DARE-GRAM : Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices

Ismail Nejjar
EPFL, Switzerland
ismail.nejjar@epfl.ch

Qin Wang
ETH Zurich, Switzerland
qwang@ethz.ch

Olga Fink
EPFL, Switzerland
olga.fink@epfl.ch

Abstract

Unsupervised Domain Adaptation Regression (DAR) aims to bridge the domain gap between a labeled source dataset and an unlabelled target dataset for regression problems. Recent works mostly focus on learning a deep feature encoder by minimizing the discrepancy between source and target features. In this work, we present a different perspective for the DAR problem by analyzing the closed-form ordinary least square (OLS) solution to the linear regressor in the deep domain adaptation context. Rather than aligning the original feature embedding space, we propose to align the inverse Gram matrix of the features, which is motivated by its presence in the OLS solution and the Gram matrix's ability to capture the feature correlations. Specifically, we propose a simple yet effective DAR method which leverages the pseudo-inverse low-rank property to align the scale and angle in a selected subspace generated by the pseudo-inverse Gram matrix of the two domains. We evaluate our method on three domain adaptation regression benchmarks. Experimental results demonstrate that our method achieves state-of-the-art performance. Our code is available at https://github.com/ismailnejjar/DARE-GRAM.

1. Introduction

Regression problems, in which models learn to predict continuous variables, are one fundamental paradigm in machine learning. Regression problems are omnipresent in many different applications, including computer vision tasks, such as head-pose estimation [76], facial landmark detection [37], human pose estimation [80], depth estimation [20] and eye-tracking problems [58], and also widely in industrial applications, such as product quality prediction and condition monitoring [69]. Nevertheless, real-world applications are often subject to the environmental conditions under which the data are collected and other influencing factors, hence domain gaps between datasets are inevitable.

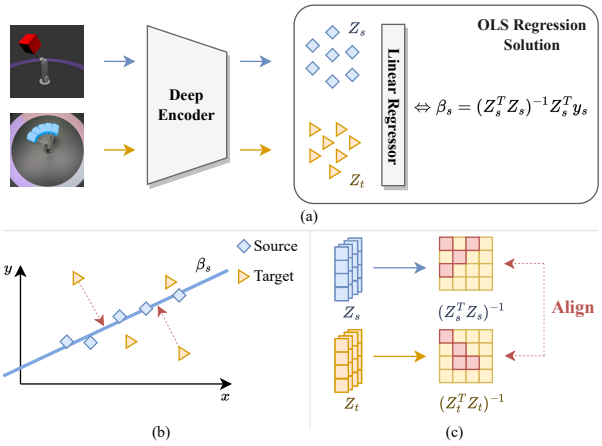


Figure 1. Illustration of the UDA for regression setup and our main motivation. (a) Deep domain adaptation networks commonly use a shared deep feature encoder and a shared linear regression layer. We propose to pay close attention to the linear regressor, where the ordinary least square (OLS) solution is well-known. (b) Given a trained source linear regressor β_s, the target features Z_t may not be calibrated to β_s. (c) Unlike previous adaptation methods which align in the original feature embedding space, we propose to align the inverse Gram matrix of the features $(Z^T Z)^{-1}$, which is motivated by its presence in the OLS solution and the Gram matrix's ability to capture the feature correlations.

Unsupervised Domain adaptation (UDA) aims to overcome the distributional shift between a labeled source domain and an unlabelled target domain. Many UDA methods have been proposed to alleviate the domain shift problem. One common UDA direction is feature alignment by adversarial learning [45] or explicit losses such as maximum mean discrepancy [44] to learn domain-invariant representations. Input alignment [77] and self-training using pseudo-label refinement [41] are also popular UDA directions. While many DA methods have been developed and evaluated for classification and segmentation problems, some are not directly transferable to DA regression [10]. Pioneer works in Domain Adaptation Regression (DAR) [12, 46] introduced theoretical analysis for the problem. A few algorithms were proposed to tackle DAR.

For example, importance weighting [14, 74] and feature alignment [7, 51] have shown improved results over learning only from the source. Most recent unsupervised DAR methods [10,59] use the deep learning framework and focus on learning a shared deep feature extractor by directly minimizing the discrepancy between source and target features. By doing so, it is implicitly assumed that if the feature discrepancy is small, a shared linear regressor can be easily learned from the source supervision. This formulation used by existing works focuses solely on the feature extractor.

In this work, we propose to look at the DAR problem from a different perspective. In particular, we pay close attention to the linear regressor, which is attached directly after the feature extractor. Motivated by the closed-form ordinary least squares (OLS) regression solution, we analyze the potential optimal regressor for each domain. We reveal in Section 3.2 that even when the discrepancy between source and target features is small, the learning of a shared linear regressor could still be difficult because of the *inverse Gram matrix* term in the OLS solution.

In light of this, we propose an ordinary least squares inspired deep domain adaptation method for regression called Domain Adaptation Regression by aligning the inverse GRAM matrices (DARE-GRAM). As shown in Figure 1, unlike previous methods, which directly align the features, we align the inverse Gram matrix of the features. This is motivated by its presence in the closed-form solution of the ordinary least squares. More specifically, we leverage the low-rank property of the pseudo-inverse to align a selected subspace in scale and angle engendered by the Gram Matrix, which represents the intensity and pairwise interactions between different features for the source and target domains. The scale and angle alignment based on the Gram matrix can lead to a better-calibrated regressor with regard to both source and target data. The contributions of this work are as follows:

- We offer a new perspective to understand the UDA for regression problems by leveraging the well-known closed-form solutions to the linear regression problem.
- Rather than aligning the original feature embedding space, we propose to align the inverse Gram matrix of the features.
- Empirical results on three benchmarks validate the superiority of the DARE-GRAM over baseline methods.

