THE UNIVERSITY
of EDINBURGH

UNIVERSITY OF EDINBURGH

•

School of Informatics

•

Doctor of Philosophy

Image Classification over Unknown and Anomalous Domains

Lucas Deecke — August 2021

Barrator, Noggershock, Falconi, Eiermann, 3000, Merte Bings.

Abstract

A longstanding goal in computer vision research is to develop methods that are simultaneously applicable to a broad range of prediction problems. In contrast to this, models often perform best when they are specialized to some task or data type. This thesis investigates the challenges of learning models that generalize well over multiple unknown or anomalous modes and domains in data, and presents new solutions for learning robustly in this setting.

Initial investigations focus on normalization for distributions that contain multiple sources (e.g. images in different styles like cartoons or photos). Experiments demonstrate the extent to which existing modules, batch normalization in particular, struggle with such heterogeneous data, and a new solution is proposed that can better handle data from multiple visual modes, using differing sample statistics for each.

While ideas to counter the overspecialization of models have been formulated in sub-disciplines of transfer learning, e.g. multi-domain and multi-task learning, these usually rely on the existence of meta information, such as task or domain labels. Relaxing this assumption gives rise to a new transfer learning setting, called latent domain learning in this thesis, in which training and inference are carried out over data from multiple visual domains, without domain-level annotations. Customized solutions are required for this, as the performance of standard models degrades: a new data augmentation technique that interpolates between latent domains in an unsupervised way is presented, alongside

a dedicated module that sparsely accounts for hidden domains in data, without requiring domain labels to do so.

In addition, the thesis studies the problem of classifying previously unseen or anomalous modes in data, a fundamental problem in one-class learning, and anomaly detection in particular. While recent ideas have been focused on developing self-supervised solutions for the one-class setting, in this thesis new methods based on transfer learning are formulated. Extensive experimental evidence demonstrates that a transfer-based perspective benefits new problems that have recently been proposed in anomaly detection literature, in particular challenging semantic detection tasks.

Lay Summary

Deep learning has enabled automated systems to make difficult decisions, for example telling which object is shown in an image. Depending on the task, computer systems now make such decisions with better accuracy than the average human. However, algorithmic models of the world often only perform well on narrowly defined problems. This thesis explores concepts and ideas, typically associated with a subcategory of machine learning research called transfer learning, that aim to venture beyond this.

One important problem in transfer learning is to understand images that depict the same objects in different styles, for example dogs that appear as photos or cartoons, known as multi-domain learning in the literature. Existing solutions typically assume that the type of style, or *domain*, an image is associated with, is known a priori. Evidence in this thesis shows that when such information is not included in the data, then standard computational techniques struggle with making the right predictions.

New ideas are proposed that improve methods in such circumstances: through new normalizations, which standardize data and allow the training of bigger, more expressive models; through new augmentations, which combine images in data to produce new ones to learn from; or through new modules, pluggable into models for classification.

This thesis also investigates the detection of unusual images in data. This could be new objects that previously were not in the dataset, uncommon poses or shapes, or new domains — e.g. sketches of cats, where before there were only photos of them.

Acknowledgements

I feel grateful and very fortunate for having shared time and ideas with many different people and collaborators in the research that lead up to the writing of this thesis.

My gratitude goes to Hakan Bilen for including me in his initial period at the University of Edinburgh, I sometimes find it hard to believe that just a few years ago the group was a fifth the size it is now! Special thanks also to Iain Murray and Timothy Hospedales for their co-supervision.

Edinburgh has been a lovely basis for putting together this thesis, I am not sure many places exist in the world where excellent research is surrounded by such nature. I would like to thank several people here: Andreas Bueff, Arthur Bražinskas, Cian Eastwood, Arushi Goel, Taha Kocyigit, Wei-Hong Li, Octave Mariotti, Simon Reinkemeier, Lazar Valkov, Robin Vogel, and Bo Zhao.

This thesis would not have come about without having been drawn into the development of early ideas around deep anomaly detection together with Marius Kloft, Stephan Mandt, Lukas Ruff, and Robert Vandermeulen. Thanks also to Marcel Langer and Matthias Rupp, who continue to intertwine physics with machine learning.

Lastly, I would like to thank Tatiana Tommasi and Amos Storkey for finding the time to evaluate my thesis, I am very honored to have such experienced and distinguished researchers as my examiners.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Lucas Deecke)

Contents

Chapter 1

Introduction

Computer vision has advanced rapidly over the last decade, achieving impressive performance in tasks like image classification (Krizhevsky et al., 2012) or instance segmentation (He et al., 2017). Learning several tasks concurrently however often presents models with considerable difficulty (Vandenhende et al., 2020), and best model performances are typically achieved when learning objectives can be narrowly defined.

An important goal in computer vision is to design models that can process data from multimodal distributions, which Vuorio et al. (2019) define as spanning multiple input and label domains. Datasets collected from such distributions contain different relevant subsets, or "modes" — for example they may include both grayscale and colored images, scenes captured during day and night (Sultani et al., 2018), or in multiple weather conditions (Pitropov et al., 2021).

Learning joint models over visually diverse subsets introduces different challenges, such as deciding between which modes to share and how to balance them (Bilen and Vedaldi, 2017). This thesis empirically investigates limitations of existing models when dealing with multimodal data, and presents new methods that improve learning in such settings.

The type of multimodality focused on here should be distinguished from the problem of learning over data from multiple modalities (Ngiam et al., 2011; Baltrušaitis et al., 2018), in which different data sources, such as audio/video, or text/images, are processed jointly. Instead, this thesis investigates problems in which models are learned for multiple modes that reside in one joint space (i.e. images with equivalent height and width), but can potentially contain very different visual characteristics, for example objects appearing as photos and sketches (Li et al., 2017).

Chapter 2 studies normalization (Huang et al., 2020) in the context of multimodal object classification. Different normalization strategies are evaluated on data from multiple source domains (see Section 2.4.1) and real-world benchmarks (Section 2.4.2) such as ImageNet (Deng et al., 2009), which is considered multimodal as it contains a large number of diverse classes that form implicit subsets/modes (Abdollahzadeh et al., 2021).

As highlighted in the background review (see Section 2.2), arguably the most prominent method for normalization is batch normalization (BN, Ioffe, 2017). Experiments in Section 2.4 however show that this technique is not particularly well suited for normalizing multimodal data. To address this limitation, a new strategy called mode normalization is proposed, which combines normalization with mixtures of experts (Jacobs et al., 1991; Jordan and Jacobs, 1994) to account for multimodality in intermediate feature distributions, and improves performance significantly over BN when jointly learning from multiple modes.

Chapter 3 investigates object classification over multimodal datasets that contain different visual domains, for example everyday objects like chairs that appear in different styles, e.g. cartoons and sketches (Li et al., 2017). A central research question is how to best learn multi-domain classification models without access to domain labels, which indicate to which visual domain an image belongs. Previous works (introduced in detail in Section 3.1) required such labels to be present in the data (Rebuffi et al., 2017, 2018; Bulat et al., 2019; Mancini et al., 2020b).

New methods are proposed in Section 3.3 for learning over multiple unlabeled domains,

or *latent domain learning* for short. The experiments in Section 3.4 indicate that the proposed methods positively influence performance in particular for domains/modes with fewer examples to learn from. Section 3.4.2 highlights that these improvements extend to other imbalanced settings, e.g. empirical fairness problems (Wang et al., 2020), in which protected attributes, such as gender, are overrepresented for portions of the data. And as experimental results in Section 3.4.3 show, they also benefit learning in settings where the label distribution is heavily skewed (Liu et al., 2019b; Cao et al., 2019).

