MSc Artificial Intelligence
Master Thesis

Multi-domain invariant learning with embedding alignment

by
Sander Adriaan Kohnstamm
10715363

September 28, 2022

48EC
15th December, 2021 - 15th August, 2022

Supervisors:
dr. Gertjan Burghouts,
prof. dr. Cees Snoek
Examiner:
prof. dr. Cees Snoek
Second reader:
Zehao Xiao MSc

Universiteit van Amsterdam

Contents

Abstract

The goal of this thesis is to study and improve a deep neural network's robustness to a shifting domain. In computer vision and object recognition, domains can represent many different groups of images, such as images made at a specific time of day, in a certain style, with a certain kind of weather or from a specific angle. Deep neural networks often struggle when the domain changes between training and testing data. Improving out-of-domain performance within image recognition, improves a neural network's efficiency, application and robustness. We have found that aligning feature representation embeddings for images from different domains can learn a network to be more robust to domain shifts, resulting in higher accuracies when testing on a new domain. We propose an algorithm for a three-domain setting, two of which are labelled. By minimising the element-wise distance between the feature embeddings of two training domains we improve the accuracy on the third target domain. In this thesis, the three domains are represented by three distinct styles of image. We test our proposal in two versions. One version does not use the target domain data, and the other does, but without its labels. These respective forms are called domain generalisation and unsupervised domain adaptation. We provide an analysis of the performance of our proposal on these two different forms and its domain-shifting implications. We achieve a small performance increase for the unsupervised domain adaptation setting and learn the importance of distinctive information in both these situations.

Chapter 1

Introduction

Deep neural networks can be incredibly well optimised for specific data-sets, however, they often struggle when the testing set contains environmental constants that are shifted with respect to the training set. We call these constants domains. For image classification, the domain-specific information often incorporates some form of causal relation or correlation between the subject and its background; for instance birds in the sky versus birds on the ground, a shifting in camera position or a time-of-day difference. The question arises whether the subject is a cause of the context, vice versa or simply a correlation. The goal of this thesis is to study and improve a network's robustness to such a domain shift. We do this because robust transferring of learned information to new domains and data distributions leads to less training time, better understood models and allows for wider access to machine learning models across the field. By training a network not to be swayed by environmental or domain-specific information, it can become more *domain invariant*. A domain invariant network produces the same output for a given class no matter what domain.

In 2010 researchers clearly defined the problem of domain shift for the first time by studying the domain shift problem in object recognition [37]. They introduce the Office-31 data-set, where domains are represented by different styles of photos of office objects. To further illustrate what a domain can be, we present figure 1.1 from a similar data-set, which we shall use in this work. In this figure, the domains are represented by the differences between the styles cartoon, photo and painting. The goal will be to improve accuracy on one of the domains, using the other two for training.

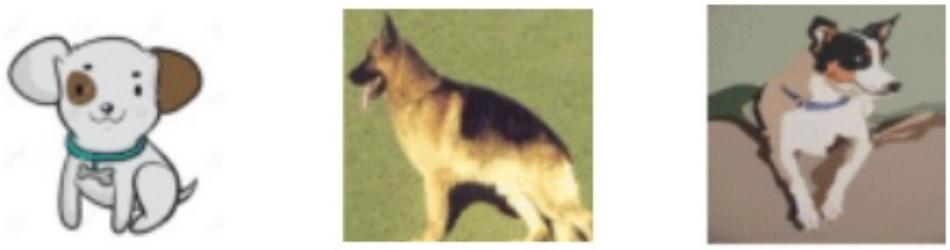


Figure 1.1: An example of three domains in an object recognition task. The object class is dog and the three domains are, from left to right, cartoon, photo and painting. Optimising for domain shift means optimising for the performance on one domain while training on the other domains, where the goal is better generalisation to unseen domains.

Most image classification training has assumed that training data and target data follow the same distribution [5], which is a simplifying assumption that is still present in most learning algorithms. In practical applications, these distributions often differ. Training data for image

classification often have significant biases such as, but not limited to, time of day, orientation or physical background, which do not sustain into the target domain. For instance, when detecting cows on sandy beaches instead of in the training data sets on grassy mountains, performance dropped heavily [4]. The problem is also relevant in medical imaging, where knowledge does not generalise well from patient to patient [32] or from one point of view to the other [6]. Improving on these kinds of situations has positive impacts beyond just accuracy. Transferring knowledge from domain to domain improves efficiency because it enables more applications from the same information.

Currently, the domain shift problem is actively researched. A part of this work assumes the availability of the target domain data, with which a model can be fine-tuned. This is called a domain adaptation problem and has been approached from many angles including retraining with pseudo labels [12], a feature selection method [22], adverserial networks [10], causal inference [28] and many more [11, 16, 27, 53]. Meanwhile, domain generalisation works aim to generalise a network without use of the target domain. Domain generalisation has also been extensively researched, including with causal matching [29], image generation [54], regularising mutual information with a domain invariant source [8] and many more [20, 26, 47]. A common theme in many works for both problem settings is the use of a regulariser which penalises a network for retaining domain-specific information.

We take inspiration from these works, especially from two fields: causality and information theory. From the first, we learn the importance of features that are likely to be domain invariant. A classification head is guaranteed to make a domain invariant prediction if and only if all the features used in this prediction are the full set of domain invariant features. From the second, we learn how we can generate these features that are likely to be domain invariant. Adding a domain invariant regulariser to the loss function is an effective method to promote domain invariance. We shall use such a regulariser to train the network to generate domain invariant features.

1.1 Domain invariance and its variations

We have noted that training a network for *domain invariance* means training a network to be more robust against domain shift. The intended outcome of training networks for domain invariance is to achieve the best possible performance on a target domain with a set of source domains. This problem is separable into two forms. Domain adaptation allows for the usage of the target domains (un)labelled data, whereas domain generalisation does not. In domain adaptation, a network can be trained to classify a specific new domain by using the data of that domain. Domain generalisation, however, aims to generalise a network as broad as possible to handle all possible new domains. Therefore it cannot use the data or the labels of a target domain during training.

Similar restrictions also separate domain adaptation from transfer learning, a subset of domain adaptation. Transfer learning uses the target data and labels to adapt to the new domain. Regarding domain adaptation, we focus on unsupervised domain adaptation, where we do not use the target labels during training, because, in practice, often the data of the target domain is available, but not the labels. We also investigate domain generalisation, because it provides a stepping stone into domain invariance and verification of the proposed solutions functioning.

An overview of the different forms of domain invariance is presented in table 1.1. To recap; in this thesis, we shall focus on domain generalisation and unsupervised domain adaptation. We research both to validate the added value of the target domain data. By investigating these two different views of improving performance in an unlabelled, target domain, we can improve our understanding of image classification within new environments.

	Target domain access		
	Data X_t	Labels Y_t	This thesis
Domain generalisation	No	No	✓
Transfer Learning	Yes	Yes	-
Weakly Supervised Domain Adaptation	Yes	Sparse	-
Unsupervised Domain Adaptation	Yes	No	✓

Table 1.1: Domain invariant learning variations and their respective access to the target distributions.

1.2 Embedding Alignment

We shall use feature embedding alignment to promote performance for domain generalisation and unsupervised domain adaptation for image classification. We do this because aligning feature embeddings from different domains per class can aid the classification head by providing better separable and similarly structured distributions across new domains. We present the embeddings resulting from the network backbone from the training domains and the target domain in figure 1.2. This figure shows the embeddings of the features from the source/training domain on the left and the target domain on the right. The different colours represent the seven different classes: dog (blue), elephant (light blue), giraffe (orange), guitar (light orange), horse (green), house (light green) and person (red). The network that produced these embeddings was trained only on the source domain. The problem: they are very differently distributed and the target domain is not separable.

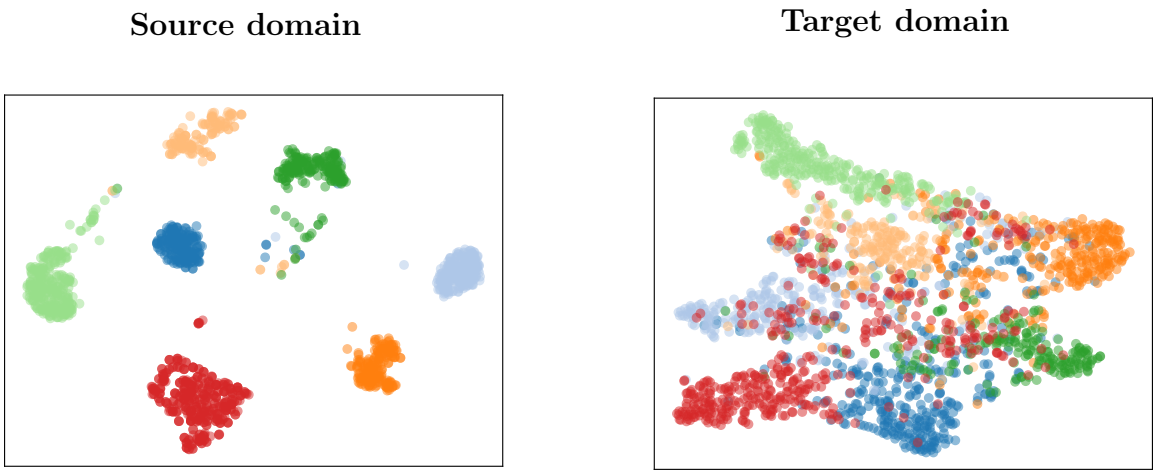


Figure 1.2: t-sne plots of the feature embeddings of seven different classes: dog (blue), elephant (light blue), giraffe (orange), guitar (light orange), horse (green), house (light green) and person (red). These are embedding visualisations of either the training domain data (left) or the unseen target domain data (right) by a network trained on the source domain. The goal is two-fold; to make the source class embeddings separable and make both embeddings as similar as possible across domains.

The features of the classes from the training domains are clearly better separable than the features from the target domain. The goal of domain invariant learning is to have no difference between the distributions of the training domain embeddings and the target domain embeddings for their respective classes. If a network produces the same embedding distribution for each domain, it is domain invariant [53].

Feature embedding alignment has shown promise in unsupervised, few-shot learning [40], transfer learning [56] and many other image classification tasks. Usually, the distances between feature embedding distributions from domain to domain are minimised to enhance the performance on a target domain. A network is trained to embed the distributions from each class closer to the distributions from another domain. The distance between these distributions is often added to the loss function for training. This addition teaches the network to output similar distributions for both inputs from a given class.

In this work, we propose a similar approach. The network will be forced to make the feature embeddings domain invariant by regularising the loss function with a distance measure. Domain invariance often comes at a cost, however. An entirely agnostic model is also domain invariant but cannot provide discriminative predictions. Therefore, it is critical to maintain the network's discriminative ability. We will further explore the theory behind this balance in chapter 2 and propose a solution in chapter 3.

For this thesis, two distance measures that are currently actively researched, as well as a baseline measure, are regarded and validated. We select three distance measures because we would like to compare the two top contenders to each other and a baseline. The first two are the Hilbert Schmidt Independence Criterion (HSIC)[19] and the Wasserstein distance [46] from the optimal transport problem [30]. The baseline measure is the l^2-norm. We have chosen the HSIC for its use in quantifying dependency relationships between distributions. We shall use this quality for quantifying the dependency between distributions of features from the different domains. We have chosen the optimal transport problem because of its efficient ordering of elements per distribution. This quality is important because we hypothesise that the differences between the most similarly embedded data points are the most important to minimise. By monitoring the impact of the distance measures as a regularisation term added to the loss function, we shall test whether we can improve out-of-domain performance by alignment of features in a novel way while maintaining the distinctive embedding information in the model.