2. Related Work

Unsupervised Domain Adaptation. The goal of unsupervised domain adaptation (UDA) [53] is to address the domain-shift problem between a labeled source and an unlabeled target domain. UDA has been widely studied for classification and segmentation problems [64, 66] to mitigate the gap between features across different domains. Early works addressed this problem via instance weighting [30, 60], feature transformation [51], and feature space alignment [16]. More recently, unsupervised domain adaptation has shown impressive results [29, 49, 71]. Discrepancy minimization [33, 44] and domain adversarial learning [17, 28] have been widely used within UDA methods to mitigate the gap between features across different domains. Moreover, feature regularization-based approaches [9] and domain-specific normalization-based methods [8, 40] have also demonstrated good performance. While most approaches perform feature alignment in the encoding feature space, some works proposed to carry out alignment in the input space [77]. More recently, self-training has also demonstrated encouraging results by training the network with gradually improved target pseudo-label [42,68,78,79]. Existing UDA methods mostly focus on classification and segmentation problems. While some UDA techniques can directly be applied to regression problems, recent works have shown that many do not perform well in the regression setup [3, 10].

Domain Adaptation for Regression. Domain Adaptation for Regression (DAR) has received relatively little attention in comparison to classification problems. Early theoretical properties for DAR were introduced in [12, 46]. Different algorithms were proposed to tackle DAR [55]. Unfortunately, most algorithms require access to a labeled target domain and are unsuitable for UDA regression. For instance, Boosting strategies have been explored [52, 67] to extend previous classification domain adaptation methods based on AdaBoost [47] to regression tasks. Other instance weighting methods in the shallow regime, [14, 74, 75] have been explored for a different range of applications. Some specific vision applications have been explored in the context of UDA [31, 34, 36, 50] , such as monocular depth estimation [1, 5, 43, 63] or gaze estimation [4, 26]. However, these methods aim at improving upon a specific task and not for regression tasks in general. Recent works for UDA regression were proposed [10, 59, 72]. A key finding in RSD [10] is that in regression problems, deep neural networks are less robust to feature scaling than classification, and aligning the distributions of deep representations will alter feature scale and impede domain adaptation regression. To tackle this challenge, the authors of [10] proposed to match the orthogonal bases of both domains to close domain shifts without altering their feature scale by introducing a new geometrical distance. While RSD-based methods have shown improved results for DAR, matching only the eigenvectors can have some disadvantages, such as more loose numerical error bound [2] and may not satisfy the more strict conditions for distribution estimation [35]. In contrast to RSD, we propose using the inverse Gram Matrix, which carries the nec-

essary information to align the source and target features while being less sensitive to the batch size.

Gram Matrix and Subspace Alignment. Distribution alignment approaches have been used for domain adaptation [16, 62, 70]. Subspace-based domain adaptation has demonstrated good performance in visual domain adaptation [23, 24], modeling distribution change by finding the best intermediate subspaces. The methods first independently compute a domain-specific d-dimensional subspace for the source and target data. Then project the source and target data into intermediate ones along the shortest geodesic path connecting the two d-dimensional subspaces on the Grassmann manifold. Instead of computing a large number of intermediate subspaces, the authors of [16] directly aligns the two subspaces. Furthermore, the authors in [61] proposed to incorporate distribution alignment into subspace adaptation to align the source and target features. Given their close relation, distribution alignment approaches have also been used for Neural Style Transfer (NST) [6, 32]. Early works on NST [19] introduced the Gram Matrix as the statistics of feature maps to extract style-specific attributes. Although the connection between aligning distributions and NST may not be straightforward, it was demonstrated in [39] that the style loss in [19] may be expressed as an unbiased empirical estimate of the Maximum Mean Discrepancy (MMD) [25] with a quadratic kernel. Unlike previous works in neural style transfer which directly aligns the Gram matrix, we propose to align the inverse Gram matrix as it is presented in the OLS solution. We will show in our method and our ablation study that this is critical for regression problems.

3. Methods

3.1. Problem Definition

In UDA, we are given labeled samples $\chi_s = \{(x_s^i, y_s^i)\}_{i=1}^{N_s}$ from the source domain and unlabeled samples $\chi_t = \{(x_t^i)\}_{i=1}^{N_t}$ from the target domain, where N_s and N_t denote the number of samples in χ_s and χ_t. In contrast to the discrete labels $\mathcal{Y}$ in classification problems, this work focuses on the regression problem where $\mathcal{Y} \subset \mathbb{R}^{N_r}$ is multidimensional and continuous, and N_r correspond to the number of regression tasks. The discrepancy between $P(\chi_s)$ and $P(\chi_t)$ is one of the main challenges for UDA. We aim to learn a model $F : x \mapsto y$, which can generalize well on the target domain. Formally, we want to minimize the expected error on the target data:

$$\arg\min_F \mathbb{E}_{(x^t, y^t)} \|F(x^t), y^t\|_2^2, \tag{1}$$

where y^t is not known during the training.

A source-only baseline can be learned by using the supervision from the source data by minimizing the Mean square error loss (MSE) between the prediction and the ground truth label on the source samples:

$$\mathcal{L}_{src} = \frac{1}{N_s} \sum_{i=1}^{N_s} \|\tilde{y}_s^i - y_s^i\|_2^2, \tag{2}$$

where $\tilde{y}_s^n = F(x_s^i)$ is the predicted value for the training source image x_s^i. To overcome the distribution gap between the source and target, additional constraints should be given.

3.2. Motivation

In deep domain adaptation models, given an input image x, a feature encoder h_θ is used to learn the deep representation $z = h_\theta(x)$ of p dimensions. A linear layer g_β is then applied on z to make the final prediction:

$$\tilde{y} = F(x) = g_\beta(h_\theta(x)) = g_\beta(z). \tag{3}$$

During training, the feature matrix is $Z = [z^1, ..., z^b]$ where $Z \in \mathbb{R}^{b \times p}$ for a batch of b images. For many adaptation methods [10, 59], the focus has been on minimizing the distribution difference between source features Z_s and target features Z_t. Given the aligned features, it is often assumed that they will then lead to a good performance on the target domain. However, this formulation focuses solely on the feature extractor h_θ and does not take the discrimination ability of the final linear layer g_β into account. Target features aligned with the source domain may not be adapted to the linear layer. This can be especially dangerous for regression problems because it has been demonstrated empirically [10] that in the DA for regression context, the models can be sensitive to feature scale differences.