Multimodal data may also feature in other learning tasks besides object classification, for example in anomaly detection (Ruff et al., 2021). While this has traditionally been understood as learning from a single class only (a review of existing methods can be found in Section 4.1), recent work of Ahmed and Courville (2020) explored new problem cases in which the training data is constituted by a multimodal distribution that contains multiple objects, without labels for each class.

Chapter 4 investigates new transfer-based methods for such multimodal anomaly detection. Section 4.4.1 proposes controlled intervention experiments that construct anomalies from data originally released for disentanglement research (Gondal et al., 2019) to identify the shortcomings (and benefits) of different strategies. Next, Section 4.4.2 shows that transfer-based approaches can better differentiate novel anomalies from multimodal collections of non-anomalous objects available for training.

To conclude the thesis, Chapter 5 summarizes the broader impact of the proposed methods, discusses their limitations, and proposes future research directions for learning over multimodal data.

1.1 Contributions

The main contributions in this thesis are the development of new methods for normalization over multimodal data, the formulation of a new transfer learning setting called

latent domain learning, as well as new transfer-based strategies for anomaly detection over multimodal data. The following individual methodological components constitute this thesis:

- Mode normalization (MN, Section 2.3): a normalization strategy that extends normalization to more than a single mean and variance. It is demonstrated that MN outperforms BN and other widely used normalization techniques in several benchmarks, including single and multi-domain datasets.
- Latent domain exchange (LDE, Section 3.3.1): a new augmentation strategy that exchanges style (e.g. coloring/brush stroke) between images to interpolate between domain-specific characteristics. Experiments show that this technique performs better than existing augmentation strategies when learning over multimodal data from latent domains.
- Sparse latent adapters (SLA, Section 3.3.3): a novel module that assigns examples to combinations of linear corrections at every layer of deep networks. Besides improving performance in latent domain learning, SLA is shown to enhance robustness for small domains, thereby improving existing methods in fairness and long-tailed recognition benchmarks.
- Anomaly detection with an inductive bias (ADIB, Section 4.3.2) and with residual adaptation (ADRA, Section 4.3.3). These new transfer-based anomaly detection methods are examined through the utilization of datasets originally released for the development of disentangled representations (Gondal et al., 2019), and are shown to outperform other existing strategies for anomaly detection on several established benchmarks.

1.2 Thesis Structure

The thesis is divided into five chapters. Chapter 1 contains the introduction, followed by Chapter 2, which proposes a new normalization over multimodal distributions. Chapter 3 focuses on latent domain learning. Chapter 4 focuses on anomaly detection and presents new transfer-based methods for this setting. Chapter 5 contains ideas for future research and discusses the broader impact of the methods proposed in this work.

The ideas in Chapters 2-4 are based on peer-reviewed papers. The publications associated with the individual chapters are as follows:

- Chapter 2 is based on "Mode Normalization", L. Deecke, I. Murray, and H. Bilen, *International Conference on Learning Representations* (2019).
- Chapter 3 is based on "Visual Representation Learning over Latent Domains", L. Deecke, T. Hospedales, and H. Bilen, *International Conference on Learning Representations* (2022).
- Chapter 4 is based on "Transfer-Based Semantic Anomaly Detection", L. Deecke, L. Ruff, R. A. Vandermeulen, and H. Bilen, *International Conference on Machine Learning* (2021).

Chapter 2

Mode Normalization

Normalization methods are a central building block in the deep learning toolbox. They accelerate and stabilize training, while decreasing the dependence on manually tuned learning rate schedules. When learning from distributions that contain heterogeneous data however, Bilen and Vedaldi (2017) showed that the effectiveness of batch normalization (Ioffe and Szegedy, 2015), arguably the most prominent normalization, is reduced.

While performance in this setting – associated with multimodal (Wang et al., 2017; Kalayeh and Shah, 2019; Vuorio et al., 2019; Abdollahzadeh et al., 2021) and multi-domain learning (Rebuffi et al., 2017, 2018) in existing literature – can be improved by extending models with multiple normalization units, for many applications this is prohibitive, in particular where efficiency is required, e.g. when confined to mobile platforms (Howard et al., 2017; Sandler et al., 2018), in federated learning (Yang et al., 2019; Bhagoji et al., 2019), or when processing data from sensors with limited power supplies (Chen et al., 2017).

This chapter introduces new strategies to address such limitations. Section 2.1 provides a high-level introduction into why normalization is useful, followed by a background

review of established normalization techniques in Section 2.2. Mode normalization, which extends the normalization to more than a single mean and variance, forms the central contribution of this chapter in Section 2.3. This is followed by experiments in Section 2.4 and a discussion of limitations in Section 2.5.

2.1 Introduction

The experimental evaluations in this thesis focus exclusively on image data. When not indicated otherwise, this means that it is assumed that examples are sampled i.i.d. from a data-generating distribution $\mathbb{P}$ that can be decomposed into channels, height, and width, i.e. the underlying space decomposes as $\mathcal{X} = C \times \mathcal{H} \times \mathcal{W}$.

Models are denoted as f_θ (or g_ϕ, h_η, etc.) throughout this thesis, with the learnable parameters θ residing in some parameter space, e.g. $\theta \in \Theta$. The thesis focuses on developing new methods for deep learning architectures, a large and flexible hypothesis class (Hornik et al., 1989). Optimal model parameters $\theta^\star$ are obtained through empirical risk minimization (Vapnik, 1992):

$$\theta^\star = \arg\min_{\theta \in \Theta} \Big\{ R_N[f_\theta] = \frac{1}{N} \sum_{n=1}^{N} L(y_n, f_\theta(x_n)) \Big\}, \tag{2.1}$$

where N denotes the number of examples sampled i.i.d. from $\mathbb{P}$, L denotes the task-specific loss function (e.g. cross-entropy for multi-class classification, ℓ_2 loss for regression), and y_n the ground-truth label associated with an example $x_n \in \mathcal{X}$.

One established strategy for optimization of R_N and approximating $\theta^\star$ is to minimize the risk via gradient-based techniques (Bottou, 2010). These follow negative gradient directions in parameter space, corresponding to local reductions of the risk in eq. (2.1):

$$\theta \leftarrow \theta - \gamma \nabla_\theta \Big[\frac{1}{N} \sum_{n=1}^{N} L(y_n, f_\theta(x_n)) \Big], \tag{2.2}$$

where the learning rate $\gamma \in \mathbb{R}_+$ scales the update of θ. While a precise understanding of how generalization is achieved in deep learning is still lacking (Zhang et al., 2017a), gradient-based optimization is understood to play an important role in this (Kleinberg et al., 2018; Liu et al., 2020b). The most popular techniques are stochastic gradient descent (SGD) and related methods, such as Adagrad (Duchi et al., 2011) or Adam (Kingma and Ba, 2014). These sample small mini-batches of data $\{x_n\}_{n=1,\dots,N}$ from $\mathbb{P}$ to estimate R_N, and such mini-batch strategies are also used throughout this thesis to optimize models.