1.3 Contribution

The algorithm we introduce in this work is a three-stage method for embedding alignment between three domains. The algorithm works on three domains: two training domains and one unlabelled target domain, as shown in figure 1.3, and it has an application for both domain generalisation and unsupervised domain adaptation. In short, the three stages consist of training the network on the labelled domains, generating pseudo-labels for the target domain and training the same network a second time with the feature embedding distance added to the loss function.

Others have implemented either a three-stage method [12], or feature alignment for domain invariance [22], or optimal transport[12] and HSIC [17]. Our novelty consists of combining these methods with multiple training domains. Using a three-domain setup for feature alignment and the comparison between the chosen distances, are both feature alignment steps for domain invariance which we have not found in the available literature. Figure 1.3 illustrates a simplified version of the setup of this work. The setup concerns two training domains and one target domain, where each domain differs. The domains are represented by the image styles of cartoon, painting and photo.

We hypothesise that we can maximise domain invariance by minimising the distance between the embeddings of different domains. By regulating this behaviour, we can improve out-of-distribution performance. By testing this hypothesis, the contribution of this thesis is two-fold:

1. We propose a novel 3-step feature alignment algorithm using pseudo labels, a distance measure and three domains in unsupervised domain adaptation and analyse the out-of-

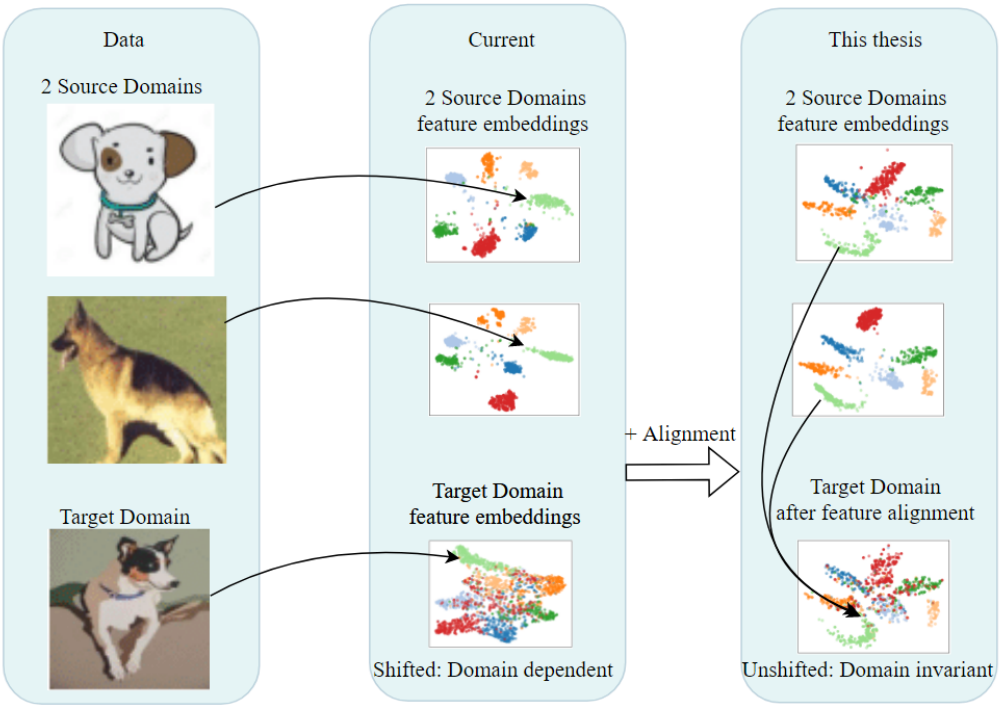


Figure 1.3: Overview of the feature alignment on three domains by introducing pseudo-labels in the unlabelled third target domain. This is proposed in this thesis and its resulting target domain embeddings are shown. The resulting embeddings after feature alignment on the bottom right show better and more domain invariant class structures and more with our method.

distribution accuracy, improved distribution alignment and errors from this algorithm.

2. We analyse the impact and difference in feature alignment in a domain generalisation and unsupervised domain adaptation setting of the three different distance measures with an ablation study.

Chapter 2

Related work

This chapter gives information on the domain generalisation and unsupervised domain adaptation techniques used in this work. We present related work for this field. The reasoning for our approach is supported by the accumulation of the different methods from these works.

2.1 Causality for domain invariance

Viewing domain invariance from a causal perspective can clarify the objective. The objective is to use domain invariant information. If a domain causes information, such as the night causing low light, the dark sky should be neglected by a network, but the object's colour can still be significant. Using the invariance of features for causal discovery and inference was first proposed by [21], which the researchers built on the causal inference work [33]. In their work on causal inference, they introduce *invariant causal prediction* using the example of an environment or domain depicted in figure 2.1. In this figure, Y is the target label to be predicted and X_2 to X_5 are the random variables used for this prediction. They leverage these relations using a linear model and a target response variable. By intervening in the features and monitoring the target response, they find the invariant subset of features, or the direct causal subset $\{X_2, X_4\}$, with which the conditional distribution of the target variable Y remains invariant. Using only these features for their prediction increases out-of-domain performance significantly and opens a new road for causal discovery.

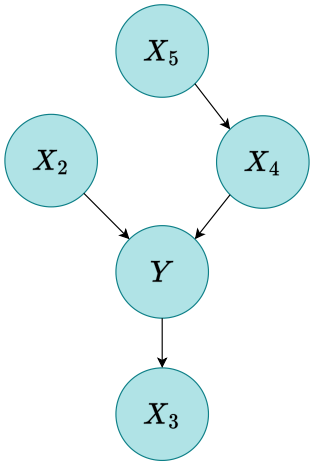


Figure 2.1: example of an environment with random variables X and their causal relations with label Y. [33]

According to their work, if and only if a prediction is based on the complete set of direct causal variables, this prediction is guaranteed to be domain invariant. This complete set of variables is represented by $\{X_2, X_4\}$ in figure 2.1 as they have a direct causal relationship with Y. If all and only direct causal variables are used in a prediction, no other information can be incorporated. For instance, let a picture of a cow with a beach as the background be the input. This background can be represented by X_5. If cows on beaches often are brown instead of black, then colour can be represented by X_4. One would still need to incorporate X_4 in one's prediction as a pink cow is not present in any domain, but just being brown and on a beach does not make the subject a cow. Invariant prediction cannot be guaranteed without a subset of these variables or a prediction based on indirect or anti-causal variables. The set $\{X_3, X_5\}$ in figure 2.1 can contain environmental or domain-specific information, which will influence the out-of-distribution prediction. In our case, the features are not given a priori, so we must learn them. We aim to learn only domain invariant features.

Much research followed up on the invariant causal prediction work [28, 34, 38, 39]. Including multiple continuations of the authors themselves. One approach is to add a regulariser to the loss function [36]. They use a domain label as one of their target labels. The regulariser takes the form of the negative of the prediction error on the domain label information, acting as an anchor during training. Hence the method is called Anchor Regression.

The complete set of direct causal variables cannot be guaranteed when learning features from high-dimensional image data. However, the method from this thesis will steer the embeddings to be domain invariant by penalising domain-specific features that cause a difference between domains for a given class. This penalisation will take a similar form as the Anchor Regression regulariser in that it adds the negative of domain-specific knowledge to the loss function. Backpropagating this loss will force the network to generate a more direct causal set of features.

Notable is the introduction of *invariant risk minimisation* [3], building on this causal inference framework. They state that equivalent to learning features whose correlation with the target variables remain stable is the use of a data representation $\Phi : \mathcal{X} \to \mathcal{H}$ which has a classifier $w : \mathcal{H} \to \mathcal{Y}$ that is optimal for all domains k. Here $\mathcal{X}, \mathcal{H}$ and $\mathcal{Y}$ represent the data, features and labels, respectively. A fair amount of discussion has followed this publication on the merit of invariant risk minimisation relative to the foundational *empirical risk minimisation* principle [2, 35, 45]. This trade-off between empirical and invariant risk minimisation often represents the trade-off between distinctive ability and domain invariance. This thesis also explores the balancing of this trade-off. We use empirical risk minimisation as the baseline and monitor the influence of invariant risk minimisation with a scalar hyperparameter.

2.2 Information theory for domain invariance

From an information theoretical perspective, the most common way of achieving domain invariance is with information bottlenecks [1, 13, 14, 25, 42]. The general idea is to limit the space in a network for information, forcing the network to retain information only present in all data. If the trainable data available for a network is varied enough, only the information invariant to the shifting distributions in the training data will be retained in the network weights.

One of the top performers on a benchmark data set called DomainNet [31] uses the mutual information between the trainable domains and an oracle domain to increase domain invariance [8]. This oracle is entirely domain generalised and optimised for any possible domain. In practice, it is impossible to generate such an oracle, so they substitute it with an ImageNet pre-trained ResNet50, which they assume is generalised enough and can approximate such an oracle. Achieving complete domain invariance in a data-set or network is often a trade-off with domain-specific information. Therefore, this oracle domain has no use on its own. By regulating the loss function with the mutual information between the model and their oracle, they ensure that the model weights favour domain invariant information while also training for classification accuracy. With this method, they generate more domain invariant features and achieve impressive generalisation results.

Because of these results, we implement a similar approach in this thesis. We also aim to influence the learning process with a domain invariant information source. Here, our distance between domains can be viewed as a surrogate for the oracle. It is domain invariant information and has no predictive use on its own. Letting the distance or dependency between the distributions regularise the loss function replaces the mutual information.

2.3 Feature Alignment for domain invariance

Feature alignment for domain invariance has been studied extensively [9, 18, 27, 41]. Various distances were used for the aligning features from a source to a target domain, such as KL-divergence [55], Maximum mean discrepancy [51] and the Wasserstein distance [49]. These approaches are have varying success, shifting from domain to domain and accumulating errors using the predicted pseudo-labels in their training. That is why it is important to have adequate performance on a target domain before using the distance measures and to suppress the possible error accumulation. Using multiple source domains before predicting pseudo labels is hypothesised to increase this performance significantly, thus allowing for a better examination of the feature alignment technique. This examination will be one of our contributions.

Another contribution of this thesis is the comparison of multiple distance measures on their capabilities of generalising and adapting a network to unseen domains using feature alignment. There are many different ways of comparing two distributions with each other. The distance between the two distributions should be differentiable and efficient for many dimensions. These are the criteria to be compatible with our deep learning approach. We have selected the measures l^2-norm, the Hilbert-Schmidt Independence Criterion (HSIC)[19] and the Wasserstein distance [30]. We have chosen the l^2-norm because it is the most simple distance, which serves as a baseline. Next, we explain the choice of the other two distances.

Hilbert-Schmidt Independence Criterion The HSIC is a method of quantifying dependency relations, which we would like to exploit as mentioned in section 2.1. HSIC as a statistical dependence measure can be used for an extensive set of problems such as dimensionality reduction, kernel learning [48] and learning without backpropagation [24]. As a dependence measure, these methods always include a minimisation or maximisation of some causal relationship quantified with HSIC. That is why we propose to utilise the dependency between features of different domains by maximising the HSIC between domains as a regulariser in the loss function.

Other examples of the use of HSIC are in a clustering setting and feature selection. Certain features which minimise the HSIC between each other can be clustered together, or features which minimise the dependence between them and their respective class labels can be used for classification [17]. This learning with HSIC for classification work shows that the criterion can also be incorporated in the loss function where it compares feature-level predictive information. We use this ability to compare features from different domains. This is different from the other use of the HSIC for the distance minimisation between feature and label distribution in the learning without backpropagation work [24].

Optimal transport The optimal transport problem has seen an increase in attention in the last few years as the approximations become more robust[15, 43]. It effectively leverages the difference between domain-specific distributions in [11], but this work concerns a two-domain setting. A semi-supervised approach was also proposed [49, 52]. One of these works [49] also leverages pseudo-labels in the process of feature alignment with the Wasserstein distance. However, using some labelled target domain data to predict the pseudo labels, they position themselves in the weakly supervised domain adaptation position (table 1.1). In contrast, we only use learned information from the source domains to predict the pseudo-labels.