In this work, we propose to take the linear prediction layer g_β into account for the distribution alignment in domain adaptation regression problems. The proposed research is motivated by the question *How to find a feature space, on which a shared linear regressor can easily learn*?

Fortunately, for the linear regression problem, a closed-form solution exists and is well-studied. Given the feature Z and regression ground truth label Y, the problem of estimating the parameter β for a linear layer $Y = Z\beta$ has the ordinary least-squared (OLS) closed-form solution [21]:

$$\hat{\beta} = (Z^T Z)^{-1} Z^T Y \tag{4}$$

where $(Z^T Z)^{-1} \in \mathbb{R}^{p \times p}$ is the inverse of the Gram Matrix. Entries are then the inner products of the basis functions of the finite-dimensional subspace. $Z^T Y \in \mathbb{R}^{p \times N_r}$ projects features to the label space.

The final linear prediction layer g_β is shared by the source and target domains. Therefore, the estimated value from the two domains should be similar $\hat{\beta}_s \sim \hat{\beta}_t$ where

$$\begin{aligned} \hat{\beta}_s &= (Z_s^T Z_s)^{-1} Z_s^T Y_s, \\ \hat{\beta}_t &= (Z_t^T Z_t)^{-1} Z_t^T Y_t. \end{aligned} \tag{5}$$

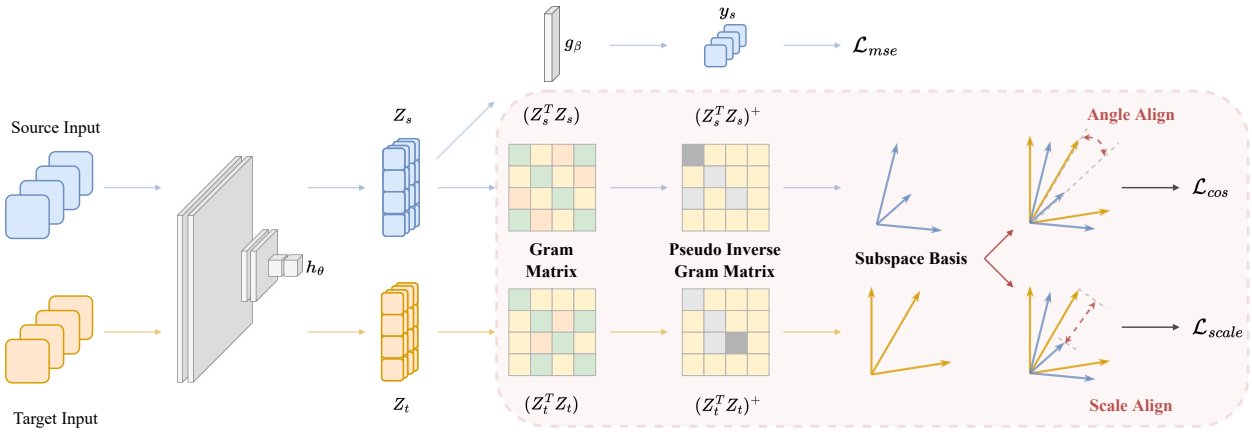
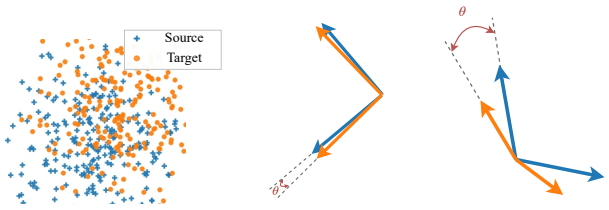


Figure 2. An overview of the proposed DARE-GRAM approach for domain adaptive regression problems. Rather than aligning the features Z, we align the inverse Gram matrix, which is motivated by the ordinary least square solution. To achieve this, we compute the pseudo-inverse Gram Matrices for source and target features and align their angle and scale.

Most of the previous DAR works regularize the neural network by minimizing the distance between source and target data in the feature representation subspace of Z, i.e. aligning Z_s and Z_t. However, because of the inverse operation in Equation 5, even if the distance between Z_s and Z_t is small, the distance in terms of $(Z^T Z)^{-1}$ can be large, as seen in Figure 3. This can further lead to distinct $\hat{\beta}_s$ and $\hat{\beta}_t$ and makes it potentially infeasible to learn a common regressor that performs well for both domains. Given this observation and motivated by the closed-form OLS solution, we propose to focus on the subspace of inverse Gram matrix $(Z^T Z)^{-1}$ for the alignment. More specifically, we propose to align the angle between the source and target pseudo-inverse Gram Matrix, formed by a subset of the eigenspace. In addition, we propose to ensure the same scale of Z for the source and target reflected by the Gram matrix $(Z^T Z)$ by minimizing the distance between selected eigenvalues of both domains.



(a) Two distributions (b) Z (c) $(Z^T Z)^{-1}$

Figure 3. Illustration of the impact the inverse Gram operation in the OLS solution. (a) Assume that the features of the source and target follow two Gaussian distributions with slightly different mean and variance. (b) Under the representation subspace distance, the feature subspaces of Z_s and Z_t are well aligned with a very small angle difference. (c) However, because of the inverse Gram operation, the difference in terms of the inverse Gram matrix $(Z^T Z)^{-1}$ can still be large in terms of both angle and scale.