Deep learning networks contain feed-forward layers that tend to get repeated throughout the model, for example fully connected layers as in multi-layer perceptrons (Rosenblatt, 1961), hidden states in recurrent networks (Sherstinsky, 2020), or self-attention in transformer architectures (Vaswani et al., 2017). This thesis focuses on the processing of image data, and hence builds on convolutional networks (LeCun et al., 1995; Dumoulin and Visin, 2016), a firmly established model class for learning over images.

Besides functional layers, deep networks typically also include activation functions, such as the sigmoid or the popular rectified linear unit $\mathrm{ReLU}(x) = \max(0, x)$ (Fukushima and Miyake, 1982; Nair and Hinton, 2010).

Deep architectures have become the de-facto standard for learning over high-dimensional distributions, and image data in particular. A defining property in deep learning is the learning of representations without excessive amounts of manual finetuning, required for example in support vector networks (Cortes and Vapnik, 1995). This is enabled via efficient end-to-end optimization of all layers via gradient-based backpropagation of errors (Rumelhart et al., 1986), c.f. eq. (2.2).

That being said, the large number of sequential transformations that are jointly optimized in deep networks results in a continuous change of the input distribution at every layer of the model, giving rise to a fundamental challenge in the optimization of deep learning models that complicates their training. Normalization methods are aimed at overcoming this issue — often referred to as internal covariate shift (Shimodaira, 2000).

The next section describes the most established forms of normalization in detail, while also highlighting some important existing limitations.

2.2 Background

Practical optimization problems often contain small inconsistencies between estimators associated with different samples from the same dataset, for example between training and validation sets, or even two mini-batches. It is especially important to account for such subtle changes since errors quickly accumulate in deep networks, complicating model training (Hochreiter, 1991).

Normalizations are an important building block in deep neural networks that makes their optimization more stable. They increase the smoothness of the optimization landscape (Santurkar et al., 2018), enabling the training of very deep networks, shortening training times by supporting larger learning rates, and reducing sensitivity to parameter initializations. While normalizations continue to be explored from a theoretical perspective (Kohler et al., 2018; Lubana et al., 2021), they have become an integral element in many state-of-the-art machine learning techniques (He et al., 2016; Silver et al., 2017).

2.2.1 Batch Normalization

Normalizing input data (LeCun et al., 1998b) or initial weights of neural networks (Glorot and Bengio, 2010) have been known for some time to support faster model convergence. With the advent of deep learning, normalization has been evolved into functional layers to adjust the internal activations of neural networks. An early example of this is the use of local response normalization (Lyu and Simoncelli, 2008; Jarrett et al., 2009), used in various models (Krizhevsky et al., 2012; Sermanet et al., 2014) to perform normalization in a local neighborhood, and thereby enforcing competition between adjacent pixels in a feature map.

Arguably the most prominent normalization technique is called batch normalization (BN, Ioffe and Szegedy, 2015). BN uses the statistics of individual batches, as they appear in SGD and related first-order optimization strategies (see Section 2.1), to regulate data statistics at intermediate layers of deep networks. In a slight abuse of notation, the symbol x is also used to represent the features computed by layers within the network, producing a three-dimensional tensor that resides in $\mathcal{X} = C \times \mathcal{H} \times \mathcal{W}$ where the dimensions indicate the number of feature channels, height and width. In BN, every example in the batch $\{x_n\}_{n=1,\dots,N}$ is first projected onto its channels via so-called average pooling using a projection $\phi\colon \mathcal{X} \to C$.[1]

Next, the projected statistics are averaged into $\mu = N^{-1} \sum_n \phi(x_n)$, and standard deviations estimated into $\sigma = N^{-1} \sum_n (\phi(x_n) - \mu)^2$. These are applied to transform every input in the following way:

$$\mathrm{BN}(x_n) = \alpha \frac{x_n - \mu}{\sigma + \varepsilon} + \beta, \tag{2.3}$$

where $\alpha, \beta \in \mathbb{R}^{\dim(C)}$ parametrize a learnable linear transformation that benefits performance (Ioffe and Szegedy, 2015), and $\varepsilon \ll 1$ is a small parameter that is meant to ensure numerical stability.

Ioffe and Szegedy (2015) showed that by normalizing the hidden representations (or features) at every layer of the network, even very deep models may still be trained robustly. One important property of BN is that each individual transformation is light-weight, enabling their insertion at every layer of modern network architectures: for example in the widely-used residual networks (He et al., 2016), they are placed after each convolution. Mode normalization, introduced in Section 2.3.2 of this thesis, also adheres to this light-weight design principle.

A peculiarity of BN is that performance benefits from storing a moving average of the estimators computed from each batch, e.g. $\mu_t = \lambda\mu + (1-\lambda)\mu_{t-1}$ with some $\lambda \in (0, 1]$. At test time μ and σ are not computed from batches of examples, but the running estimates are applied to normalize during this stage. This requirement makes it non-obvious

[1] For a projection it holds that $p^q = p$ for all $q \in \mathbb{N}_{\setminus 0}$.

how to apply BN to recurrent networks, and is one central motivation for alternative formulations such as layer normalization (Ba et al., 2016) discussed in Section 2.2.2.

A notable alternative strategy to BN is batch renormalization (Ioffe, 2017) which rescales estimators via additional affine transformations, up to a predefined range to prevent degenerate cases. While renormalization is somewhat effective for training sequential and generative models respectively, it has not been able to reach the same level of performance as BN, and was therefore not adopted as widely.

Despite its great success, BN has drawbacks due to its strong reliance on the mini-batch statistics. While the stochastic uncertainty of the batch statistics acts as a regularizer that can boost the robustness and generalization of the network, it has significant disadvantages when batch sizes become small, as the estimates for the mean and variance become less accurate. This vulnerability to small batch sizes has been reported to have a detrimental effect on models that incorporate BN (Ioffe, 2017; Wu and He, 2018). A strategy for group-wise normalization to overcome this limitation is reviewed in Section 2.2.4 of this background.

Sometimes BN is not an ideal choice for the task at hand. This is for example the case in style exchange, where a purpose-built solution called instance normalization is a more popular choice. This is discussed in Section 2.2.3.

2.2.2 Layer Normalization

To be able to contrast different normalization strategies against one another, it is helpful to generalize the formulation of BN in eq. (2.3). In particular the projection ϕ plays a crucial role in differentiating the various normalizations proposed in recent literature.

As outlined in Section 2.1, images in the mini-batch $\{x_n\}_{n=1,\dots,N}$ sampled i.i.d. from $\mathbb{P}$ can be decomposed into channels, height, and width: an individual example is therefore identified via a four-dimensional index via x_{nchw}. How this tensorized information