Another implementation of the optimal transport problem for unsupervised domain adaptation [12] generates the coupling matrix between the source and target domain data using optimal transport. This coupling matrix generates the target domains pseudo labels, with which a network is trained. They call this method DeepJDOT, which was an inspiration for this thesis. However, we propose to use the optimal transport problem and its use of the Wasserstein distance in the three-domain setting using predicted pseudo labels.

Chapter 3

Method

In this chapter, we start with some preliminary and technical information on the problem, our proposed method for three domains, the involved distances and implementation. After that, we explain our hypothesis and proposal to test it. Lastly, we give an in-depth explanation of the two variants for respectively domain generalisation and unsupervised domain adaptation.

3.1 Problem and hypothesis

Problem statement

Here, we define the problem. Let our supervised learning problem be $P(X, Y)$ be the joint distribution and $P(Y|X)$ be the posterior we want to learn. The random variables are $x_i \in X$ and $y_i \in Y$. Here, in our case, x_i represents the features resulting from the network backbone. The marginals $P(X)$ and $P(Y)$ contain the input features and labels, respectively. A learning function $f(x) = y$ is learned by optimising a loss function $l(y', y)$. Here the loss l function can be any criterion. The predicted targets are y', and y are the given labels.

Extending this to a multi-domain setting, we observe a set $D = D_1..D_K$ of K different domains. We assume access to the joint distribution $P_n(X, Y)$ of the source domains with $n \in 1..K-1$ and its marginals, together with (X_n, Y_n) and therefore $x_n^i \in X_n$ and $y_n^i \in Y_n$. We also assume some difference between $p_n(X, Y)$ and $p_m(X, Y)$ when $n \neq m$. The goal for out-of-distribution learning is then to learn a function $f(x_k) = y'_k$ with a minimal $l(y'_k, y)$, with k being a target domain.

Domain generalisation is a category that focuses on maximising accuracy on any *unspecified* domain without x_k or y_k. In domain adaptation, there is a sole focus on improving accuracy on a new *specific* target domain, with or without the use of y_k. We can separate domain generalisation and domain adaptation by toggling the access to x_k during training.

To summarise, given multiple labelled source domain(s), we want to learn a function to predict these labels that are invariant to the changes between the domains and thus generalise well to an unseen target domain. The training utilises the data distribution of the target domain as an input variable or not. These variations are named domain adaptation and domain generalisation, which we will both study in this thesis. We focus on two source domains and one unlabelled target domain.

Hypothesis

Minimising the difference between feature embeddings from different domains is an essential step toward domain invariance. We hypothesise that by minimising the distance between feature embedding distributions of different domains, we minimise the difference between these

distributions and maximise domain invariance. Once a network produces identical embedding distributions per class for different domains, this network is entirely domain invariant [53]. A fully domain invariant network backbone can optimise its classification on the features that remain present in all domains, which improves out-of-distribution performance [21]. Therefore, minimising the distance between the domain-specific feature embeddings of each class maximises domain invariance of the network, which can boost domain generalisation and unsupervised domain adaptation performance if balanced with distinctive learning.

3.2 Two algorithms

We start with an overview of our proposed algorithms in figure 3.1, which also highlights the two contributions in the bottom blocks. The algorithm is used in both the domain generalisation (DG) and unsupervised domain adaptation (UDA) settings. These variations are marked on the far left with DG and UDA, respectively. We shall start with the similarities between the algorithms to avoid double information. These similarities concern some algorithmic steps and the involved model components. The details of the two individual algorithm variations are given in the next sections.

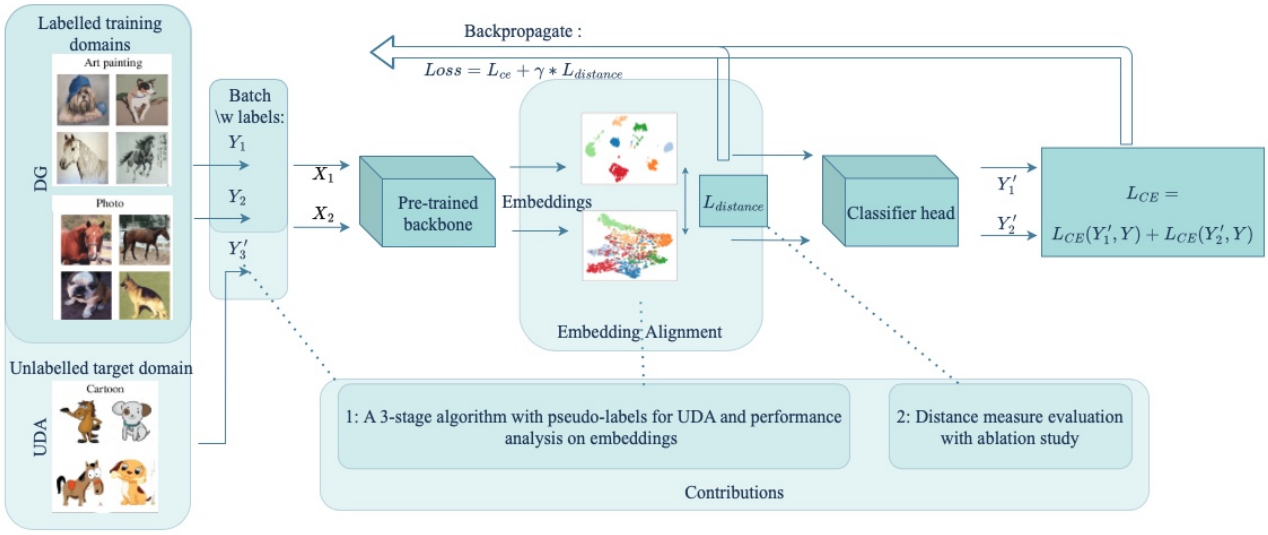


Figure 3.1: Combined illustration of the algorithm for domain generalisation (DG) and unsupervised domain adaptation (UDA). Two batches of the domains move from left to right. The distance is calculated between the embeddings from the respective domains and added to the loss function. Our contributions are highlighted at the bottom, connected to their respective implementation locations. The first contribution is split between the algorithm and its performance analysis.

The pre-trained backbone of a ResNet18, trained on ImageNet, serves as a domain invariant base embedder [8]. The backbone can be adapted and generalised to our domain and test our hypothesis without making the iterations take much time. The classifier head consists of two hidden linear layers with two ReLU activation functions to promote non-linear prediction. We expect non-linear dependency between the embeddings because of the embeddings shown in figure 1.2.

The network's input consists of two batches X_1 and X_2 from two different domains as input. The data loader can sample X_1 and X_2 from the training domains for the domain generalisation setup or from one training and the target domain for the unsupervised domain adaptation setup. The method uses batches with the same class order. The classes can be shuffled within a batch,

but the classes from batch 1 are mirrored with the classes in batch 2. By mirroring the classes, the distribution alignment is not performed across classes. Aligning the embeddings of different classes would impair training because the distributions of different classes are supposed to be separate in the feature space.

The embeddings of the input batches produced by the backbone are extracted, and their element-wise distance is measured between them with one of the proposed distance measures. This loss will be noted as $L_{distance}$. The classifier head predicts two sets Y_1' and Y_2' of target labels with the embeddings of each input domain. The error of these predictions is calculated with a cross-entropy loss as the criterion. We denote this loss as L_{CE}, consisting of the loss of both inputs. Adam is used as an optimiser for its resistance to variations in new environments [23].

The distance is added to the cross entropy loss to penalise the model for producing features which differ too much between domains. A hyperparameter γ is added to this function to balance the penalisation. The full loss function then takes the following form:

$$Loss = L_{ce} + \gamma * L_{distance}. \tag{3.1}$$

This loss is back-propagated through the network. Here, the cross entropy loss affects the entire network, and the distance loss only affects the backbone.

Our proposed method empirically calculates the distances between each of the two respective domains' batches during training. Notable is that these batch distances are calculated element-wise. Only the elements on the same index in the two batches are compared. Because the complete domain data sets are sampled in the same order, the same element-wise comparison holds for the total domain distributions. An exemption is the optimal transport distance, which has a reshuffling of the batches to find optimal matches for each index. We provide more information on ordering these distributions within the batch for each implementation and their differences per algorithm in the next sections Algorithm 1 & 2.

We test domain generalisation and unsupervised domain adaptation on a third, unlabelled domain. For the domain generalisation, this domain is entirely unseen, but in the unsupervised domain adaptation setting, the unlabelled data of this domain is leveraged. The performance in this third target domain is our main focus for the ablation study. We also detail the use of the target domain data explanation sections Algorithm 1 & 2.

Algorithm 1: Domain generalisation

The setup for the domain generalisation algorithm uses two training domains and one unseen testing domain. We improve the empirical risk minimisation method by leveraging the difference between the two training domains. By setting $\gamma > 0$, the network will be penalised for the difference between feature embeddings from the same class from the two input domains. We clarify the process with the addition of the pseudo-code in algorithm 1. We denote the two source domains with $d1$ and $d2$.

Algorithm 1: Domain generalisation with embedding alignment. Note that the same Y is used for both inputs. Meaning each input has been sampled to the same class order as the other.

Data: $data_{d1,d2}$, Network
Result: Domain generalised network
$\gamma > 0$;
for $batch_{d1,d2}$ *in* $data_{d1,d2}$ **do**
  $input_{d1}$, $input_{d2}$, $Y = batch_{d1,d2}$;
  emb_{d1}, emb_{d2} = Network.Backbone($input_{d1}$), Network.Backbone($input_{d2}$);
  Y'_{d1}, Y'_{d2} = Network.Classifier(emb_{d1}), Network.Classifier(emb_{d2});
  $L_{CE} = CE(Y'_{d1}, Y) + CE(Y'_{d2}, Y)$;
  $L_{distance}$ = Distance(emb_{d1}, emb_{d2});
  $Loss = L_{CE} + \gamma^* L_{distance}$;
  Backpropogate($Loss$);
end

This process will balance the minimisation of the invariant risk [3] and the empirical risk explained in 2.1. A higher γ promotes the features which remain stable across the domains. A lower γ promotes the features optimal for predicting the training data. As this algorithm optimises the network to any possible target domain without the target labels, it is classified as domain generalisation, following table 1.1.

Algorithm 2: Unsupervised domain adaptation

The next proposal is an algorithm which is an extension of the previous. For this extension, we assume access to the target domain data without the labels: unsupervised domain adaptation (table 1.1).

The two input domains have to be mirrored in their respective classes. The target domain labels are unknown, but if the base prediction accuracy is sufficient, the predicted pseudo labels can be utilised in the domain alignment setup. This way, we can extend the algorithm to a three-domain unsupervised domain adaptation combination.

The model trained on the first two domains is used to predict pseudo labels Y' for the target domain. With these pseudo labels, the model can be fine-tuned once more with the distance measure to train it to embed the features of the test domain as more domain invariant. The model will be penalised during training if the test domain differs too much from the two training domains. This algorithm has three stages:

1. Train the pre-trained network on the two train domains simultaneously, as in Algorithm 1.

2. Predict pseudo labels Y' for the third test domain using the trained network.

3. Fine tune the network with the third domain to ensure that the features of the third domain have similar distributions to those of the first two domains.

The following pseudo-code of stages 2 and 3 is added for clarification in algorithm 2. Here the two source domains are noted with ds, and the target domain is noted with dt. After the code, a description is given as well.

Algorithm 2: Unsupervised domain adaptation with embedding alignment. Stage 2 & 3.

