAE-MAE	HSIC-HSIC	MEE-MEE
BKT	0.16 ± 0.02	0.16 ± 0.01	0.26 ± 0.07	0.16 ± 0.014
NT1	0.47 ± 0.01	0.48 ± 0.01	0.48 ± 0.01	0.45 ± 0.007
NT2	0.59 ± 0.01	0.57 ± 0.01	0.59 ± 0.01	0.59 ± 0.013
NT3	0.45 ± 0.01	0.45 ± 0.01	0.45 ± 0.01	0.44 ± 0.005
PMT	0.44 ± 0.04	0.45 ± 0.05	0.52 ± 0.04	0.47 ± 0.021

In the results shown in Table 6, our fine-tuning approach using MEE is able to outperform the WANN in all tasks, and it performs significantly better than TRB in 4 out of 5 tasks. This shows that fine-tuning is still a competitive method for time-series transfer learning.

It is important to emphasize that WANN and TRB were originally conceived for tabular data, so, besides our efforts to tune all hyperparameters to the time-series tasks, they might still be improved by more thorough adaptation. It could be an interesting future work direction to adapt specialized transfer learning regression algorithms such as WANN and TRB to time-series tasks and to use MEE instead of MSE.

Table 6. Target squared error of fine-tuning using MEE and other SOTA transfer learning regression methods.

Data	MSE-MSE	TRB	WANN	MEE-MEE
BKT	0.16 ± 0.016	1.2 ± 1.2	0.48 ± 0.043	0.16 ± 0.014
NT1	0.47 ± 0.009	0.53 ± 0.011	0.94 ± 0.03	0.45 ± 0.007
NT2	0.59 ± 0.014	0.66 ± 0.007	0.68 ± 0.095	0.59 ± 0.013
NT3	0.45 ± 0.008	0.52 ± 0.004	0.77 ± 0.018	0.44 ± 0.005
PMT	0.44 ± 0.044	0.38 ± 0.018	1.2 ± 0.49	0.47 ± 0.021

5 Conclusion

In this paper, we revisit the robustness of the MEE loss function to show that, besides its resilience to non-Gaussian response variable noise, it is also robust to covariate shift, a common challenge in many machine learning applications. We draw awareness to the fact that MEE is better for training machine learning models for regression tasks than the commonly used MSE, and can even outperform state-of-the-art loss functions such as the HSIC. We validate our hypothesis empirically on synthetic data representing different degrees of covariate shift. Additionally, we show that MEE in combination with fine-tuning can outperform other loss functions and even other state-of-the-art transfer learning regression methods in real-world time-series tasks.

There are many future work possibilities based on our results about the robustness of the MEE loss. Existing transfer learning algorithms such as TRB and WANN can be adapted to use MEE instead of MSE. Our results also suggest that the out-of-distribution transfer learning generalization might also be improved by using MEE, which might lead to new interesting theoretical studies.

Acknowledgements

This work has been conducted as part of the Just in Time Maintenance project funded by the European Fund for Regional Development.

References

[1] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish, 'Invariance principle meets information bottleneck for out-of-distribution generalization', *Advances in Neural Information Processing Systems*, **34**, 3438–3450, (2021).

[2] Naveed Akhtar and Ajmal Mian, 'Threat of adversarial attacks on deep learning in computer vision: A survey', *IEEE Access*, **6**, 14410–14430, (2018).

[3] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun, 'An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling', *arXiv:1803.01271 [cs]*, (2018).

[4] Ba-Dong CHEN, Jin-Chun HU, Yu ZHU, and Zeng-Qi SUN, 'Information theoretic interpretation of error criteria', *Acta Automatica Sinica*, **35**(10), 1302–1309, (2009).

[5] Badong Chen, Lei Xing, Bin Xu, Haiquan Zhao, and José C. Príncipe, 'Insights into the robustness of minimum error entropy estimation', *IEEE Transactions on Neural Networks and Learning Systems*, **29**(3), 731–737, (2018).

[6] Badong Chen, Yu Zhu, Jinchun Hu, and Ming Zhang, 'A new interpretation on the mmse as a robust mee criterion', *Signal Processing*, **90**(12), 3313–3316, (2010).

[7] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu, 'Boosting for transfer learning', in *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, p. 193–200, New York, NY, USA, (2007). Association for Computing Machinery.

[8] Antoine de Mathelin, Guillaume Richard, François Deheeger, Mathilde Mougeot, and Nicolas Vayatis, 'Adversarial weighting for domain adaptation in regression', in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 49–56, (2021).

[9] Harris Drucker, 'Improving regressors using boosting techniques', in *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, p. 107–115, San Francisco, CA, USA, (1997). Morgan Kaufmann Publishers Inc.

[10] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[11] D. Erdogmus and J.C. Principe, 'An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems', *IEEE Transactions on Signal Processing*, **50**(7), 1780–1786, (2002).

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, 'Domain-adversarial training of neural networks', *J. Mach. Learn. Res.*, **17**(1), 2096–2030, (jan 2016).