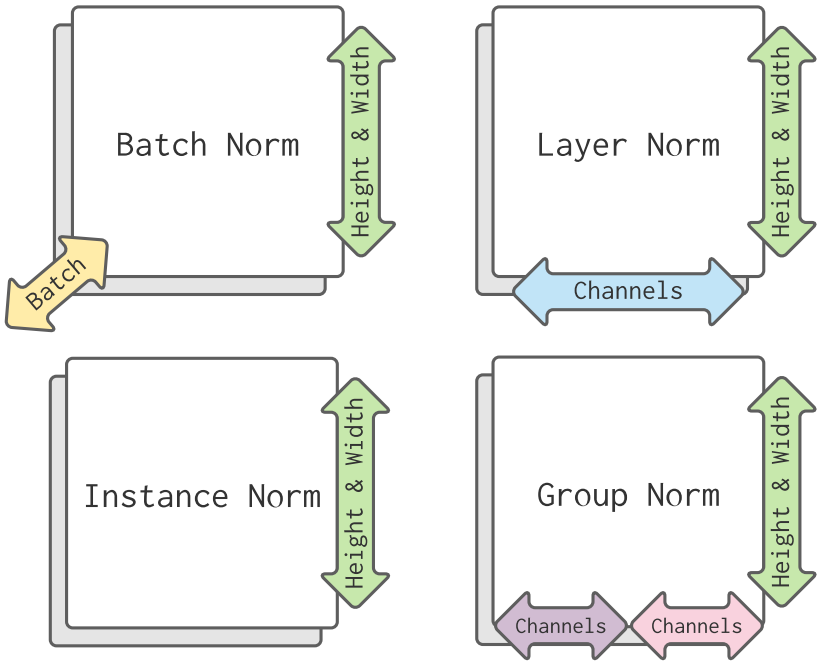


Figure 2.1: A comparison between popular strategies for normalization. The arrows indicate which dimensions are contracted to compute estimators μ and σ. Batch norm projects height and width of each image jointly for all examples in the mini-batch. Layer norm is instance-specific, and only contracts height, width, and channels of each image, whereas in instance norm estimators are computed from height and width alone. Group normalization partitions the channels into different subgroups, and averages over height and width of each group.

is reduced in projections is how the different existing normalization strategies can be differentiated against one another. A visual overview over the most prominent normalization strategies, and their corresponding projection, is given in Figure 2.1.

In layer normalization (LN, Ba et al., 2016) the projection becomes $\phi_{\mathrm{LN}}(x_{nchw}) = x_n$. Just as in BN, after computing estimators μ, σ from the projected batch, the normalization of individual examples comes next, c.f. eq. (2.3).

LN was devised with recurrent neural networks in mind, which vary with the length of the input sequence. This makes the application of BN, which uses moving averages μ_t

and σ_t during inference, non-obvious. Different from BN, in LN the same normalization is applied during training and testing stages such that no moving averages μ_t and σ_t are required, and Ba et al. (2016) show that this normalization strategy performs very well in tasks where recurrent networks are the preferred model class.

2.2.3 Instance Normalization

BN averages statistics over the entire mini-batch, which can be problematic in applications like style transfer (Gatys et al., 2015, 2016) where some amount of diversity is highly desired, and empirical results confirm that qualitative performance degrades when BN is coupled with larger batches of content images (Ulyanov et al., 2016a).

Ulyanov et al. (2016b, 2017) observe that style transfer models should be independent of the contrast in content images, and therefore benefit from having some form of contrast normalization built into the network architecture. For this they proposed instance normalization (IN), which enables contrast normalization by modifying the projection to $\phi_{\text{IN}}(x_{nchw}) = x_{nc}$ before computing estimators. Just as in layer normalization, the same procedure to estimate statistics μ, σ is used during training and testing, such that no moving averages are required.

2.2.4 Group Normalization

Performance losses occur in BN when estimating from small batch sizes. Wu and He (2018) proposed a simple yet effective alternative called group normalization (GN), which first divides the channels into groups and then performs normalization within each of them. This avoids averaging over the entire mini-batch, and the authors show that it can be coupled with small batch sizes without registering any significant performance losses, while delivering comparable results to BN when the batch size is large.

To motivate GN, Wu and He (2018) argue that many classical methods like scale-

invariant feature transform (Lowe, 1999) compute group-wise features, and apply individual normalizations to each group. Following this, GN first associates all channels $c = 1, \ldots, |C|$ with fixed groups G_j, and then jointly computes estimators over each of these groups, e.g. for the mean $\mu_j = |G_j|^{-1} \sum_{x_c \in G_j} x_c$.

The associated projection in GN is $\phi_{\text{GN}}(x_{nghw}) = x_{ng}$, and Wu and He (2018) show that this simple strategy, which is independent of the batch size, outperforms BN in tasks such as COCO object classification and segmentation (Lin et al., 2014b), where larger batch sizes, which would be required in BN, are unfeasible. Note GN does not require moving averages μ_t that are needed in BN. GN is extended in Section 2.3.3 of this thesis to automatically infer filter groupings in mode group normalization.

Another alternative to existing normalization strategies like BN or GN is to directly normalize the weights of the neural network in a strategy called weight normalization (Desjardins et al., 2015; Arpit et al., 2016). While such methods show promising results, they can only be paired with certain non-linearities and functional layers, so are less flexible than e.g. BN.

Besides the normalization strategies introduced here, another line of work investigates instance-specific conditioning of the normalization, i.e. the normalization parameters α, β in eq. (2.3) are functions of some external information like image captions. While not the focus of this thesis, such conditional normalizations were shown to be highly beneficial for specialized tasks, e.g. visual question answering (Perez et al., 2018), or semantic image synthesis (Park et al., 2019).

2.3 Methods

Bilen and Vedaldi (2017) showed that when training a deep neural network on images that come from a diverse set of visual domains, each with significantly different statistics, then BN is not effective at normalizing the activations with a single mean and variance.

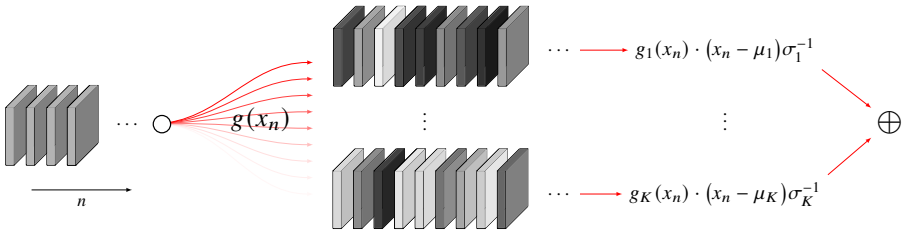


Figure 2.2: In mode normalization, incoming samples $\{x_n\}_{n=1,\ldots,N}$ are weighted by a gating function $g\colon \mathcal{X} \to [0,1]^K$. After gating, samples contribute to component-wise estimators μ_k and σ_k, under which the data is normalized, and a weighted summation of the batch is passed to the next layer (c.f. Alg. 1). Note that during inference, estimators are computed from running averages instead (Alg. 2).

Throughout this section the assumption that the entire mini-batch should be normalized with the same mean and variance is relaxed. This gives rise to a novel normalization method, called mode normalization (MN), that first assigns samples in a mini-batch to different modes via a gating network, and then normalizes each sample with estimators for its corresponding mode. This concept is displayed in Figure 2.2.

Section 2.3.3 shows that MN can be incorporated into other normalization techniques such as GN (Wu and He, 2018) (introduced in Section 2.2.4) by automatically learning which filters should be grouped together.

The proposed methods can easily be implemented as layers in standard deep learning libraries, and their parameters are learned jointly with the other parameters of the network in an end-to-end manner. MN and MGN are evaluated in multiple classification tasks in Section 2.4, demonstrating that they achieve a consistent improvement over BN and GN.

2.3.1 Mixtures of Experts

The heterogeneous nature of complex datasets motivates the proposition of more flexible treatments of normalization. Before carrying out the actual normalization, the data first has to be organized into modes or domains to which it likely belongs.

To achieve this, one can reformulate the normalization in the framework of mixtures of experts (MoE) (Jacobs et al., 1991; Jordan and Jacobs, 1994; Xu et al., 1994). This encompasses a family of models that involve combining a collection of simple learners to split up the learning problem. Samples are thereby allocated to differing subregions of the model that are best suited to deal with a given example.