Data: $data_{ds1,ds2,dt}$, Network, Distance
Result: Domain adapted network
$\gamma > 0$;
for $batch_{dt}$ *in* $data_{dt}$ **do**
  $input_{dt}$, _ = $batch_{dt}$;
  Y_{pseudo} = Network.Full($input_{dt}$);
  $data_{dt}$ = Merge($input$, Y_{pseudo});
end
for $batch_{d1,d2,d3}$ *in* $data_{d1,d2,d3}$ **do**
  $input_{ds}$, Y= Alternate($batch_{ds1}$, $batch_{ds2}$);
  $input_{dt}$, Y_{pseudo} = $batch_{dt}$;
  emb_{ds}, emb_{dt} = Network.Backbone($input_{ds}$), Network.Backbone($input_{dt}$);
  Y'_{ds}, Y'_{dt} = Network.Classifier(emb_{ds}), Network.Classifier(emb_{dt});
  $L_{CE} = CE(Y'_{ds}, Y) + CE(Y'_{dt}, Y)$;
  $L_{distance}$ = Distance(emb_{ds}, emb_{dt});
  $Loss = L_{CE} + \gamma^* L_{distance}$;
  Backpropogate($Loss$);
end

In the following sections concerning unsupervised domain adaptation, we refer to steps 1, 2 and 3 of the unsupervised domain adaptation method as stages one, two and three, respectively. The setup for unsupervised domain adaptation is no different than for domain generalisation after the pseudo labels are obtained in stage 2, and the batches are sampled. In the batch generation process, for X_1 we randomly sample a batch of either of the training domains. The batches from the target domain occupy X_2. The two resulting batches then become the input. Note that the input is a combination of one of the training domains and the target domain. The batch generation of this method is also represented in figure 3.1 noted with unsupervised domain adaptation on the left. The loss function, criterion and optimiser are also the same as before.

3.3 Distance implementations

Norm In our case, a distribution is a collection of points in a feature space. The point-wise distance between the points from each distribution would be the most straightforward way of approaching the distance between the distributions. This distance is the l^2-norm, which takes the form:

$$\|\boldsymbol{z}\|_2 := \sqrt{z_1^2 + \cdots + z_n^2}. \tag{3.2}$$

Here z is the value resulting from the difference between the element e in feature vectors of each distribution on the same index $z_i = x_{i,1} - x_{i,2}$, where $x_{i,k}$ is feature vector from domain k on index i. The l^2-norm is a simple way of calculating the distance between point clouds. It will serve as a baseline to which we compare the overall performance of the other two distances.

Hilbert-Schmidt Independence Criterion The first improvement on the baseline is the Hilbert Schmidt independence criterion (HSIC) [19]. This criterion measures the dependence between two distributions. If a causal relationship from features to the target variable is domain invariant, it is present in every other domain. However, if the relationship between the features and the target value is domain-specific, it will not exist in another domain or at least be

different. This link presents a dependency relationship between the features of each domain to those of the following domain. We quantify this relationship with the HSIC. The calculation of the HSIC results in a differentiable value. Because it is differentiable, incorporating it into the loss results in suppression or promotion of the HSIC value. A HSIC value of 0 means a fully *independent* relationship. We want to promote a more *dependent* relationship. The implementation we use is normalised [24]. Therefore, we add one minus the negative of the HSIC value to the loss function in the form of:

$$Loss = L_{ce} + (1 - L_{HSIC}). \tag{3.3}$$

We provide a full definition of the HSIC in the appendix. To summarise, the HSIC is the cross-covariance norm of two independently drawn variables which have been mapped to a reproducing kernel Hilbert space (RKHS). The variables are mapped to the RKHS using a kernel, where the common choice is the gaussian kernel. This process allows us to quantify causal relations within non-linear data as the kernels approximate any non-linear function. To make this norm useful in practice, we estimate it over batches of our data and approximate the HSIC empirically. We use a normalised HSIC or nHSIC [7] implementation [24]. The nHSIC is calculated with the trace of two matrices $\mathbf{K}_{X^1}$ and $\mathbf{K}_{X^2}$ as follows:

$$\text{nHSIC}(\mathcal{D}, \mathcal{H}, \mathcal{G}) = \text{tr}\left(\tilde{\mathbf{K}}_{X_1}, \tilde{\mathbf{K}}_{X_2}\right) \tag{3.4}$$

with $\widetilde{\mathbf{K}}_{X_1} = \overline{\mathbf{K}}_{X_1}\left(\overline{\mathbf{K}}_{X_1} + \epsilon m \mathbf{I}_m\right)^{-1}$ and $\widetilde{\mathbf{K}}_{X_2} = \overline{\mathbf{K}}_{X_2}\left(\overline{\mathbf{K}}_{X_2} + \epsilon m \mathbf{I}_m\right)^{-1} \cdot \overline{\mathbf{K}}_{X_1}$ and $\overline{\mathbf{K}}_{X_2}$ are the matrices resulting from the kernel operation and centering on the random variables X_1 and X_2, representing the features of the two domains, and ϵ is a small constant. m is the number of i.i.d samples drawn.

Wasserstein distance Approximating the optimal transport (OT), Monge or earth movers problem is an effective method for calculating the distance between two distributions. Each point in one distribution is sorted with its optimal counterpart from the other to achieve a minimum total distance. We hypothesise that this sorting can positively affect feature alignment for domain invariance because comparing data points from different distributions which are most similar keeps the focus on the difference between the distributions, not on the data points themselves. We aim to decrease the difference between a cartoon and a painting, not the difference between two breeds of dog.

The original formulation of the optimal transport problem is straightforward. Let two mountains of sand represent two distributions of equal size. How would each particle need to move to most effectively move one mountain to the shape and spot of the other mountain so that the sum of the distances is minimal? The problem was introduced by Monge [30] and has recently gained new attention for its applications in computational statistics and computer vision. In short, the problem aims to solve the minimum cost to transport each element in source distribution μ_s to target distribution μ_t. This equation takes the form of

$$L_{OT} = \min \int c(x, m(x)) d\mu_s(x). \tag{3.5}$$

Here m is the transportation $x_i^1 \rightarrow x_i^2$ and c the cost function. Approximating this optimal transport problem results in the Wasserstein distance. This work uses the optimal transport and Wasserstein names interchangeably for the distance they represent.

The implementation used for this work is $\mathcal{O}(n^2)$, however, the feature spaces of the embeddings in this work are not as high dimensional, and the implementation used [15], utilises an algorithm which is quick and efficient [44].

Chapter 4

Experiments

In this chapter, we first cover some details of the evaluation process. Afterwards, the examination of our method and its variants will be split into domain generalisation and unsupervised domain adaptation. We compare both to their respective baseline without the proposed alignment, perform ablation studies on different hyperparameters and evaluate the effectiveness of the different implementation details. The best performing model is evaluated further by t-sne plots, confusion matrices and error examples. This way, the impact of our algorithm on embeddings as a whole, and the individual classes will become apparent.

4.1 Data-set and domains

The PACS data-set [26] is selected for the performance analysis. It is often used for its size, quality, and clearly defined domains [47]. The 9,991 images are enough for ablation studies but not too many to produce resource concerns. The data-set consists of four domains Photo (1,670 images), Art Painting (2,048 images), Cartoon (2,344 images) and Sketch (3,929 images), each with 7 class categories. Some examples are shown in figure 4.1.

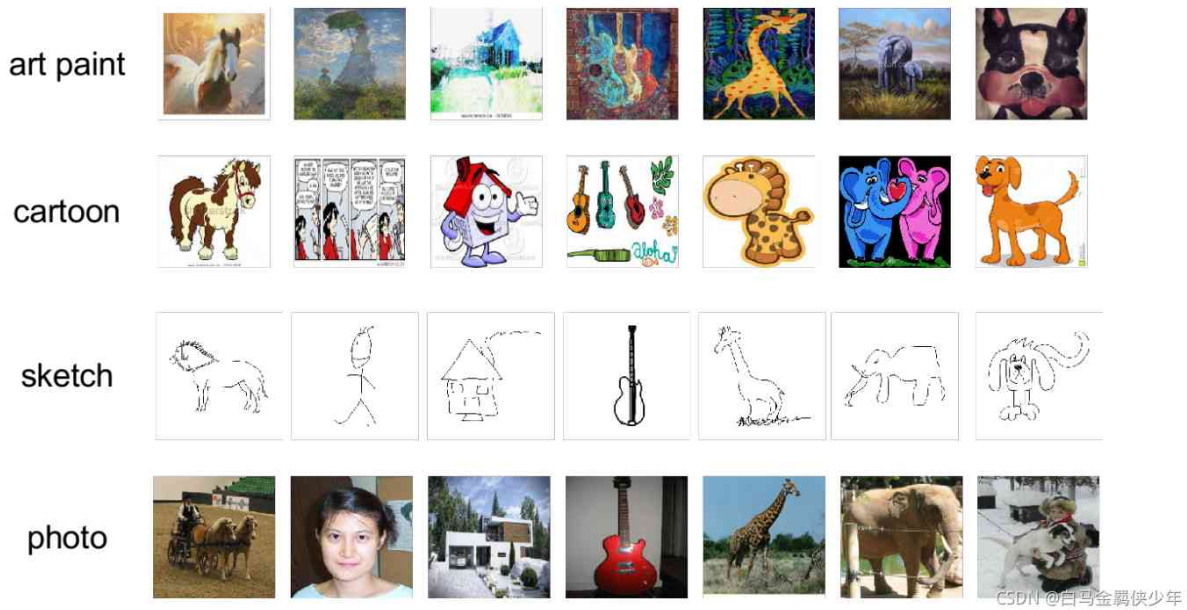


Figure 4.1: Some examples of the PACS data-set. Each column represents a class, and each row represents a domain.[50]

The substantial discrepancy between the distributions of the classes across the domains was

noted. As shown in figure 4.2 the domains Photo, Art and Cartoon have similar amounts of images for each class, but the Sketch (blue) domain has a vastly different distribution. This domain contains more than double the images of five classes and half of the images in the other two than the other domains. We mirror each image class-wise from the domains. Therefore, the amount of data per class per domain has importance in our algorithm. Also, we are exploring a three-domain setting. So, we have chosen to continue the ablation study and final comparisons with just the former three domains; art, cartoon and photo.

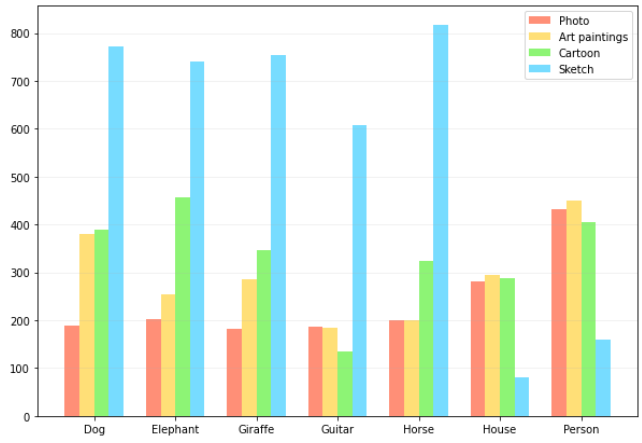


Figure 4.2: Amounts of images within the domains and classes in the PACS data-set [50]. Notice a strongly varying distribution for the cartoon domain in blue. That is why we do not use this domain for the evaluation.

The ablation studies in the domain generalisation and unsupervised domain adaptation setups are performed with the cartoon and photo domains as baseline domains while testing on the art domain. These domains were chosen in their respective positions because the art domain is seen as a middle ground between the other two domains. In this setting, we presume embedding alignment works well because the combined position of the embedded features in the feature space should most closely match the optimal position of the domain that is halfway in between. To further test the hypothesis, we include the performance on the other two orders of domains at the end.

4.2 Implementation

Various algorithm variations have been implemented and tested, to validate the main design choices. Here we explain in short what function each alteration has, its implementation and related hyperparameters.

Optimal transport ordering Ordering the samples within batches aids the distance measure as much as possible by setting similar data points next to each other on the informative differences between different domains. To this end, we add efficient sorting of the two opposing domains. A by-product of the Python Optimal Transport solver [15] is that it produces a sorted list of the most efficient matching between the points of two different distributions. This list is used to order the two respective batches.