3.3. Angle Alignment for Gram Matrix Inverse

The first term in Equation 5 concerns the Gram matrix. The Gram matrix is sometimes regarded as a style representation as it calculates the correlations between the different features. It can also be seen as an unbiased empirical estimate of the MMD with a quadratic kernel [19]. The inverse operation is also essential because it first relates the variance of the unbiased estimator β to the eigenvalues of $(Z^T Z)$. Particular attention must be paid to the small eigenvalues, which have a maximum inflationary effect on the variance of the least squares estimator by significantly destabilizing the estimator when it approaches zero. Secondly, the ill-conditioned Gram matrix motivates using a low-rank inverse approximation [11], which allows obtaining regularised basis to be aligned for the source and target domain.

However, such an alignment is non-trivial because the Gram matrix can be non-invertible in deep learning models. During training, the batch size b is generally smaller than the embedding dimension p. Given a feature matrix $Z \in \mathbb{R}^{b \times p}$, with $b < p$, the Gram Matrix $(Z^T Z) \in \mathbb{R}^{p \times p}$, has rank r smaller or equal to b. Hence the Gram Matrix is not fully ranked and thus not invertible. The Moore-Penrose pseudo-inverse in this case can generalize the concept of matrix inverse when the matrix may not be invertible.

We propose to consider only a selected subspace of the Gram matrix to solve this problem. As not all basis vectors contribute equally, the basis vectors with the highest eigenvalues are the most influential. Therefore, we only consider the most dominant basis vectors in the alignment process.

This step has two main objectives : (i) maximize the mutual information between the two distributions by considering only a selected subset, (ii) avoid numerical instability when not considering degenerate eigenspace.

Concretely, given the singular value decomposition (SVD) [65] of the feature matrix Z defined by $Z = UDV^T$. The Gram matrix (Z^TZ), can be decomposed using the SVD of Z as :

$$\begin{aligned}(Z^TZ) = (UDV^T)^T(UDV^T) = V\Lambda V^T,\\ \lambda_k := \Lambda_{k,k} = D^2_{k,k} \quad \text{for} \quad k = 1, ..., p.\end{aligned} \tag{6}$$

where the orthogonal matrix $V \in \mathbb{R}^{p\times p}$ is identical to the matrix in the SVD of Z and $\Lambda \in \mathbb{R}^{p\times p}$ is the diagonal matrix containing the squared eigenvalues of Z.

Given the ordered eigenvalues of the Matrix (Z^TZ) $\lambda_1 \geq ... \geq \lambda_k \geq ... \geq \lambda_p \geq 0$, the Moore-Penrose pseudo-inverse [54] can be derived by discarding the singular values that are below λ_k and treating them as zero. The pseudo-inverse of (Z^TZ) can be expressed as:

$$(Z^TZ)^+ = V^T\Lambda^+V = V^T \left(\begin{array}{ccc|c} \frac{1}{\lambda_1} & & & \\ & \ddots & & 0 \\ & & \frac{1}{\lambda_k} & \\ \hline & 0 & & 0 \end{array}\right) V \tag{7}$$

The operation is equivalent to removing the dimensions with the largest singular value in the inverse matrix. This is in line with [9] as it has been shown that penalizing high eigenvalues is beneficial in domain adaptation.

The selection of k (the number of principal components used) can be achieved through a threshold on the cumulative sum of the eigenvalues of (Z^TZ). Since the smaller eigenvalues do not contribute significantly to the cumulative sum, the corresponding principal components may be continued to be dropped as long as the desired threshold limit is not exceeded. Given λ_s and λ_t respectively the eigenvalues of the matrix $(Z_s^TZ_s)$ and $(Z_t^TZ_t)$, the goal is to find k, s.t.

$$\frac{\sum_{i=0}^{k}\lambda_{s,i}}{\sum_{i=0}^{p}\lambda_{s,i}} > T \quad \text{and} \quad \frac{\sum_{i=0}^{k}\lambda_{t,i}}{\sum_{i=0}^{p}\lambda_{t,i}} > T, \tag{8}$$

where T is a threshold controlling the proportion of explained variance by the first k principal components. In the following, the pseudo-inverse with respect to k of the Gram matrix for source and target is denoted as $G_s^+ = (Z_s^TZ_s)^+$ and $G_t^+ = (Z_t^TZ_t)^+$, respectively. Following [10], the cosine similarity is used to calculate the angle difference between source and target. Unlike previous methods, the angle calculation directly uses the column space of G_s^+ and G_t^+, forming a subspace of $\mathbb{R}^p$ spanned by the column vectors of G_s^+ and G_t^+. A direct measurement of the principal angles is defined as follows:

$$\cos(\theta_i^{S\leftrightarrow T}) = \frac{G_{s,i}^+ \cdot G_{t,i}^+}{\|G_{s,i}^+\| \cdot \|G_{t,i}^+\|} \tag{9}$$

where $i \in [1, p]$, and G_i^+ represent the ith column of the inverse Gram matrix G^+. The cosine similarity between the span of the subspace for both the source and target feature are stored in $M = [\cos(\theta_1^{S\leftrightarrow T}), \ldots, \cos(\theta_p^{S\leftrightarrow T})]$. The loss to align the selected basis from the pseudo-inverse of the Gram matrix can be written as:

$$\mathcal{L}_{cos}(Z^S, Z^T) = \|\mathbb{I} - M\|_1^1 \tag{10}$$

with $\mathbb{I}$ a vector of ones, of shape p. Minimizing the above term maximizes the cosine similarity between the source and target representation subspace by reducing the angle between the basis of both domains.

Discussion The proposed method is also more robust and stable compared to the direct feature alignment of Z (*e.g.* RSD [10]). An important difference between RSD and DARE-GRAM lies on the choice of subspace for the alignment. RSD relies on the U basis derived from the SVD decomposition of Z. However, the vectors U are first, not unique for a matrix with repeated singular values and, secondly, may be numerically unstable since the gradient depends on $\frac{1}{\lambda_i - \lambda_j}$. Morever, A drawback of RSD is that a large batch size $b \geq p$ can result in full space, causing the principal angles(RSD [10]-Eq.2) between two subspace to become zero. In this case, no alignment can be performed by RSD. Our method does not have this drawback.