There is a large body of literature describing how to incorporate MoE with different types of expert architectures such as SVMs (Collobert et al., 2002), Gaussian processes (Tresp, 2001), or, more recently, deep neural networks (Shazeer et al., 2017).

An important element in MoE are the gates, small learnable functions that assign weights to the outputs of different experts. For example Eigen et al. (2013) proposed using a different gating network at each layer in a multi-layer network to enable an exponential number of combinations of expert opinions. While MN and MGN also use a gating function at every layer to assign the samples in a mini-batch to separate modes, it differs from existing MoE approaches in two key aspects: (i.) assignments are used from the gating functions to normalize the data within a corresponding mode, (ii.) the normalized data is forwarded to a common module (i.e. a convolutional layer) rather than to multiple separate experts.

2.3.2 Mode Normalization

Mode normalization (MN, Algorithm 1) introduces a gating function $g : \mathcal{X} \rightarrow [0, 1]^K$ whose output satisfies $\sum_k [g(x)]_k = 1$ for all x. Each sample in the mini-batch is then

Algorithm 1 Mode normalization, training phase.

Input: hyperparameters λ, K, batch of feature vectors $\{x_n\}$, small ε, learnable channel-wise parameters α, β and $\Psi\colon \mathcal{X} \to \mathbb{R}^K$.

Compute expert assignments:

$g_{nk} \leftarrow \big[\sigma \circ \Psi(x_n)\big]_k$

for $k = 1$ to K **do**

Determine new component-wise statistics:

$N_k \leftarrow \sum_n g_{nk}$
$\langle x\rangle_k \leftarrow \frac{1}{N_k} \sum_n g_{nk} x_n$
$\langle x^2\rangle_k \leftarrow \frac{1}{N_k} \sum_n g_{nk} x_n^2$

Update running means:

$\overline{\langle x\rangle}_k \leftarrow \lambda \langle x\rangle_k + (1-\lambda)\overline{\langle x\rangle}_k$
$\overline{\langle x^2\rangle}_k \leftarrow \lambda \langle x^2\rangle_k + (1-\lambda)\overline{\langle x^2\rangle}_k$

end for
for $n = 1$ to N **do**

Normalize samples with component-wise estimators:

$\mu_k \leftarrow \langle x\rangle_k$
$\sigma_k^2 \leftarrow \langle x^2\rangle_k - \langle x\rangle_k^2$
$y_{nk} \leftarrow g_{nk} \frac{x_n - \mu_k}{\sqrt{\sigma_k^2 + \varepsilon}}$

end for
Return: $\{\alpha \sum_k y_{nk} + \beta\}_{n=1,\ldots,N}$

normalized under voting from its gate assignment:

$$\mathrm{MN}(x_n) \triangleq \alpha\Big(\sum_{k=1}^{K} [g(x_n)]_k \frac{x_n - \mu_k}{\sigma_k}\Big) + \beta, \tag{2.4}$$

where α and β are a learned affine transformation, just as in standard BN, see eq. (2.3). While experiments with learning individual $\{(\alpha_k, \beta_k)\}_{k=1,\ldots,K}$ for each mode have been carried out as well, no additional gains in performance were observed from this.

The estimators for mean μ_k and variance σ_k are computed under weighing from the

Algorithm 2 Mode normalization, test phase.

Input: refer to Algorithm 1.

Compute expert assignments:

$g_{nk} \leftarrow \left[\sigma \circ \Psi(x_n)\right]_k$

for $n = 1$ to N **do**

Normalize samples with running average of component-wise estimators:

$\mu_k \leftarrow \overline{\langle x \rangle}_k$
$\sigma_k^2 \leftarrow \overline{\langle x^2 \rangle}_k - \overline{\langle x \rangle}_k^2$
$y_{nk} \leftarrow g_{nk} \frac{x_n - \mu_k}{\sqrt{\sigma_k^2 + \varepsilon}}$

end for
Return: $\{\alpha \sum_k y_{nk} + \beta\}_{n=1,\ldots,N}$

gating network, e.g. the k'th mean is estimated from the batch as

$$\mu_k = \langle x \rangle_k = \frac{1}{N_k} \sum_n [g(x_n)]_k \cdot x_n, \tag{2.5}$$

where $N_k = \sum_n [g(x_n)]_k$. In the experiments, the gating networks are parametrized via an affine transformation $\Psi \colon \mathcal{X} \to \mathbb{R}^K$ which is jointly learned alongside the other parameters of the network. This transformation is followed by a softmax activation $\sigma \colon \mathbb{R}^K \to [0,1]^K$, similar to the gates in gated recurrent neural networks (Graves et al., 2013; Gregor et al., 2015; Vinyals et al., 2015).

As in BN, during training samples are normalized with estimators computed from the current batch. To normalize the data during inference (see Algorithm 2), component-wise running estimates are kept track of, borrowing from online EM approaches (Cappé and Moulines, 2009; Liang and Klein, 2009). Running estimates are updated in each iteration with a memory parameter $\lambda \in (0, 1]$, e.g. for the mean:

$$\overline{\langle x \rangle}_k = \lambda \langle x \rangle_k + (1 - \lambda) \overline{\langle x \rangle}_k. \tag{2.6}$$

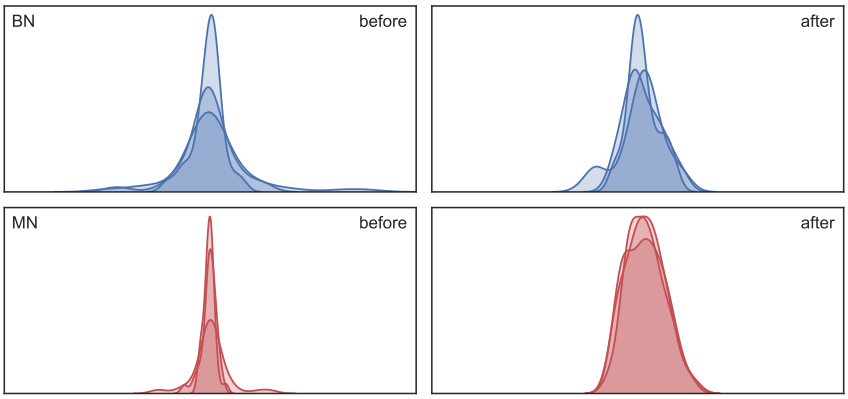


Figure 2.3: Histograms for three channels in `conv3-64-1` in VGG-13. Top row shows a network trained with BN before and after its normalization is applied to data from the CIFAR-10 test split. The bottom row shows how the layer's distribution is transformed when the network was trained with MN instead, in which case the multimodal input appears to be normalized more flexibly.

Bengio et al. (2015) and Shazeer et al. (2017) propose the use of additional losses that either prevent all samples to focus on a single gate, encourage sparsity in the gate activations, or enforce variance over gate assignments. In ablations with MN for CIFAR-10 and CIFAR-100, the introduction of such additional penalties did not improve performance, and they are therefore not introduced alongside MN.

MN generalizes BN, which can be recovered as a special case by setting $K = 1$, or if the gates collapse: $[g(x_n)]_k = \text{const.}\ \forall\, k, n$. Importantly, MN should be able to seek out a form in which it passes all examples through a single gate, in particular when the input distribution at a particular layer is not multimodal. In the experimental evaluations (see Section 2.4), such gate behavior was however not observed: samples are usually assigned to individual modes, and gates tend to receive a relatively even share of samples overall.