Pseudo label selection The set of pseudo-labels contains a number of errors. An improved selection can be made from the total set of pseudo-labels. This selection will reduce the error count in the pseudo labels and the total count of the adaptation set. The error count reduction should increase the robustness and the total effect of the adaptation process, but the total count reduction can also reduce the adaptation performance. We implement and compare the pseudo label selection by selecting the whole set or just the top κ labels, which were predicted with the most confidence.

Error accumulation regulariser We can attempt to reduce the impact of the pseudo labelled data in the cross entropy. The predictions after stage two have imperfect accuracy.

Training with improper labels and using their respective prediction errors in stage 3 to back-propagate and counterbalance the distance minimisation would increase the variance of the learning. A hyperparameter δ is added to the target domains part of the L_{CE} to regulate this behaviour. The CE loss for the third stage takes the form:

$$L_{CE} = L_{ce,domain1/2} + \delta * L_{ce,targetdomain}. \tag{4.1}$$

Decreasing δ will reduce the amount of influence the prediction loss of the target domain has on the total loss.

4.3 Task 1: Domain Generalisation

The algorithm that serves as a baseline is similar to domain generalisation with regular training. We can use the proposed algorithm described above for this purpose. γ is the hyperparameter which regulates the amount of influence the distance between respective domains has in the loss function. If we set $\gamma = 0$ and train the backbone with the classification head as usual on two domains, the distance measure will not have any effect. This is the approach empirical risk minimisation dictates in section 2.1. It states that the data provided contains all the available information, and we can only empirically approximate optimal out-of-distribution performance by training on a shuffled known data-set. In this baseline approach, we shuffle data of domains one and two to approximate optimal performance on domain three.

We start with ablation of the different implementation details. For domain generalisation, these include the various γ-values for each of the two distance measures of interest and the ordering of the samples with optimal transport. For each of these combinations, a new pre-trained network is used. These setups are then compared to the baseline, which has been trained without a distance measure, i.e. $\gamma = 0$. These combinations are each subjected to 10 iterations to obtain a robust result. The performance measure for this evaluation is the accuracy of the network on the third target domain after training.

Note that for the conciseness of the text in and around images and tables from the results we may use abbreviations, these are: Optimal transport (OT), domain generalisation (DG), unsupervised domain adaptation (UDA) and HSIC.

Effect of alignment strength The ablation study using various γ values is essential to test the effect of optimising for out-of-distribution performance. Together with the optimal transport ordering procedure, the effect of changing the γ hyperparameter will be evaluated. These results can evaluate the balance between empirical and invariant risk minimisation or the balance between domain generalisation and distinctive ability.

γ	HSIC Accuracy %	OT Accuracy %
0	69.9 ± 1.7	69.9 ± 1.7
0.3	71.8 ± 1.2	70.0 ± 1.9
0.5	70.8 ± 1.2	71.4 ± 3.1
0.7	72.3 ± 0.9	**72.0** ± 3.3
1	**72.5** ± 1.4	69.3 ± 3.8
1.5	72.4 ± 1.1	66.7 ± 1.9

Table 4.1: The results of the γ ablation study both HSIC and optimal transport distances for domain generalisation. The results show clear drops in performance above and below each peak represented in bold to show optimal performance.

The results of the alterations to the γ values in the domain generalisation sets are given in table 4.1. As expected, the performance of the different distance measures peak at a specific γ value with performance degradation at higher and lower γ values. This peak indicates the balance of domain alignment and distinctive ability within the networks. The optimum of this balance is to set $\gamma = 1$ and $\gamma = 0.7$ for the HSIC and optimal transport distances, respectively.

Effect of batch ordering The impact of the implementation choice to order the data points of each class to link with the optimal counterpart in the other with the optimal transport implementation is presented in table 4.2. The optimal γ value is also noted because this value did not stay constant across variations. Surprisingly, the effect of this ordering is negative across all distance measures. Therefore, minimising the feature distance between two images of one class is *not* more effective if those images are more similar, such as the same breed of dog, which we expected. The accuracy of all variations degrades when the ordering procedure is used, indicating more complicated embedding structures should be compared.

Distance	γ	OT ordering	Accuracy %
Baseline	0	False	69.9 ± 1.9
Norm	0.03	False	70.1 ± 2.2
	0.03	True	69.7 ± 2.7
HSIC	1	False	**72.5** ± 1.4
	0.7	True	71.3 ± 1.8
OT	0.7	False	**72.0** ± 3.7
	0.3	True	70.8 ± 1.8

Table 4.2: The results of ordering by optimal transport for domain generalisation show a negative effect on each of the distance measures.

Effect of distance measure The most important findings of the domain generalisation setup are presented in table 4.3. These are the optimally functioning model variations for each distance measure and their accuracy score on the test set. For these results, the learning rate was set to 0.001, and no optimal transport ordering was used. The top performing γ value is noted in the table because it does not stay constant across the distance measures. We note a slight but clear improvement in the test set prediction accuracy with the optimal hyperparameters, suggesting the benefit of balanced domain alignment for domain generalisation.

Distance	γ	Accuracy %
Baseline	-	69.9 ± 1.9
Norm	0.03	70.1 ± 2.2
HSIC	1	**72.5** ± 1.4
OT	0.7	72.0 ± 3.7

Table 4.3: Results of the domain generalisation setup distance measure comparison. These are achieved with the optimal hyperparameters. The HSIC and optimal transport distance minimisation show clear improvement on baseline and norm distance methods.

4.4 Task 2: Unsupervised Domain Adaptation

The evaluation for the unsupervised domain adaptation approach is slightly different. A network is trained in stage one using the two training domains. The pseudo-labels of the target

domain are predicted in stage two. After the prediction, the network is adapted in stage three using the pseudo-labels and the same setup as for domain generalisation. The same base network and pseudo-labels resulting from stage two are used for each variation of hyperparameter and distance measure. To clarify: the network is trained in stage one, the pseudo-labels of the target domain are predicted in stage two, and these two are used as a base to train each hyperparameter/distance measure variation. We reuse the base network and predictions for the efficiency of the process and to keep the comparisons between the variations consistent. For unsupervised domain adaptation, the hyperparameters are γ (section 3.2) and δ (section 4.2)from the unsupervised domain adaptation section.

The main target in this setting is the gain between the accuracy on the third target domain before and after stage three. This accuracy gain can be interpreted as the performance gain of the network on the target domain with the domain adaptation strategy using pseudo labels. This complete process is run ten times for each hyperparameter setting to get a robust accuracy score.

The empirical approach mentioned above, and invariance balance in the unsupervised domain adaptation setup is evaluated with three hyperparameters. The γ value has been covered extensively. The δ and κ values, however, have a similar effect of reducing the adaptive effect of the algorithm, but in more specific areas. The δ and κ hyperparameters were introduced in section 4.2. The δ value suppresses the error accumulation by suppressing the contribution of the pseudo label prediction error in the final loss function. The κ value suppresses the number of pseudo labels used for adaptation by selecting the top most confidently predicted. This selection reduces the total amount of pseudo labels used and, therefore, the total amount of adaptation training batches. Changing each of the three hyperparameters γ, δ and κ, together with the OT ordering, will be evaluated individually in the unsupervised domain adaptation section.

The same abbreviations as before are used to declutter the images and tables: Optimal transport (OT), domain generalisation (DG), unsupervised domain adaptation (UDA) and HSIC.

Effect of selected pseudo-labels The positive performance of the networks from table 4.7 are promising for our unsupervised domain adaptation setup. Next, the different hyperparameter ablation studies that led to these performances are given to study the trade-off inherent to domain invariance.

	HSIC	OT
κ	Absolute accuracy gain %	Absolute accuracy gain %
0.3	2.2 ± 2.2	1.8 ± 2.3
0.5	**6.6** ± 1.8	**5.9** ± 1.8
0.7	3.8 ± 1.9	3.1 ± 1.1
1	1.5 ± 1.0	1.4 ± 1.3

Table 4.4: κ most confident pseudo labels and their impact on the accuracy gain of the target domain. Using half of the pseudo-labels (0.5) for the domain adaptation stage results in optimal performance.

The performance of the network with the top κ most confident pseudo labels is presented in table 4.4 on the pseudo label selection. Knowing that the predictions are not perfect, a subset of the most confident pseudo labels should result in more stable domain adaptation as well as a reduced length of stage 3. The highest performance, denoted in bold again, indicates a clear optimal κ of 0.5 for both distance measures. This optimum is concentrated as there are steep

performance drops at either higher or lower κ values. We continue the ablation studies with $\kappa = 0.5$.

Effect of alignment strength and error accumulation reduction The balance between empirical and invariant risk minimisation is also visible in 4.5. This table shows the impact of variations on γ and δ on the accuracy gain, or the effect of alignment strength and error accumulation reduction respectively. Again, we note two distinct performance summits within these matrices, denoted in bold. The optimal values for the hyperparameters are $\delta = 1$ and $\gamma = 0.3$, surprisingly, for both distance measures, indicating that the prediction loss on the pseudo-labels does not need to be reduced. We expected that reducing the influence of the pseudo-label prediction loss on the total loss would be beneficial because it is training with imperfect labels. This result could be because κ is already set to 0.5, which improves the pseudo-labels accuracy. Also notable is the results table with multiple distinct local optima in this limited amount of space.

		γ				
HSIC %		0	0.1	0.3	0.7	1
δ	0.5	4.8	4.6	5.3	6.2	5.0
	0.75	4.7	5.7	6.3	5.5	5.8
	1	5.6	4.5	**6.6**	5.3	6.2

		γ			
OT %		0.1	0.3	0.7	1
δ	0.5	3.4	3.4	5.7	5.7
	0.75	5.2	4.7	5.3	3.5
	1	4.5	**5.9**	5.7	5.3

Table 4.5: γ and δ ablation for both HSIC and Optimal Transport (OT) distances for unsupervised domain adaptation. The accuracy gains for variations in both the hyperparameters show a local optimum.

The impact of ordering the batch classes during stage 3 on the accuracy gain is given in table 4.6. As we have seen in section 4.3, the ordering has no positive impact across the distance measures in the unsupervised domain adaptation setup. Similar to the domain generalisation setup, it limits the algorithm's adaptation performance, which indicates adverse embedding effects.

Distance	γ	δ	OT ordering	Accuracy gain %
Norm	0.7	0.75	False	**4.4** $\pm$ 0.9
	0.3	1	True	1.8 $\pm$ 2.3
HSIC	0.3	1	False	**6.6** $\pm$ 1.8
	1	0.75	True	5.5 $\pm$ 2.7
OT	0.3	1	False	**5.9** $\pm$ 1.3
	1	1	True	4.4 $\pm$ 1.2

Table 4.6: Ordering of the batches with optimal transport and its negative impact on the accuracy.

Effect of three stages with optimized hyperparameters The top-performing model variations for each distance measure in the unsupervised domain adaptation setup are presented in table 4.7. In this table, the accuracy gain resulting from stage three is denoted for the optimal network configuration of each distance measure. The γ and δ values are noted because they are not constant across models. The networks were constantly found to be optimal without the optimal transport ordering and selecting only the top 0.5 most confident pseudo labels. Therefore these are not noted in these results. These results show an apparent positive effect of the unsupervised domain adaptation method on the test set performance, indicating a similar

positive effect to the domain generalisation results. Aligning the embedding distributions *can* improve OOD performance with the correct hyperparameters.