3.4. Scale Alignment

Preserving the source feature scale is critical in domain adaptive regression problems [10]. In addition to the angle alignment presented in the previous section, we propose to explicitly align the scale of the target subspace to the source.

More specifically, the scale of the matrix Z can be estimated by its trace norm $\|Z\|_1 = \text{Tr}(\sqrt{Z^TZ}) = \sum_{i=1}^{N}\sqrt{\lambda_i}$, where the last term is the sum of the singular values of Z. The scale of Z is therefore the sum of the diagonal elements of the Gram-Matrix. The scale distance between source and target feature is regularized by minimizing the difference between the k-principal eigenvalues:

$$\mathcal{L}_{scale}(Z^S, Z^T) = \|\lambda_{s,i=1,\ldots,k} - \lambda_{t,i=1,\ldots,k}\|_2. \tag{11}$$

Unlike the previous methods [10], which explicitly avoid aligning the feature scale, the pseudo-inverse Gram columns that form the basis of our subspace are not necessarily orthonormal. Therefore, matching the source and target basis scale is also essential to complete the alignment process. As a note, the eigenvectors from the SVD decomposition are orthonormal, and the length of the vectors is fixed and set to one, as shown in Figure 3(b).

3.5. Overview

Combining our angle alignment for the inverse gram and scale alignment, the total loss used for the end-to-end training can be written as:

$$\mathcal{L}_{total}(Z^S, Z^T) = \mathcal{L}_{src} + \alpha_{cos}\mathcal{L}_{cos}(Z^S, Z^T) + \gamma_{scale}\mathcal{L}_{scale}(Z^S, Z^T), \quad (12)$$

where $\alpha_{cos}, \gamma_{scale}$ are hyper-parameters controlling the effect of the angle and scale alignment. An overall of our method is presented in Figure 2.

4. Experiments

4.1. Experimental setup

We evaluate our proposed method on three domain adaptations for regression benchmark datasets: dSprites [48], MPI3D [22] and Biwi Kinect [15].

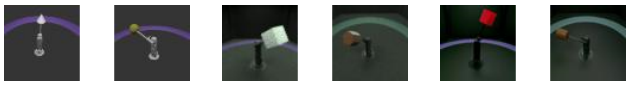
dSprites [48] is a synthetic 2D dataset generated from five ground truth independent latent factors. Following common practice [10], we treat the three variants of the datasets as three different domains. They are generated by adding Color (**C**) or background noise such as Scream (**S**) and Noise (**N**), shown in Figure 4. These three domains comprise 737,280 images each. Dsprites can be used as a benchmark for regression domain adaptation, especially if we consider scale, position X, and Y. Similarly to the setup in [10], the orientation factor is excluded from consideration. We evaluate all methods on the three sub-regression tasks on six adaptation directions: **C** $\rightarrow$ **N**, **C** $\rightarrow$ **S**, **N** $\rightarrow$ **C**, **N** $\rightarrow$ **S**, **S** $\rightarrow$ **C**, and **S** $\rightarrow$ **N**.

(a) Noise (b) Color (c) Scream

Figure 4. Sample example of different domains in dSprites.

MPI3D [22] is a benchmark dataset that consists of 1,036,800 examples of 3D objects from three different domain : Toy (**T**), RealistiC (**RC**) and ReaL (**RL**), as shown in Figure 5. This real-world robotics dataset allows the investigation of the domain gap between real data and simulated ones. This dataset was recorded in a controlled environment, defined by seven factors of variation such as object color, shape, size and position, camera height, background color, and two degrees of freedom of motion of a robotic arm The task is to predict these intrinsic factors from the input image. For this paper, we evaluate our method on six transfer tasks: **RL** $\rightarrow$ **RC**, **RL** $\rightarrow$ **T**, **RC** $\rightarrow$ **T**,**RC** $\rightarrow$ **RL**, **T** $\rightarrow$ **RL** and **T** $\rightarrow$ **RC**. We only considered the two regression tasks, rotation about a vertical and horizontal axis.

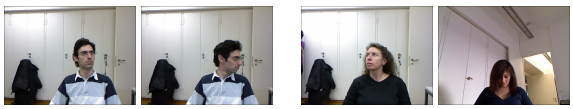
Biwi kinect [15] is a real-word dataset containing over 15K images of 20 people, 6 Females (**F**) with 5874 images and



(a) Toy (b) Realistic (c) Real

Figure 5. Sample example of different domains in MPI3D.

14 Males (**M**) with 9804 images, recorded with a Microsoft Kinect sensor while turning their heads around freely. The example images are shown in Figure 6. The three factors of variations used to evaluate our method are yaw, pitch, and roll angles. We evaluate our method on two transfer tasks: **M** $\rightarrow$ **F** and **F** $\rightarrow$ **M**.



(a) Male (b) Female

Figure 6. Sample example of different domains in Biwi kinect

Evaluation metrics. Following previous works [10, 38], Mean Absolute Error (MAE) is used as our evaluation metric across all the regression tasks. Each experiment is repeated three times, and the average results are reported.

Implementation Details. A pre-trained ResNet-18 [27] on ImageNet is used as the backbone for all methods. For all the experiments, the different tasks share the same encoder but a separated single linear regressor with a Sigmoid activation function. The source and target labels were scaled in the range $[0, 1]$ to eliminate the effects of diverse scales in regression values. We use the SGD [56] optimizer with a momentum of 0.9. The weight decay is set to $1e^{-3}$ for the loss optimization. The newly added layers are trained with a learning rate ten times that of the pre-trained layers, which is initialized to $\eta_0 = 1e^{-2}$. We further adopt the same learning rate scheduler $\eta = \eta_0 \cdot (1 + 0.0001 \cdot p)^{-0.75}$ as [17, 45], where p is the number of iterations changing from 0 to the maximum number of iterations. The images are resized to 224 × 224 and concatenated into batches of size $b = 36$. The number of iterations was set as in [10] to 20,000, 10,000, and 1,500 iterations for dSprites, MPI3D, and Biwi Kinect, respectively. These setup choices are identical to RSD [10]. An NVIDIA RTX 3090 GPU was used for all the experiments.