To demonstrate how the extra flexibility of MN helps, Figure 2.3 shows histograms for the feature values found in the first three channels after application of the convolution associated with layer `conv3-64-1` of VGG-13 (Simonyan and Zisserman, 2015) trained with BN (top) and MN with $K = 2$ (bottom). Histograms are collected for 1024 test

Algorithm 3 Mode group normalization.

Input: parameter K, channel-wise feature vector x_c with $c = 1, \ldots, \dim(C)$, small ε, learnable channel-wise parameters α, β and $\Psi\colon \mathbb{R} \to \mathbb{R}^K$.

Compute channel-wise gates:

$$g_{ck} \leftarrow \left[\sigma \circ \Psi(x_c)\right]_k$$

for $k = 1$ to K **do**

Update estimators and normalize:

$$\mu_k \leftarrow \langle x \rangle_k$$
$$\sigma_k^2 \leftarrow \langle x^2 \rangle_k - \langle x \rangle_k^2$$
$$y_k \leftarrow \frac{x - \mu_k}{\sqrt{\sigma_k^2 + \varepsilon}}$$

end for

Return: $\frac{\alpha}{K} \sum_k y_k + \beta$

samples from CIFAR-10, both before (left) and after (right) the normalization via BN and MN. MN is able to normalize channels that show multimodal behavior, something that is not possible in the transformation that underlies BN.

2.3.3 Mode Group Normalization

As discussed in Section 2.2.4, GN is less sensitive to the batch size (Wu and He, 2018). Here it is shown that similarly to BN, GN can also benefit from soft assignments into different modes. In contrast to BN, GN computes averages over individual samples instead of the entire mini-batch. This makes modifications necessary, resulting in mode group normalization (MGN, Algorithm 3).

In MGN a gating network $g\colon \mathbb{R} \to [0, 1]^K$ is learned that associates channels with K modes. After average-pooling across height and width of a feature vector, estimators are computed by a weighted average over channel values $x_c \in \mathbb{R}$ and $c = 1, \ldots, \dim(C)$, for example for the mean $\mu_k = \langle x \rangle_k = C_k^{-1} \sum_c [g(x_c)]_k \cdot x_c$, where $C_k = \sum_c [g(x_c)]_k$.

This replaces the fixed group assignments in GN (c.f. Section 2.2.4) with a learnable grouping. Each feature vector x is subsequently transformed as:

$$\mathrm{MGN}(x) \triangleq \frac{\alpha}{K} \sum_{k} \frac{x - \mu_k}{\sigma_k} + \beta, \tag{2.7}$$

where α and β are learnable channel-wise parameters just as in BN (c.f. eq. 2.3). One of the notable advantages of MGN (that it shares with GN) is that inputs are transformed in the same way during training and inference, and that each sample is handled individually (i.e. there is no estimation alongside the batch dimension, c.f. Figure 2.1), whereby the small batch size limitation of BN is circumvented. As with MN, the computation underlying MGN is light-weight, such that it may be inserted throughout all layers of a deep network.

The gating in MGN can be interpreted as a clustering mechanism that groups together different channels. A potential risk for such approaches is that clusters or modes might collapse into one Xu et al. (2005). Although it is possible to address this with a regularizer, it has not been an issue in the experiments of Section 2.4. This is likely a consequence of the large dimensionality of feature spaces that these normalizations are applied to in this thesis, as well as sufficient levels of variation in the data.

2.4 Experiments

Two experimental settings are considered to evaluate the proposed methods: multi-domain (Section 2.4.1) and single task (Section 2.4.2). All experiments use standard routines within PyTorch (Paszke et al., 2019).

2.4.1 Multi-Domain

Data In the first experiment, heterogeneity is enforced in the data distribution via an explicit design of the distribution as $\mathbb{P} = \sum_d \pi_d \mathbb{P}_d$ with diverse $\mathbb{P}_d$, and $\sum_d \pi_d = 1$. This is realized through a dataset whose images are a combination of four image datasets: (i.) **MNIST** (LeCun, 1998) which contains grayscale scans of handwritten digits. The dataset has a total of 60 000 training samples, as well as 10 000 samples set aside for validation. (ii.) **CIFAR-10** (Krizhevsky and Hinton, 2009) is a dataset of colored images that show real-world objects of one of ten classes. It contains 50 000 training and 10 000 test images. (iii.) **SVHN** (Netzer et al., 2011) is a real-world dataset consisting of 73 257 training samples, and 26 032 samples for testing. Each image shows one of ten digits in natural scenes. (iv.) **Fashion-MNIST** (Xiao et al., 2017) consists of the same number of single-channel images as are contained in MNIST. The images contain fashion items such as sneakers, sandals, or dresses instead of digits as object classes.

It is assumed that labels are mutually exclusive, and a single network — LeNet (LeCun et al., 1989) with a 40-way classifier at the end — is trained to jointly learn predictions on them. LeNet is chosen here for a set of initial exploratory trials using a relatively simple network, and the main findings are then evaluated for more advanced architectures in Section 2.4.2. Inserting normalizations after convolutions and before the activation gave the best performance overall, and follows a common layout for example also found in residual networks (He et al., 2016).

Mode Normalization All models are trained for 3.5 million data touches (15 epochs; performance did not improve significantly when extending training by up to 5 million touches), with learning rate reductions by 1/10 after 2.5 and 3 million data touches. The batch size was set to $N = 128$ for which BN receives a sufficiently large set of candidates for estimation and has been reported to perform well (He et al., 2016). Running estimates were kept with $\lambda = 0.1$, the default value in PyTorch (Paszke et al., 2019).

The number of modes in MN is varied over $K = \{2, 4, 6\}$. The weights in MN were

BN	IN	LN	MN	K
26.91 ± 1.08	28.87 ± 2.28	27.31 ± 0.71	<u>23.16</u> ± 1.23	2
			24.25 ± 0.71	4
			25.12 ± 1.48	6

Table 2.1: Test set error rates (%) of batch norm (BN), instance norm (IN, Ulyanov et al., 2017), layer norm (LN, Ba et al., 2016), and mode norm (MN) in the multi-domain setting for a batch size of $N = 128$. Shown are average top performances over five initializations alongside standard deviations. Additional results for $N = \{256, 512\}$ are shown in Table 2.2.

initialized using the default Kaiming initializer (He et al., 2015), which was found to give stable results for the standard architectures and optimization settings evaluated in this and subsequent sections. Note MN replaces every BN in the model, not just the initial normalization. Average performances over five seeds and standard deviations are shown in Table 2.1: MN outperforms BN, and all other normalizations.

The additional overhead of MN is small; however increasing K did not always improve the results. The performance of higher mode numbers can be reduced as a result of statistics being estimated from smaller and smaller partitions of the batch, a known issue that also occurs for small batch sizes in traditional BN.

Experiments with larger batch sizes, shown in Table 2.2, support this argument. While the same network and hyperparameters are used as in the previous multi-domain experiment, the batch size is varied over $N = \{256, 512\}$ here.

For larger batch sizes, increasing K to values larger than two can increase performance, while for a smaller batch size of $N = 128$ (c.f. Table 2.1), errors incurred by finite estimation prevent this benefit from appearing. In all remaining trials, which involve single datasets and deeper networks, $K = 2$ is therefore fixed.