Distance	γ	δ	Accuracy gain %
Norm	0.7	0.75	4.4 ± 0.9
HSIC	0.3	1	$\mathbf{6.6} \pm 1.8$
OT	0.3	1	5.9 ± 1.3

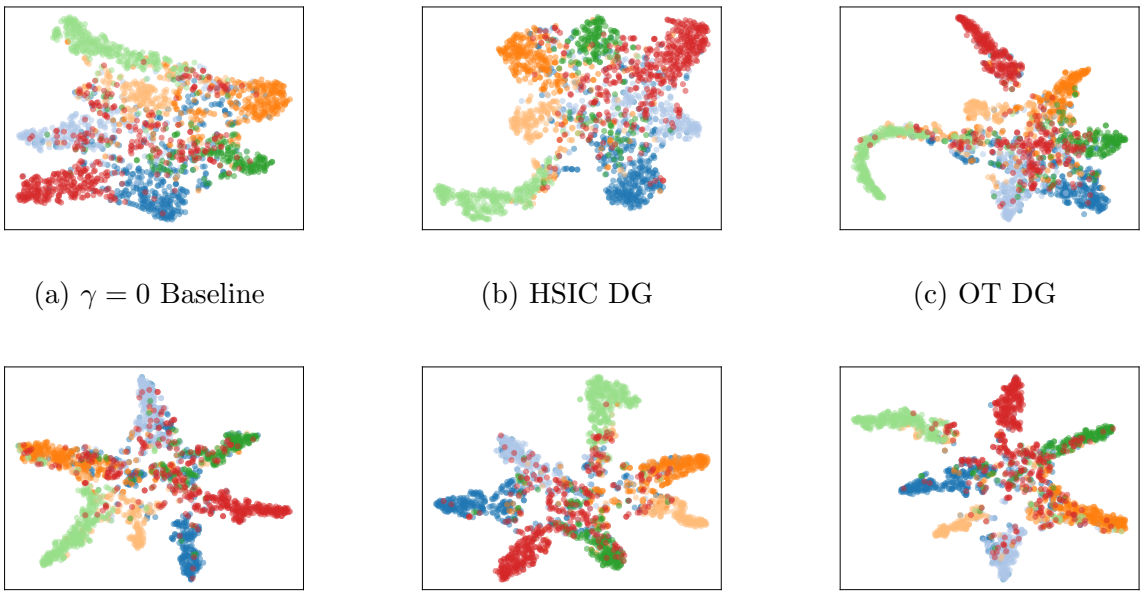
Table 4.7: Results of the unsupervised domain adaptation setup with top-performing hyperparameters show a performance gain for the target domain after stage three. The HSIC distance is the most effective.

4.5 Performance Analysis

We analyse the top-performing models and gain an understanding of the impact of the proposed algorithm by inspecting t-sne plots, confusion matrices, and error examples. The embedding distributions provide insight into the working of the algorithm. If the algorithm performs as hypothesised, these distributions per class from the target domain will be more explicitly divided when produced by the backbone of a network with $\gamma > 0$. The confusion matrices and error analysis will provide insight into the class performance and edge cases. Comparing these matrices between the variations will clarify which classes are difficult to adapt or generalise when shifting domains. Analysing errors will provide a better foundation for these last findings on the class difference. Note the abbreviations once more: Optimal transport (OT), domain generalisation (DG), unsupervised domain adaptation (UDA) and HSIC.

Understanding the performance the algorithms have achieved and the impact the domain generalisation and unsupervised domain adaptation methods have on the embeddings is a critical aspect of this thesis. To verify that the algorithms achieve their domain invariant feature distributions goal, we present the t-sne plots in figure 4.3. Here the top row represents the domain generalisation setup, and the bottom row the unsupervised domain adaptation setup. The baseline, HSIC implementation and optimal transport implementation are shown in order from left to right for both rows. Considering the domain generalisation setup in the top row of this figure, we notice slightly more defined embedding structures in the two images in the middle and on the right (4.3b, 4.3c). The individual classes are more compressed in their respective embeddings distributions when produced with the domain generalisation embedding alignment during training. However, the same is true for the full data-set, its distribution is, in general, mapped closer together. Having all the classes embedded closer together could be a disadvantage. However, considering the accuracy improvement for domain generalisation, this does not seem to outweigh the more clearly defined classes.

Analysing the unsupervised domain adaptation embedding results in the bottom row and comparing it with the top row, or (4.3a,4.3b, 4.3c) with their counterparts in (4.3d,4.3e, 4.3f), we notice four compelling elements. Firstly, the adapted embeddings offer more clearly bounded distributions. Each class has a clear, dense centre. This indicates the positive impact of training with pseudo-labels of the test batch for the embeddings. Secondly, the embeddings of the HSIC and optimal transport/unsupervised domain adaptation-aligned networks in the middle and on the right (4.3e, 4.3f) are only marginally better. This marginal improvement is expected, considering for instance the accuracy gains for HSIC in table 4.5 with $\gamma = 0$ and $\gamma = 0.3$ only differs 1%, which already indicates the marginal effect of the distance measure. Thirdly, there is a slight difference between the embeddings with either HSIC or optimal transport distance alignment for domain generalisation and unsupervised domain adaptation. Lastly, the class



(a) $\gamma = 0$ Baseline (b) HSIC DG (c) OT DG

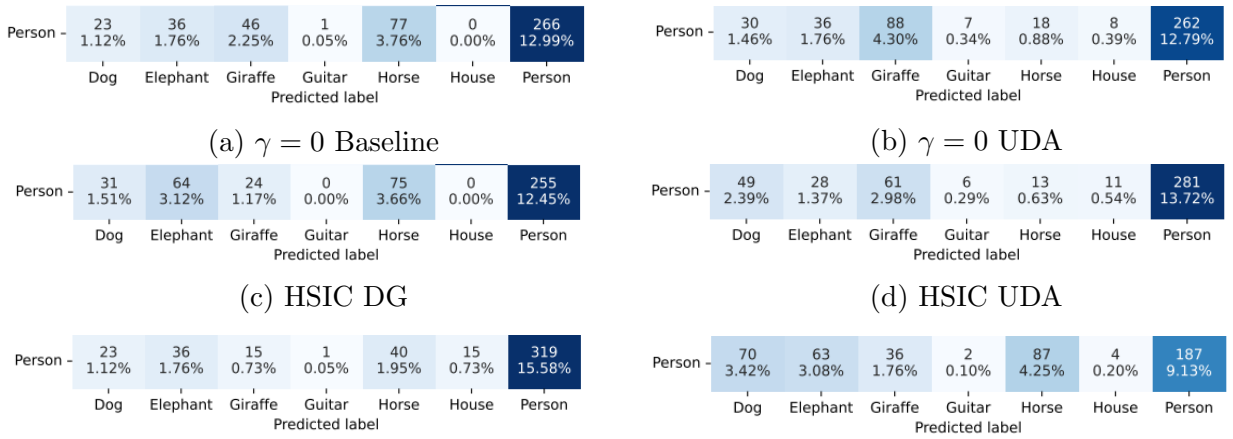
(d) $\gamma = 0$ UDA (e) HSIC UDA (f) OT UDA

Figure 4.3: t-sne visualisations of the network's ability to split the classes from the *unseen* domain. The top row represents the different domain generalisation implementations and shows improved embedding structures from the baseline (left) to the HSIC (middle) and optimal transport (right) versions. The bottom row represents the unsupervised domain adaptation implementation. The structures after the domain generalisation method are an improvement on the baseline and the structures after unsupervised domain adaptation are even better still.

that stands out as difficult to classify and segment in all sub-figures of figure 4.3 is person, represented in red. This class is often mapped to the wrong classes or in the centre.

Error analysis To investigate the problematic classes and their performance differences between models, we present the confusion matrices in figures 4.4. In these figures, the x-axis represents the predicted labels, and the y-axis represents the actual labels. The values show which target label is predicted as another label, which is noted on the y-axis. For instance, the value in the fourth square for figure 4.4a is 1, or: one image of a person is predicted as a guitar, or 0.05% of all images in the whole set. The percentages do not add up to 100%, because these are subsets of the confusion matrices of the entire data-set, provided in the appendix in figure A.1. Here we show domain generalisation on the left, unsupervised domain adaptation on the right, and the baseline HSIC and optimal transport implementations shown from top to bottom. Only the person class is presented for each of the variations. We highlight this class because of the difficulty of embeddings shown above in the t-sne plots in figure 4.3. We note that the optimal performance for the person class is with the optimal transport/domain generalisation (4.4e) procedure within domain generalisation and the HSIC/unsupervised domain adaptation (4.4d) procedure for unsupervised domain adaptation. Both have lower maximum errors and a higher correct score than their respective counterparts in their section. Surprisingly, comparing these top performers from each task with each other shows that optimal transport/domain generalisation (4.4e) is the top performer in the most challenging class, because of the highest percentage. Looking back at the t-sne plot in figure 4.3c, we can see the person class in red is also clearly separated from the rest. Optimal transport seems to improve each class seperation, but takes this too far in unsupervised domain adaptation. There it maps many points to the wrong class as seen in the t-sne figure 4.3f, which is supported by the result in confusion matrix

4.4f.



(a) $\gamma = 0$ Baseline

(b) $\gamma = 0$ UDA

(c) HSIC DG

(d) HSIC UDA

(e) OT DG

(f) OT UDA

Figure 4.4: Confusion matrices of the prediction on the target data. The x-axis represents the predicted labels for class 'person'. Note that these are subsections of complete confusion matrices, which are given in the appendix in figure A.1. This is why the percentages from left to right do not add up to 100%.

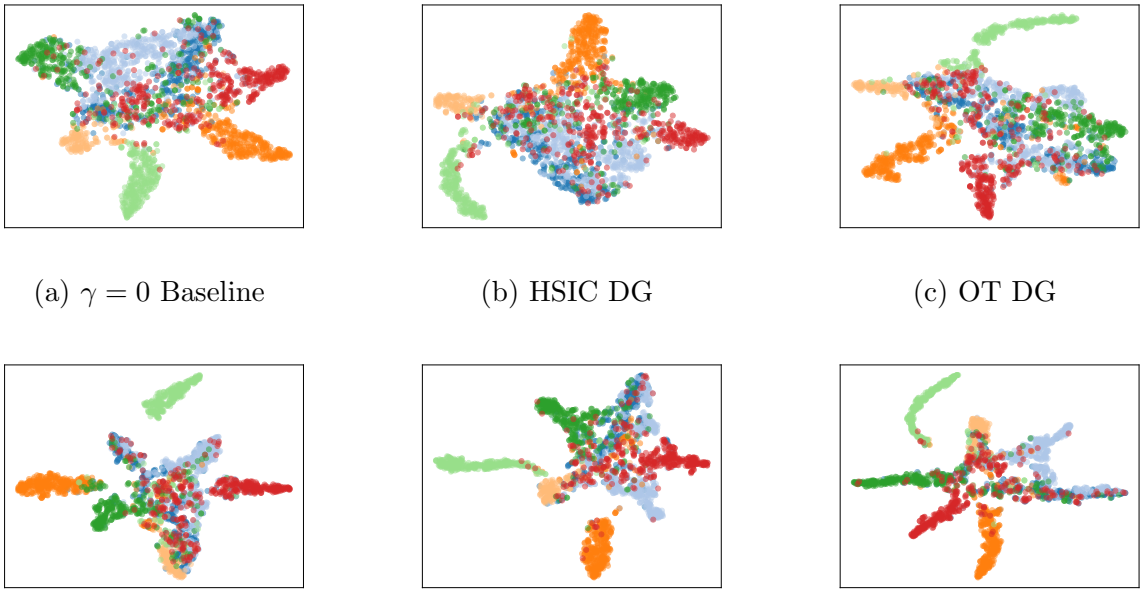
Domain variations The results so far were produced with cartoons and photos as the two training domains and art as the target domain. This ordering is important for two reasons. Firstly, it hypothesised that the order used for the in-depth analysis and ablation studies, i.e. art as the target, has more effective embedding alignment because stylistically, art is between cartoon and photo. The second reason concerns unsupervised domain adaptation. The accuracy of the network while predicting the pseudo labels will be different if training on different domains. To further investigate the robustness of our method toward different domains, we present table 4.8. This table shows the accuracy of the baseline and the two distance measures on the other two permutations of domain orders in the domain generalisation setting (left), as well as the accuracy gain of the unsupervised domain adaptation setup for the two distance measures (right). The hyperparameters were optimised for each result in these tables. Surprisingly, for the domain generalisation setup, there is minimal difference between the accuracy with or without embedding distance minimisation. The HSIC seems to have a small positive effect and the optimal transport a small negative effect for both target domains, but these effects are all within the error margin. The error margin does decrease with the better baseline accuracy of the photo target domain.