[13] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso, 'Monash time series forecasting archive', in *Neural Information Processing Systems Track on Datasets and Benchmarks*, (2021).

[14] Daniel Greenfeld and Uri Shalit, 'Robust learning with the Hilbert-schmidt independence criterion', in *Proceedings of the 37th International Conference on Machine Learning*, eds., Hal Daumé III and Aarti Singh, volume 119 of *Proceedings of Machine Learning Research*, pp. 3759–3768. PMLR, (13–18 Jul 2020).

[15] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola, 'A kernel two-sample test', *The Journal of Machine Learning Research*, **13**(1), 723–773, (2012).

[16] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf, 'Measuring statistical dependence with hilbert-schmidt norms', in *Algorithmic Learning Theory*, eds., Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita, pp. 63–77, Berlin, Heidelberg, (2005). Springer Berlin Heidelberg.

[17] Xin Guo, Ting Hu, and Qiang Wu, 'Distributed minimum error entropy algorithms', *The Journal of Machine Learning Research*, **21**(1), 4968–4998, (2020).

[18] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M. Hospedales, 'Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference', in *Proceedings of the IEEE/CVF CVPR (CVPR)*, pp. 9068–9077, (June 2022).

[19] Ting Hu, Jun Fan, Qiang Wu, and Ding-Xuan Zhou, 'Learning theory approach to minimum error entropy criterion', *J. Mach. Learn. Res.*, **14**(1), 377–397, (feb 2013).

[20] Robert Jenssen, 'Kernel entropy component analysis', *IEEE TPAMI*, **32**(5), 847–860, (2009).

[21] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022.

[22] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig, 'Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing', *ACM Comput. Surv.*, **55**(9), (jan 2023).

[23] David J. C. MacKay, *Information Theory, Inference & Learning Algorithms*, Cambridge University Press, USA, 2002.

[24] Joris Mooij, Dominik Janzing, Jonas Peters, and Bernhard Schölkopf, 'Regression by dependence minimization and its application to causal inference in additive noise models', in *Proceedings of the 26th Annual ICML*, ICML '09, p. 745–752, New York, NY, USA, (2009). Association for Computing Machinery.

[25] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, 'Learning and transferring mid-level image representations using convolutional neural networks', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (June 2014).

[26] Sinno Jialin Pan and Qiang Yang, 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering*, **22**(10), 1345–1359, (2010).

[27] David Pardoe and Peter Stone, 'Boosting for regression transfer', in *ICML*, pp. 863–870, (2010).

[28] Emanuel Parzen, 'On Estimation of a Probability Density Function and Mode', *The Annals of Mathematical Statistics*, **33**(3), 1065 – 1076, (1962).

[29] Jose C Principe, *Information theoretic learning: Renyi's entropy and kernel perspectives*, Springer Science & Business Media, 2010.

[30] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, *Dataset Shift in Machine Learning*, The MIT Press, 2009.

[31] Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters, 'Anchor Regression: Heterogeneous Data Meet Causality', *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **83**(2), 215–246, (01 2021).

[32] Alfréd Rényi, 'On measures of entropy and information', *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability (June 20-July 30 1960)*, pages 547–561, (Jun 1960). Referenced by: MathSciNet [MR0132570].

[33] Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C. Principe, 'Measures of entropy from data using infinitely divisible kernels', *IEEE Transactions on Information Theory*, **61**(1), 535–548, (2015).

[34] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee, 'Learning from noisy labels with deep neural networks: A survey', *IEEE Transactions on Neural Networks and Learning Systems*, 1–19, (2022).

[35] Adarsh Subbaswamy, Peter Schulam, and Suchi Saria, 'Preventing failures due to dataset shift: Learning predictive models that transport', in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, eds., Kamalika Chaudhuri and Masashi Sugiyama, volume 89 of *Proceedings of Machine Learning Research*, pp. 3118–3127. PMLR, (16–18 Apr 2019).

[36] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang, 'A survey of transfer learning', *Journal of Big Data*, **3**(1), 9, (May 2016).

[37] S. Yu, L. Giraldo, R. Jenssen, and J. C. Principe, 'Multivariate extension of matrix-based rényi's α-order entropy functional', *IEEE TPAMI*, **42**(11), 2960–2966, (nov 2020).

[38] Shujian Yu, Francesco Alesiani, Xi Yu, Robert Jenssen, and Jose Principe, 'Measuring dependence with matrix-based entropy functional', *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(12), 10781–10789, (May 2021).

[39] Shujian Yu and Jose C Principe, 'Understanding autoencoders with information theoretic concepts', *Neural Networks*, **117**, 104–123, (2019).

[40] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan, 'Bridging theory and algorithm for domain adaptation', in *Proceedings of the 36th International Conference on Machine Learning*, eds., Kamalika Chaudhuri and Ruslan Salakhutdinov, volume 97 of *Proceedings of Machine Learning Research*, pp. 7404–7413. PMLR, (09–15 Jun 2019).