Compared Methods We compare our method with a range of adaptation methods: (i) Domain Adaptation via Transfer Component Analysis (**TCA**) [51] (ii) Maximum Classifier Discrepancy (**MCD**) [57] (iii) Joint Distribution Optimal Transportation for Domain Adaptation (**JDOT**) [13] (iv) Adaptive Feature Norm (**AFN**) [73] (v) Deep Adaptation Network (**DAN**) [44] (vi) Deep Adaptation Neural Network (**DANN**) [18] and (vii) Representation Subspace Distance for Domain Adaptation Regression (**RSD**) [10].

Method	C → N	C → S	N → C	N → S	S → C	S → N	Avg
Resnet-18 [27]	0.94	0.90	0.16	0.65	0.08	0.26	0.498
TCA [51]	0.94	0.87	0.19	0.66	0.10	0.23	0.498
MCD [57]	0.81	0.81	0.17	0.65	0.07	0.19	0.450
JDOT [13]	0.86	0.79	0.19	0.64	0.10	0.23	0.468
AFN [73]	1.00	0.96	0.16	0.62	0.08	0.32	0.523
DAN [44]	0.70	0.77	0.12	0.50	0.06	0.11	0.377
DANN [18]	0.47	0.46	0.16	0.65	**0.05**	0.10	0.315
RSD [10].	0.31	0.31	0.12	0.53	0.07	0.08	0.237
DARE-GRAM (ours)	**0.30**	**0.20**	**0.11**	**0.25**	**0.05**	**0.07**	**0.164**

Table 1. Comparisons with previous works on the dSprites regression tasks. All results are shown in sum of MAE with the ResNet-18.

Methods	RL → RC	RL → T	RC → RL	RC → T	T → RL	T → RC	Avg
Resnet-18 [27]	0.17	0.44	0.19	0.45	0.51	0.50	0.377
TCA [51]	0.17	0.42	0.19	0.42	0.50	0.50	0.373
MCD [57]	0.13	0.40	0.15	0.45	0.52	0.50	0.358
JDOT [13]	0.16	0.41	0.16	0.41	0.47	0.47	0.353
AFN [73]	0.18	0.45	0.20	0.46	0.53	0.53	0.390
DAN [44]	0.12	0.35	0.12	0.27	0.40	0.41	0.278
DANN [18]	0.09	0.24	0.11	0.41	0.48	0.37	0.283
RSD [10].	**0.09**	0.19	**0.08**	0.15	0.36	0.36	0.205
DARE-GRAM (ours)	**0.09**	**0.15**	0.10	**0.14**	**0.24**	**0.24**	**0.160**

Table 2. Comparisons with related works on the MPI3D regression tasks. All results are shown in sum of MAE with the ResNet-18.

4.2. Results

Evaluation on dSprites: As shown in Table 1, our model achieves the best performance among all competing methods. Specifically, our method outperforms the previous state-of-the-art regression-based method RSD [10] by 30.8% in terms of average MSE over all directions. On the three difficult adaptation directions **C** → **N**, **C** → **S**, **N** → **S**. **DARE-GRAM** also improves the performance over the RSD. The improvement is especially significant on the direction **C** → **S** and **N** → **S**. The improvement is by 33.3% and 52.8%, respectively. **Evaluation on MPI3D:** We further evaluate the effectiveness of our method on this more complex simulation-real data set. As shown in Table 2, in average, our method outperforms all previous methods. The improvement over the previous state-of-the-art RSD is more than 21.9%.The improvement is especially significant in the four hard adaptation directions **T** → **RL**, **T** → **RC**, **RL** → **T**, and **RC** → **T**. The performance is comparable with RSD on the **RC/RL** pair. This might be because the domain gap is relatively small between the pair and the performance (≈ 0.1) could be close to saturation.

Evaluation on Biwi Kinect: Given the much smaller size of the dataset (15,000 images compared to the number of samples in the scale of millions in the other two datasets), the Biwi Kinect regression task is particularly challenging. Additionally, the high imbalance between the two domains and the lack of separation of training and testing sets makes it more difficult and closer to real-world scenarios for DAR. The results reported in Table 3 demonstrate that our model can also consistently improve over previous methods on both directions on this more challenging task.

Method	M → F	F → M	Avg
Resnet-18 [27]	0.29	0.38	0.335
TCA [51]	0.31	0.39	0.350
MCD [57]	0.31	0.37	0.340
JDOT [13]	0.29	0.39	0.340
AFN [73]	0.32	0.41	0.365
DAN [44]	0.28	0.37	0.325
DANN [18]	0.30	0.37	0.335
RSD [10]	0.26	0.30	0.280
DARE-GRAM (ours)	**0.23**	**0.29**	**0.260**

Table 3. Comparisons with previous works on Biwi Kinect.

The performance improvement on the three datasets of very different natures demonstrates the effectiveness of our proposed method.

4.3. Discussion and Analysis

To provide more insights on the proposed Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices, we provide a detailed analysis of the different components of the methodology.

Method	C → S	N → S
Resnet-18 (source only)	0.90	0.65
RSD	0.31	0.53
Angle Alignment for Gram	0.88	0.55
Angle Alignment for truncated Gram	0.89	0.52
Angle Alignment for Gram Inverse (ours)	0.27	0.36
Scale Alignment (ours)	0.23	0.60
DARE-GRAM (ours, angle + scale)	**0.20**	**0.25**

Table 4. Ablation study of different components in our proposed method on C → S and N → S task from dSprites. All results are shown in sum of MAE.