DG accuracy %	Target domain	
	Cartoon	Photo
$\gamma = 0$ Baseline	50.9 ± 3.8	85.6 ± 1.7
HSIC	**51.1** ± 3.8	**86.4** ± 2.0
OT	49.7 ± 5.4	83.9 ± 2.1

UDA stage 3	Target domain	
Accuracy gain %	Cartoon	Photo
$\gamma = 0$ Baseline	7.9 ± 5.0	3.9 ± 1.7
HSIC	8.8 ± 5.2	**4.0** ± 2
OT	**9.3** ± 3.0	3.7 ± 2.6

Table 4.8: Accuracy of the domain generalisation method (left) and accuracy gain of stage three from the unsupervised domain adaptation method (right) compared to the baseline for the two remaining permutations of domain order. Note the big difference in baseline accuracy and its negligible impact on the domain generalisation method accuracy difference and, for unsupervised domain adaptation, the positive impact of stage three on the whole.

For the unsupervised domain adaptation setup, we can use the baseline accuracy of the domain generalisation results to reflect on the effect of the pseudo-label accuracy. The accuracy gain improves with lower baseline accuracy. The effect of stage three seems to scale inversely with the baseline accuracy—the higher the baseline accuracy, the lower the effect of stage three. We expected the lower baseline accuracy to negatively impact the accuracy of the pseudo-labels, which would impair the effect of stage three of unsupervised domain adaptation. However, the inverse effect is represented on the right of table 4.8. We thought the decreased accuracy of the pseudo-labels would have too much of an impact on the stage three training effectiveness, but lower base accuracy seems to lead to higher gains. To further analyse this behaviour once more we present figures 4.5. These are similar to the t-sne plots as before, but with cartoon as target domain to test the two methods' effectiveness within a more difficult environment. From these images, it is clear that the domain generalisation method does not improve the embedding structures much. This effect is also visible in the numerical results from table 4.8. The unsupervised domain adaptation setup performs better. Judging from the embedding structures, the improvement from using a distance measure in addition to the pseudo-label training has an even greater effect than was shown in the previous domain order. We can still see the misrepresentations of the person class, for instance in unsupervised domain adaptation with optimal transport (4.5f) as in the confusion matrix, but this does not outweigh the improvement in the rest of the classification.



(a) $\gamma = 0$ Baseline (b) HSIC DG (c) OT DG

(d) $\gamma = 0$ UDA (e) HSIC UDA (f) OT UDA

Figure 4.5: t-sne visualisations of the network's ability to split the classes from the *unseen Cartoon* domain. The top row represents the different domain generalisation implementations and shows improved embedding structures from the baseline (left) to the HSIC (middle) and optimal transport (right) versions. The bottom row represents the unsupervised domain adaptation implementation. There is a similar positive effect visible in the bottom row to the previous t-sne plots. However, the domain generalisation method seems less effective in creating embedding structures.

4.6 Discussion

The previous results have all been objectively studied without much interpretation. We now revisit some of the results to distil some useful lessons denoted in *italic*.

Task 1: Domain generalisation Starting with the highlights of domain generalisation in table 4.3. The performance of the network increases slightly with both HSIC and optimal transport, but significantly. For the optimal transport distance, the increase is within the error margin, making the result less stable. The HSIC distance lessens the error margin for the difference to become significant, however small. The difference is small, but adding a certain distance between two source domains can aid in performance on the target domain. This result, however, becomes insignificant when the other domain orders are shown in table 4.8. *The domain generalisation approach does not embed the features better and apart from the small positive effect on the art domain, there is no significant accuracy improvement.*

We note some differences between HSIC and optimal transport for domain generalisation. The class structures in the embeddings from the optimal transport process seem more defined, but also more pushed together, for both of the target domains that are shown. However, the accuracy is better with the HSIC measure. This could be attributed to the fact that optimal transport sometimes matches examples of different classes to each other if the total distance is then lower. Minimising the distance between classes leads to this centred feature space. *Centring features increases domain invariance, but decreases accuracy.* That said, the difference is small.

Task 2: Unsupervised domain adaptation The performance increase of the unsupervised domain adaptation setup is undoubtedly more significant than the domain generalisation setup. Not only do the performances significantly improve with all the distance measures, but the embeddings in figure 4.3 also show an improved structure after the domain generalisation and unsupervised domain adaptation methods have been applied. Interestingly, setting $\gamma = 0$ in unsupervised domain adaptation also seems to provide some better structure in the embedding distributions and results in a significant accuracy gain across all the different test domains. *We find that simply passing the target data through the model with the pseudo-labels results in an adaptation improvement, and even more so with embedding alignment and pseudo-label selection.*

Splitting these pseudo labels only to incorporate the top half of confidence scores is the optimal setting for accuracy improvement. This performance could indicate two things. First, knowing that the actual accuracy of the pseudo-labels is $\pm 70\%$, it would suggest that there is a $\pm 20\%$ wide section of labels not accurate enough to aid the embedding alignment. Alternatively, secondly, it could indicate that the pseudo labels, in general, are not accurate enough, and simply using half is an effective way of regulating their impact. As shown in table 4.8, the results of cartoons as the test domains also support the latter of these theories. Here we know the pseudo-labels are $\pm 50\%$ accurate, but the gain from stage three is significant, and vice versa for the photo class as target domain. This option would not suggest that an improvement in pseudo-label accuracy is a direct improvement in final accuracy gain, but that there is more to gain for networks with lower initial accuracy. Following that reasoning, we arrive at: *The unsupervised domain adaptation approach with embedding alignment works and achieves better performance in sparse situations.*

Concerning the difference between the two distance measures, the findings seem reversed with respect to task 1. The HSIC method leads to more centred feature embeddings and lower accuracy gains. This reversal could be due to the accuracy of the pseudo-labels. The optimal transport method could 'repair' some of the mistakes made in the classification of the pseudo-

labels. *This indicates that optimal transport is the better choice with less accurate pseudo-labels, and thus also in sparse data conditions.* This reasoning is also supported by the accuracy's given in table 4.8. Similarly, for the confusion matrices in figure 4.4d and 4.4f, the opposite effect is visible to task 1. HSIC generalises better than optimal transport, but optimal transport adapts better at lower base accuracy's.

On the causal and information theoretical implications Interpreting the impact of the information bottlenecks created by the regularisers on these results is difficult. However, a clear balancing effect is visible in the accuracy's. *Weighing the regulariser heavier decreases the distance loss, but only increases accuracy up to a certain point.*

The reasoning from a causal perspective from section 2.1 was that only the domain invariant features will lead to more domain invariant prediction. We have improved domain adaptation and generalisation performance, but as the theory also states, too much domain invariance decreases distinctive performance. More research is needed on a feature level, to further investigate which features are responsible for the accuracy improvement.

Chapter 5

Conclusions

Conclusion In this thesis, we have empirically shown that feature alignment positively impacts domain invariant performance in an unsupervised domain adaptation setting when correctly scaled. However, for domain generalisation, this impact is only marginal with the correct starting accuracy. We have also introduced a novel three-stage algorithm for unsupervised domain adaptation using feature alignment. Moreover, we have conducted an ablation study on different distance measures for feature alignment and their respective optimal hyperparameters, revealing the balance of a network between domain invariance and distinctive ability, or empirical and invariant risk minimisation. From these studies we have also learned that for unsupervised domain adaptation a second pass of training with pseudo-labels for the target domain is often beneficial, even more so with a correct selection of pseudo-labels, embedding alignment and a lower base accuracy.

Limitations and suggestions For this thesis, assumptions and design choices have been made that limit the generalisability of the method. Also, elements in the result were found that need to be investigated further to fund the findings better. Here we review these limitations and elements for the completeness of this work and suggest follow-up studies to either mitigate or evaluate them.

Firstly, validation during the model training was performed with a validation set randomly sampled from the training set. However, the test set was used for finding the optimal gamma to get to our optimal performance with the ablation study. The assumption of a given gamma optimised for the specific setting cannot work in practice. In future work, we hope to implement a trainable gamma which can be validated on the training set.

Secondly, the accuracy scores of the top-performing models suggest a stable improvement in the baseline networks. However, a key factor was that we knew the prediction of the pseudo-labels was accurate enough for stage three to work. However, the improvement increased with different domain orders and lowered pseudo label quality. Most likely, there are diminishing returns when decreasing accuracy, however. More research is needed to find this point. We suggest further research into the impact of lower quality pseudo-labels, perhaps with better selection techniques, on final accuracy gain scores.

Thirdly, we have selected the domains of the PACS data set with the number of images per class in mind, producing similarly distributed data sets. In practice, choosing the size of the source or target distributions is impossible. However, specific improvements are possible during the data pre-processing, which could broaden the practical applications of our method. For instance, in this work, we sample the number of images per class from each domain to match the lowest amount available. Instead of this, we suggest bootstrapping or augmenting the images of the smallest class set to match the more extensive class set, providing more use in sparse data set challenges.

In this thesis, the three domain setup together with the hyperparameters which have been optimised on the target domain inhibit correct comparisons with competing approaches. And, as an extension of the pre-processing data adaptation, we suggest testing the hypothesis on different data sets, mainly DomainNet. Implementing the trainable hyperparameters and testing on the full PACS and DomainNet data sets would allow for comparisons with other state-of-the-art methods.

Furthermore, the t-sne plots in figure 4.3 show a clear improvement for both the domain generalisation and unsupervised domain adaptation setups, but this improvement seems visually stronger than would suggest from the accuracy scores. Multiple works have used SVM classifier heads after feature alignment for their ability to separate clusters in classification. Similarly, using other backbones, such as the VGG networks, was suggested, as these should produce more reliable embeddings. These seem noteworthy improvements, but the comparison with the baseline was the focus of this research, not the final score.

Lastly, we suggest exploring techniques of stacking multiple domains as source domains. The domain generalisation and unsupervised domain adaptation ability could be significantly improved, either by alternation or some interleaving construction during the distance measure calculation. This stacking would significantly increase the adaptability to different use cases of the method.

Acknowledgements

This work was made possible by the intensive supervision of Gertjan Burghouts, whom I would like to thank, as well as Cees Snoek for providing direction. William Corsel was very helpful with the training setup and Zehao Xiao with the choice of data set and his feedback.