Mode Group Normalization GN is designed specifically for applications in which large batch sizes become prohibitive. In experiments this regime was simulated by reducing batch sizes to $N = \{4, 8, 16\}$, and training each model for 50 000 gradient

N	**BN**	**IN**	**LN**	**MN**	K
256	26.34 ± 1.82	31.15 ± 3.46	26.95 ± 2.51	25.29 ± 1.31	2
				25.04 ± 1.88	4
				<u>24.88</u> ± 1.24	6
512	26.51 ± 1.15	29.00 ± 1.85	28.98 ± 1.32	26.18 ± 1.86	2
				<u>24.29</u> ± 1.82	4
				25.33 ± 1.33	6

Table 2.2: Test set error rates (%) of multiple normalization methods in the multi-domain setting for large batch sizes. The table contains average performances over five initializations, alongside their standard deviation.

N	**BN**	**MN**	**GN**	**MGN**
4	33.40 ± 0.75	32.80 ± 1.59	32.15 ± 1.10	31.30 ± 1.65
8	31.98 ± 1.53	29.05 ± 1.51	28.60 ± 1.45	26.83 ± 1.34
16	30.38 ± 0.60	28.70 ± 0.68	27.63 ± 0.45	<u>26.00</u> ± 1.68

Table 2.3: Test set error rates (%) for BN, MN, mode group norm (MGN) and group norm (GN) for small batch sizes. Shown are average top performances over five initializations alongside standard deviations.

updates. The same configuration as before is used, except for a smaller initial learning rate $\gamma = 0.02$, which was reduced by 1/10 after 35 000 and 42 500 updates. Note that in MGN networks all normalizations are replaced in the architecture.

In GN, two groups were allocated per layer. Accordingly $K = 2$ was fixed in MGN to ensure channel statistics are estimated from similar group sizes. As additional baselines, results for BN and MN were also included. Average performances over five initializations and their standard deviations are shown in Table 2.3. As previously reported BN failed to maintain its performance when the batch size is small (Wu and He, 2018).

Though MN performed slightly better than BN, its performance also degraded in this regime. GN is more robust to small batch sizes, and MGN further improved over GN. This result highlights two important aspects: that MoE-based gating may be used to formulate new normalizations beyond BN/MN, and that such gates can benefit learning in the context of small batch sizes.

	Network In Network			VGG13	
	Lin et al.	**BN**	**MN**	**BN**	**MN**
CIFAR-10	8.81	8.82	8.42	8.28	7.79
CIFAR-100	–	32.30	31.66	31.15	30.06

Table 2.4: Test set error rates (%) with BN and MN for NIN and VGG13.

2.4.2 Single Task

Data Here MN is evaluated in standard image classification tasks, showing that it can be used to improve performance in different convolutional architectures. For this, MN is inserted into multiple architectures, and evaluated on **CIFAR-10** and **CIFAR-100**, as well as on the large-scale, multimodal **ILSVRC12** challenge (Deng et al., 2009). Unlike CIFAR-10, CIFAR-100 has 100 classes, but contains the same number of training images (600 images per class). ILSVRC12 contains around 1.2 million images from 1000 object categories.

Network In Network Since the original Network In Network (NIN, Lin et al., 2014a) does not contain any normalization layers, the network architecture is modified to add them, coupling each convolutional layer with a normalization layer (either BN or MN).

On CIFAR-10 and CIFAR-100 all models are trained for 100 epochs with SGD and momentum, using a batch size of $N = 128$. Initial learning rates were set to $\gamma = 0.1$, and reduced by 1/10 at epochs 65 and 80. Running averages were stored with $\lambda = 0.1$. During training images were flipped horizontally, and each image is cropped after padding it with four pixels on each side.

Dropout (Srivastava et al., 2014) is known to occasionally cause issues in combination with BN (Li et al., 2019b), and reducing it to 0.25 (as opposed to 0.5 in the original publication) improved performance.

Error rates on the test set for NIN on CIFAR-10 and CIFAR-100 are reported in Table

	ResNet20			ResNet56		
	He et al.	**BN**	**MN**	**He et al.**	**BN**	**MN**
CIFAR-10	8.75	8.44	7.99	6.97	6.87	6.47
CIFAR-100	–	32.24	31.52	–	29.70	28.69

Table 2.5: Test error (%) for ResNet20, ResNet56 normalized with BN and MN.

2.4 (left). NIN with BN obtains an error rate similar to that reported for the original network in Lin et al. (2014a), while MN ($K = 2$) achieves an additional boost of 0.4% and 0.6% over BN on CIFAR-10 and CIFAR-100, respectively.

VGG Networks Another popular class of deep convolutional neural networks are VGG networks (Simonyan and Zisserman, 2015). For the experiments, a VGG-13 with BN and MN is trained on CIFAR-10 and CIFAR-100. Models on both datasets are optimized using SGD with momentum for 100 epochs, setting the initial learning rate to $\gamma = 0.1$, and reducing it at epochs 65, 80, and 90 by a factor of 1/10. The batch size was $N = 128$. As before, the number of modes in MN is set to $K = 2$, and estimators are kept with $\lambda = 0.1$.

When incorporated into the network, MN improves the performance of VGG-13 by 0.4% on CIFAR-10, and over 1% on CIFAR-100 (see Table 2.4, right).

Residual Networks Residual Networks (He et al., 2016) include layer-wise batch normalization by default. For the trials shown here a ResNet20 is trained on CIFAR-10 and CIFAR-100 in its original architecture (i.e. with BN), as well as with MN ($K = 2$) (see Table 2.5, left). For both datasets a standard training procedure is followed, in which models are optimized for 160 epochs using SGD with a momentum parameter of 0.9, and weight decay of 10^{-4}. Running estimates are kept with $\lambda = 0.1$, and the batch size is set to $N = 128$.

The implementation of ResNet20 shown here (BN in Table 2.5) performs slightly better than that reported in the original publication (8.44% versus 8.75%). Replacing BN

Top-k Error	BN	MN
1	30.25	30.07
5	10.90	10.65

Table 2.6: Top-1 and top-5 error rates (%) of ResNet18 on ImageNet ILSVRC12, with BN and MN.

with MN in all layers of the residual network achieves a notable 0.45% and 0.72% performance gain over BN in CIFAR-10 and CIFAR-100, respectively.

Using the same setup as for ResNet20, additional trials used a deeper ResNet56. As shown in Table 2.5 (right), for these replacing all normalization layers with MN resulted in an improvement over BN of roughly 0.5% on CIFAR-10, and 1% on CIFAR-100.

MN is also evaluated in the large-scale image recognition task of ILSVRC12. Concretely, for the model with MN ($K = 2$) all BNs in a ResNet18 are replaced. Training is carried out for 90 epochs following He et al. (2016). The initial learning rate is set to $\gamma = 0.1$, reducing it at epochs 30 and 60 by a factor of 1/10. SGD was used as the optimizer (with momentum parameter set to 0.9, weight decay of 10^{-4}). To accelerate training models were distributed over four GPUs, with a global batch size of $N = 256$.

As can be seen from Table 2.6, MN results in a small but consistent improvement over BN in terms of top-1 and top-5 errors.