Angle Alignment and Scale Alignment In the first ablation, we study the impact of our angle alignment on the inverse Gram matrix and our scale alignment on the eigenvalues. The **C** → **S** and **N** → **S** tasks in dSprites are used for the ablation here. As shown in Table 4, both components of the proposed methodology are able to improve over the baseline. Minimizing the angle between the pseudo-inverse of the gram Matrix can reduce the MSE over the source-only baseline by 70% on the **C** → **S** task. We also compare the alternate angle alignment on the Gram matrix and truncated Gram matrix without considering the inverse. The MAE in both cases is significantly worst than our inverse version. This demonstrates the significant impact of the inverse operation (as in Equation 5) on the regression layer and verifies our motivation for considering the closed-form solution of OLS regression problems. In addition, as discussed in Section 3.4, the scaling constraints provide essential additional supervision on the alignment and further improve the model performance. Both terms are effective in improving the performance.

Effect of a Larger Batch Size Given the direct relationship between the number of samples in a batch to the variance of the estimated β in Equation 5, we studied the effect of the different training batch sizes in Figure 7 (using the same hyperparameters). As the batch size increases, our approach results in lower MAE. Furthermore, the RSD approach is more sensitive to the batch size and leads to numerical errors for batches bigger than 64. In this paper, we used only the same batch size of 36 to have a fair comparison with RSD. However, better results can be achieved with our method by further increasing the batch size to 256.

Effect of alignment factors: We conduct additional experiments to evaluate the impact on the performance when using different values of hyperparameter α_{cos}, γ_{scale} and the threshold T. As shown in Figure 8, results confirm that our method is not sensitive to hyperparameters.

Alignment performance of Z and $(Z^TZ)^{-1}$ To further validate the proposed method, we examined the cosine similarity of the k-principal components of Z_s, Z_t, as well as $(Z_s^TZ_s)^{-1}, (Z_t^TZ_t)^{-1}$, after applying our proposed method and RSD. The results are shown in Table A2 in the supplementary material. Our results demonstrate that aligning

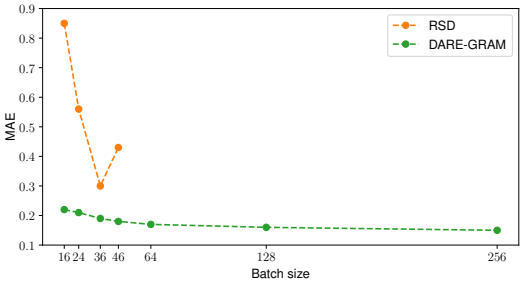
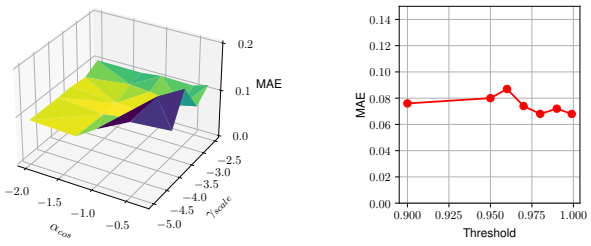


Figure 7. Batch size sensitivity on transfer task **C** → **S**. For RSD, larger batch sizes lead to numerical errors, thus results not shown. Our method DARE-GRAM is able to achieve better performance with large batch sizes because the feature correlations can be better captured by the Gram matrix with larger number of samples.



(a) Impact of α_{cos} and γ_{scale} in log scale on the sum of MAE.

(b) Impact of Threshold T on the sum of MAE

Figure 8. Hyperparameter sensitivity of our method on dSprites transfer task **S** → **N**.

Z by RSD can lead to poorly aligned inverse Gram matrix $(Z^TZ)^{-1}$. In contrast, by aligning $(Z^TZ)^{-1}$, our proposed method leads to a well-aligned Z, providing further empirical support for the effectiveness of our proposed method.

5. Conclusion

In this paper, we have presented a new domain adaptative regression method called DARE-GRAM. We tackled the domain adaptation for regression problems from a different perspective analyzing the ordinary least square solution to the linear regressor in the deep domain adaptation context. Rather than aligning the original feature embedding space, we aligned a selected subspace of the pseudo-inverse Gram matrix, leveraging the pseudo-inverse low-rank property. Finally, two new regularization terms were proposed to align the scale and angle in a selected subspace generated by the Gram matrix of the two domains. Experimental results show that DARE-GRAM achieves significant improvement in three benchmark regression datasets while ensuring the stability and robustness of the training procedure.

Acknowledgments: This work was supported by the Swiss National Science Foundation under Grant PP00P2_176878.

References

[1] Hiroyasu Akada, Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Self-supervised learning of domain invariant features for depth estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3377–3387, January 2022. 2

[2] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK users' guide*. SIAM, 1999. 2

[3] Yiwei Bao, Yunfei Liu, Haofei Wang, and Feng Lu. Generalizing gaze estimation with rotation consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4207–4216, June 2022. 2

[4] Yiwei Bao, Yunfei Liu, Haofei Wang, and Feng Lu. Generalizing gaze estimation with rotation consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4207–4216, 2022. 2

[5] Deblina Bhattacharjee, Martin Everaert, Mathieu Salzmann, and Sabine Süsstrunk. Estimating image depth in the comics domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2070–2079, January 2022. 2

[6] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017. 3

[7] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010. 2

[8] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2090–2099, 2019. 2

[9] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pages 1081–1090. PMLR, 2019. 2, 5

[10] Xinyang Chen, Sinan Wang, Jianmin Wang, and Mingsheng Long. Representation subspace distance for domain adaptation regression. In *International Conference on Machine Learning*, pages 1749–1759. PMLR, 2021. 1, 2, 3, 5, 6, 7

[11] Julianne Chung, Matthias Chung, and Dianne P O'Leary. Optimal regularized low rank inverse approximation. *Linear Algebra and its Applications*, 468:260–269, 2015. 4

[12] Corinna Cortes and Mehryar Mohri. Domain adaptation in regression. In *International Conference on Algorithmic Learning Theory*, pages 308–323. Springer, 2011. 1, 2