Appendix A

Appendix

A.1 Definitions

HSIC

To define the HSIC first have to introduce Reproducing Kernel Hilbert Spaces (RKHS). Consider all the functions ϕ within the space of functions $\mathcal{F} \to \mathbb{R}$ then $\mathcal{F}$ is a Hilbert space if for every point $x \in \mathcal{X}$ there is $\phi(x) \in \mathcal{F}$ such that $\langle \phi(x), \phi(x') \rangle_{\mathcal{F}} = k(x, x')$. If $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite and associated reproducing kernel, then $\mathcal{F}$ is a RKHS. We define a second RKHS $\mathcal{Z}$ with feature mappings from $\mathcal{Z} \to \mathcal{G}$ and its corresponding kernel function $l(z, z)$ and feature map ψ. We shall need two seperate RKHS' for the defenition of the HSIC. We also need the cross covariance operator which maps from one space to another given (x, z) on $\mathcal{X}$x$\mathcal{Z}$ jointly drawn from probability distribution P_{xy}. It is defined as in equation A.1

$$C_{xy} := E_{x,y}[x, y]^T - E_x[x]E_y[y^T] \tag{A.1}$$

Introducing our kernel functions to this cross covariance operator results in the linear operator $C_{xz} : \mathcal{G} \to \mathcal{F}$ such that:

$$C_{xz} := E_{x,z}[[\phi(x)] - E_x[\phi(x)]X[\psi(z) - E_z[\psi(z)]]] \tag{A.2}$$

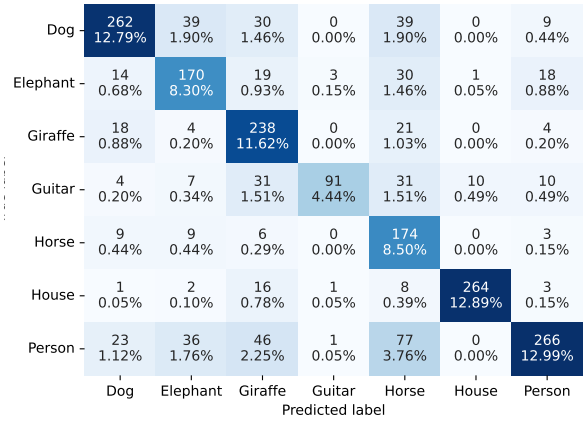
Where X is the tensor product. Filling equation A.2 with its respective kernel functions, thus skipping the function mapping step, results in the following description of HSIC:

$$\begin{aligned} \mathrm{HSIC}(p_{xy}, \mathcal{F}, \mathcal{G}) =& \mathbf{E}_{x,x',y,y'}\left[k(x, x')\, l(y, y')\right] + \mathbf{E}_{x,x'}\left[k(x, x')\right] \mathbf{E}_{y,y'}\left[l(y, y')\right] \\ & - 2\mathbf{E}_{x,y}\left[\mathbf{E}_{x'}\left[k(x, x')\right] \mathbf{E}_{y'}\left[l(y, y')\right]\right] \end{aligned} \tag{A.3}$$

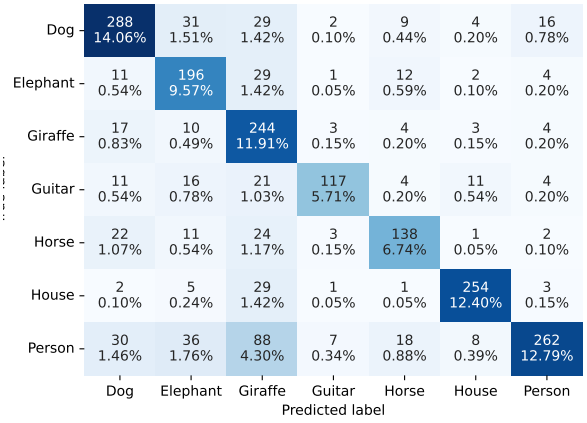
With $\mathbf{E}_{x,x',y,y'}$ being the expectation over independent pairs $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ drawn from our original joint distribution p_{xy}.

A.2 Plots

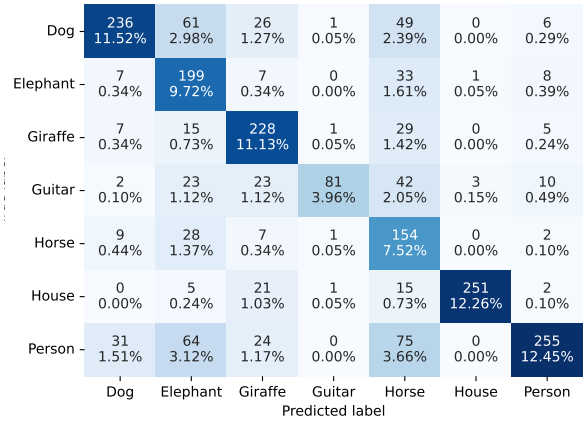
Confusion Matrices



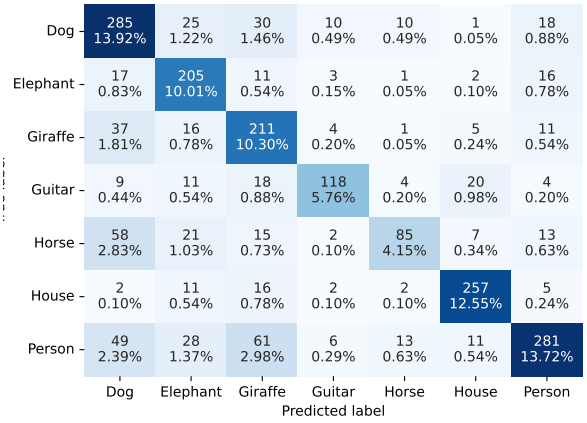
(a) $\gamma = 0$ Baseline



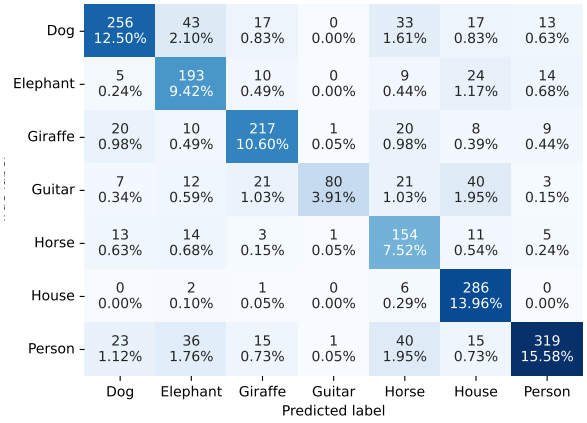
(b) $\gamma = 0$ UDA



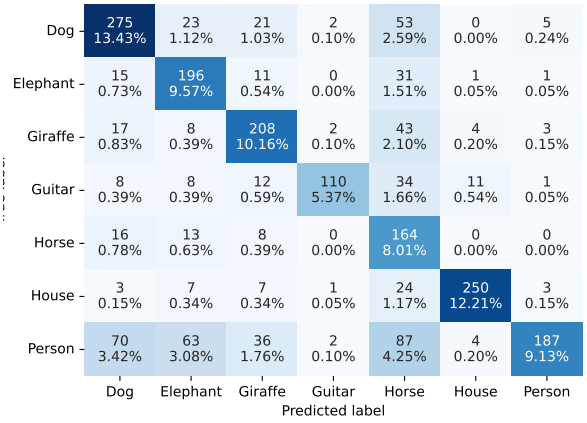
(c) HSIC DG



(d) HSIC UDA



(e) OT DG



(f) OT UDA

Figure A.1: Confusion matrices of the prediction on the target data. The x-axis represent the predicted labels, the y-axis represent the actual labels.

Bibliography

[1] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance Principle Meets Information Bottleneck for Out-of-Distribution Generalization. 6 2021.

[2] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R. Varshney. Empirical or Invariant Risk Minimization? A Sample Complexity Perspective. 10 2020.

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. 7 2019.

[4] Sara Beery, Grant Van Horn, and Pietro Perona Caltech. Recognition in Terra Incognita. Technical report.

[5] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2), 2010.

[6] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from Several Related Classification Tasks to a New Unlabeled Sample. Technical report.

[7] M B Blaschko and a Gretton. A Hilbert-Schmidt Dependence Maximization Approach to Unsupervised Structure Discovery. *Proceedings of the 6th International Workshop on Mining and Learning with Graphs (MLG 2008)*, 2008.

[8] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain Generalization by Mutual-Information Regularization with Pre-trained Models. 3 2022.

[9] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, 2019.

[10] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Re-weighted Adversarial Adaptation Network for Unsupervised Domain Adaptation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[11] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal Transport for Domain Adaptation. 7 2015.

[12] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation. In *Lecture Notes in Computer Science (including subseries Lecture*

Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 11208 LNCS, 2018.

[13] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning Robust Representations via Multi-View Information Bottleneck. 2 2020.

[14] Ian Fischer. The conditional entropy bottleneck. *Entropy*, 22(9), 9 2020.

[15] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22, 2021.

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *32nd International Conference on Machine Learning, ICML 2015*, volume 2, 2015.

[17] Daniel Greenfeld and Uri Shalit. Robust Learning with the Hilbert-Schmidt Independence Criterion. 10 2019.

[18] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test, 2012.

[19] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Sclkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3734 LNAI, 2005.

[20] Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization. 7 2020.

[21] Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 9 2018.

[22] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Feature Alignment and Restoration for Domain Generalization and Adaptation. 6 2020.

[23] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[24] Wan Duo Kurt Ma, J. P. Lewis, and W. Bastiaan Kleijn. The HSIC bottleneck: Deep learning without back-propagation. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020.

[25] Bo Li, Yezhen Wang, Shanghang Zhang, Dongsheng Li, Kurt Keutzer, Trevor Darrell, Han Zhao, Uc Berkeley, and Uc San Diego. Learning Invariant Representations and Risks for Semi-supervised Domain Adaptation. Technical report.

[26] Da Li, Yongxin Yang, Yi Zhe Song, and Timothy M. Hospedales. Deeper, Broader and Artier Domain Generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-October, 2017.

[27] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, 2016.

[28] Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M. Mooij. Domain Adaptation by Using Causal Inference to Predict Invariant Conditional Distributions. 7 2017.

[29] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain Generalization using Causal Matching. Technical report, 2021.

[30] Gaspard Monge. Mémoire sur la théorie des déblais et de remblais. In *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*. 1781.

[31] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-October, 2019.

[32] Christian S. Perone, Pedro Ballester, Rodrigo C. Barros, and Julien Cohen-Adad. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *NeuroImage*, 194, 2019.

[33] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. Technical Report 5, 2016.

[34] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19, 2018.

[35] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The Risks of Invariant Risk Minimization. 10 2020.

[36] 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, 4 2021.

[37] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6314 LNCS, 2010.

[38] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij JMOOIJ. On Causal and Anticausal Learning. Technical report.

[39] Bernhard Scholkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward Causal Representation Learning. *Proceedings of the IEEE*, 109(5):612–634, 5 2021.

[40] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, 2019.

[41] Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9915 LNCS, 2016.

[42] Naftali Tishby and Noga Zaslavsky. Deep Learning and the Information Bottleneck Principle. Technical report.

[43] Luis Caicedo Torres, Luiz Manella Pereira, and M. Hadi Amini. A Survey on Optimal Transport for Machine Learning: Theory and Applications. 6 2021.

[44] Michiel van de Panne, Wolfgang Heidrich, Sylvain Paris, and Nicolas Bonneel. Displacement Interpolation Using Lagrangian Mass Transport. *ACM Transactions on Graphics*, 30(6), 2011.

[45] V Vapnik. Principles of Risk Minimization for Learning Theory. Technical report.

[46] L. N. Vasershtein. Markov processes over denumerable products of spaces describing large systems of automata. *Problems of Information Transmission*, 5(3), 1969.

[47] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to Unseen Domains: A Survey on Domain Generalization. In *IJCAI International Joint Conference on Artificial Intelligence*, 2021.

[48] Tinghua Wang, Xiaolu Dai, and Yuze Liu. Learning with Hilbert–Schmidt independence criterion: A review and new perspectives. *Knowledge-Based Systems*, 234, 12 2021.

[49] Hanrui Wu, Yuguang Yan, Michael K. Ng, and Qingyao Wu. Domain-attention Conditional Wasserstein Distance for Multi-source Domain Adaptation. *ACM Transactions on Intelligent Systems and Technology*, 11(4), 7 2020.

[50] Jiaolong Xu, Liang Xiao, and Antonio M. Lopez. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7, 2019.

[51] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, 2017.

[52] Yuguang Yan, Wen Li, Hanrui Wu, Huaqing Min, Mingkui Tan, and Qingyao Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2018-July, 2018.

[53] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey J Gordon. On Learning Invariant Representations for Domain Adaptation. Technical report.

[54] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020.

[55] Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning: Transfer learning with deep autoencoders. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2015-January, 2015.

[56] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning, 2021.