[13] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation, 2017. 6, 7

[14] Antoine de Mathelin, Guillaume Richard, Francois Deheeger, Mathilde Mougeot, and Nicolas Vayatis. Adversarial weighting for domain adaptation in regression, 2021. 2

[15] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *Int. J. Comput. Vision*, 101(3):437–458, February 2013. 6

[16] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013. 2, 3

[17] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 2, 6

[18] 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. 6, 7

[19] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 3, 4

[20] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 1

[21] Arthur Stanley Goldberger et al. Econometric theory. *Econometric theory.*, 1964. 3

[22] Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 6

[23] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012. 3

[24] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011. 3

[25] 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. 3

[26] Zidong Guo, Zejian Yuan, Chong Zhang, Wanchao Chi, Yonggen Ling, and Shenghao Zhang. Domain adaptation gaze estimation by embedding with prediction consistency. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6, 7

[28] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell.

Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018. 2

[29] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. *arXiv preprint arXiv:2204.13132*, 2022. 2

[30] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2006. 2

[31] Junguang Jiang, Yifei Ji, Ximei Wang, Yufeng Liu, Jianmin Wang, and Mingsheng Long. Regressive domain adaptation for unsupervised keypoint detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6780–6789, 2021. 2

[32] Nikolai Kalischek, Jan D Wegner, and Konrad Schindler. In the light of feature distributions: moment matching for neural style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9391, 2021. 3

[33] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 2

[34] Donghyun Kim, Kaihong Wang, Kate Saenko, Margrit Betke, and Stan Sclaroff. A unified framework for domain adaptive pose estimation. *Proceedings of European Conference on Computer Vision*, 2022. 2

[35] Antti Knowles and Jun Yin. Eigenvector distribution of wigner matrices. *Probability Theory and Related Fields*, 155(3):543–582, 2013. 2

[36] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1482–1491, 2021. 2

[37] Hui Li, Zidong Guo, Seon-Min Rhee, Seungju Han, and Jae-Joon Han. Towards accurate facial landmark detection via cascaded transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4176–4185, 2022. 1

[38] Wanhua Li, Xiaoke Huang, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Learning probabilistic ordinal embeddings for uncertainty-aware regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13896–13905, June 2021. 6

[39] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 3

[40] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation, 2016. 2

[41] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6758–6767, 2019. 1

[42] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8003–8013, June 2022. 2

[43] Shao-Yuan Lo, Wei Wang, Jim Thomas, Jingjing Zheng, Vishal M Patel, and Cheng-Hao Kuo. Learning feature decomposition for domain adaptive monocular depth estimation. *arXiv preprint arXiv:2208.00160*, 2022. 2

[44] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 1, 2, 6, 7

[45] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 1, 6

[46] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms, 2009. 1, 2

[47] Dragos D Margineantu and Thomas G Dietterich. Pruning adaptive boosting. In *ICML*, volume 97, pages 211–218. Citeseer, 1997. 2

[48] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 6

[49] Jaemin Na, Dongyoon Han, Hyung Jin Chang, and Wonjun Hwang. Contrastive vicinal space for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 92–110. Springer, 2022. 2

[50] Takehiko Ohkawa, Yu-Jhe Li, Qichen Fu, Ryosuke Furuta, Kris M Kitani, and Yoichi Sato. Domain adaptive hand keypoint and pixel localization in the wild. In *European Conference on Computer Vision*, pages 68–87. Springer, 2022. 2

[51] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010. 2, 6, 7

[52] David Pardoe and Peter Stone. Boosting for regression transfer. In *ICML*, 2010. 2

[53] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 2

[54] Roger Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge University Press, 1955. 5

[55] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020. 2

[56] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016. 6

[57] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsuper-

vised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 6, 7

[58] Jay Shenoy, James Fong, Jeffrey Tan, Austin Roorda, and Ren Ng. R-slam: Optimizing eye tracking from rolling shutter video of the retina. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4852–4861, 2021. 1

[59] Ankita Singh and Shayok Chakraborty. Deep domain adaptation for regression. In *Development and Analysis of Deep Learning Architectures*, pages 91–115. Springer, 2020. 2, 3

[60] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. *Advances in neural information processing systems*, 20, 2007. 2

[61] Baochen Sun and Kate Saenko. Subspace distribution alignment for unsupervised domain adaptation. In *BMVC*, volume 4, pages 24–1, 2015. 3

[62] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 3

[63] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised domain adaptation for depth prediction from images. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2396–2409, 2019. 2

[64] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 2

[65] Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996. 5

[66] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 2

[67] Boyu Wang, Jorge Mendez, Mingbo Cai, and Eric Eaton. Transfer learning via minimizing the performance gap between domains. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[68] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, June 2022. 2

[69] Qin Wang, Gabriel Michau, and Olga Fink. Domain adaptive transfer learning for fault diagnosis. In *2019 Prognostics and System Health Management Conference (PHM-Paris)*, pages 279–285, 2019. 1

[70] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16643–16653, June 2021. 3

[71] Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), jul 2020. 2

[72] Jun Wu, Jingrui He, Sheng Wang, Kaiyu Guan, and Elizabeth Ainsworth. Distribution-informed neural networks for domain adaptation regression. In *Advances in Neural Information Processing Systems*, 2022. 2

[73] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1426–1435, 2019. 6, 7

[74] Makoto Yamada, Leonid Sigal, and Yi Chang. Domain adaptation for structured regression. *International journal of computer vision*, 109(1):126–145, 2014. 2

[75] Makoto Yamada, Leonid Sigal, and Michalis Raptis. No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation. In *European Conference on Computer Vision*, pages 674–687. Springer, 2012. 2

[76] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019. 1

[77] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 1, 2

[78] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424, 2021. 2

[79] Youshan Zhang and Brian D Davison. Efficient pre-trained features and recurrent pseudo-labeling in unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2719–2728, 2021. 2

[80] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11656–11665, 2021. 1