Qualitative Analysis Figure 2.4 shows which samples are assigned to which gate component in MN ($K = 2$) for images from the CIFAR-10 test split in layers `conv3-64-1` and `conv-3-256-1` of VGG-13. In particular, the figure displays the images x that have been assigned to either $[g(x)]_1$ (left) or $[g(x)]_2$ (right) with the highest scores.

In the normalization belonging to `conv3-64-1`, MN appears sensitive to a red-blue color mode, and images are assigned accordingly. In deeper layers like `conv-3-256-1` which are often associated with more semantic representations (Yosinski et al., 2014; Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2016), separations seem to occur on

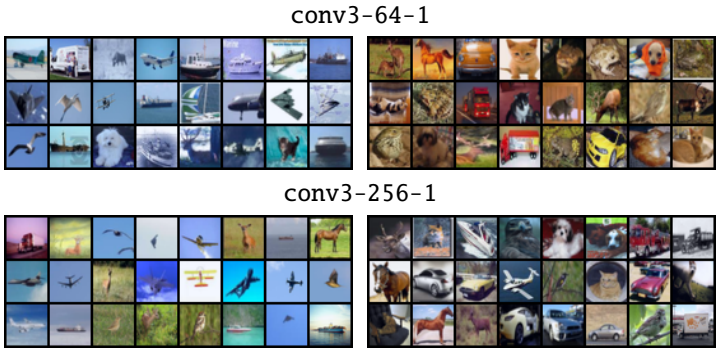


Figure 2.4: Examples from the CIFAR-10 test split for a VGG-13 trained with MN ($K = 2$) inserted, which have been assigned the highest scores towards either gate. In an early layer (top), color (blue vs. red) seems indicative of which gate examples get sent to, while in a deeper layer (bottom) there is a notable difference in object sizes.

the semantic level, and MN appears to split between smaller objects and larger ones.

2.5 Conclusion and Limitations

Stabilizing the training process of deep neural networks is a challenging problem. MN allows networks to dynamically normalize its features for multiple, visually diverse modes. The experiments in Section 2.4 demonstrated that accounting for modality in this way yields a consistent improvement in classification performance across various deep learning benchmarks and architectures.

An important ingredient in MN is its efficiency, allowing the module to be inserted at every layer of the network. An alternative idea is to employ some local module that accounts for multimodality in a pre-allocated layer (or a subset of them). Such an approach to normalization was introduced in Kalayeh and Shah (2019), who employ a Gaussian mixture model (GMM) to separate out different modes in data.

Experiments around this theme clearly indicated that using multiple GMMs to compute mode assignments in a deep network is not feasible, as each module requires a dedicated

outer optimization loop. Moreover, networks are known to process low-level and high-level features at different depths (Yosinski et al., 2014; Zeiler and Fergus, 2014), and hence modes or domains may be more separable at some layers than others. Pinpointing in advance which layers require more sharing or less is difficult, as this can change throughout model training, and depends on the exact characteristics of the data-generating distribution. By inserting an efficient module, like MN, at every layer, decisions of where to place normalizations are instead made in an automated fashion.

From this viewpoint, multimodal distributions seem to benefit from modules that can be inserted at multiple stages of the network, where they may exhibit local sharing or separating of features associated with different domains. While initially developed with MN in mind, these conceptual guidelines of end-to-end optimization, efficiency, and non-locality all resurface in the latent domain methods proposed in Chapter 3.

Another aspect to highlight in MN is its transferability to traditional classification problems, resulting for example in performance improvements for ImageNet classification (Deng et al., 2009). This aspect also reappears in Chapter 3, where new multi-domain strategies are introduced that require no domain annotations, and can therefore be deployed in classification problems related to fairness (Section 3.4.2) and long-tailed recognition (Section 3.4.3).

A notable work onto which MN has had an impact is that of Liu et al. (2020a) which proposes a simultaneous defense strategy against different types of adversarial attacks, e.g. $\ell_\infty, \ell_1, \ell_2$ attacks (Tramer and Boneh, 2019; Maini et al., 2020).[2]

A key insight in their work is that each attack type differs significantly from others in some part of the model's feature representation, and Liu et al. (2020a) account for this via a gated normalization routine, targeting an individual BN for each adversarial attack type, plus one for clean examples. The authors show that this approach significantly boosts the success rate at defending against attacks.

[2]Adversarial attacks on a sample pair (x, y) are defined by $f_\theta(x_\epsilon) \neq y$ s.t. $\|x - x_\epsilon\|_p \leq \epsilon$. The type of adversarial attack can be categorized by the ℓ_p-norm that is applied to measure the distance between x and x_ϵ.

One restriction in this work is the dependency on the type of adversarial attack being known, required to control their gated BN during training. At test time on the other hand their proposed normalization coincides with MN, i.e. it uses a soft gating mechanism to combine together individual normalizations.

The central concept of MN, to enrich the normalization unit with an activation mechanism, also entered Li et al. (2020). They compute a single estimate of μ, σ from each batch, and combine this with a mixture of channel-wise transformations, introducing multiple $\{(\alpha_k, \beta_k)\}_{k=1,\ldots,K}$ instead of the singular α, β in eq. (2.3), and using learnable mixture weights λ_k that combine the different channel-wise transformations into a generalized, more flexible affine transformation.

Ablations in their work showed that a flexible normalization contributes favorably to classification performance and can replace and sometimes surpass squeeze-and-excitation units (Hu et al., 2018). While these share some similarity with MN and also introduce a channel-wise mapping from an average-pooled representation of the feature activations in residual networks, they do not combine the outputs of multiple expert learners as in MoE models.

In this chapter flexible normalizations, such as MN, were shown to help performance when dealing with data from heterogeneous sources. The next chapter extends and formalizes this problem in a new transfer learning setting called latent domain learning, in which models are learned over multiple domains without annotation.

Chapter 3

Latent Domain Learning

A fundamental shortcoming of deep neural networks is their specialization to a single task and domain. While recent techniques in multi-domain learning enable the learning of more domain-agnostic features, their success typically relies on the presence of domain labels, requiring careful curation of datasets. This chapter formalizes a highly practical but less explored scenario: learning from data from different domains, without access to domain annotations.

The chapter begins with a detailed background review in Section 3.1 that summarizes related transfer learning settings over multiple tasks and domains, and highlights popular strategies for each.

Section 3.2 introduces latent domain learning, a multimodal learning problem that contains data from different unannotated visual domains. Two novel methods for this setting are introduced in Section 3.3, followed by experiments in Section 3.4, and a conclusion in Section 3.5.

3.1 Background

While the performance of deep learning has surpassed that of humans in a range of tasks, machine learning models tend to perform best when learning objectives are narrowly defined (Vandenhende et al., 2020). Practical realities however often require the learning of joint models over semantically different domains, for example when attempting to understand entire scenes (Zhu et al., 2016; Xiao et al., 2018), or in systems that require robust representations that can be used to jointly solve multiple perception tasks, e.g. localization and classification (Bilen and Vedaldi, 2016).

This section introduces the most prominent categories of machine learning that involve multiple tasks or domains: a discussion of the broader transfer learning scenario (Section 3.1.1) is followed by introductions of the inductive transfer setting (Section 3.1.2), domain adaptation (Section 3.1.3), multi-task learning (Section 3.1.4), continual learning (Section 3.1.5), multi-domain learning (Section 3.1.6), and domain generalization (Section 3.1.7).

3.1.1 Transfer Learning

Recent years have shown that machine learning models can match or even outperform human performance in tasks like image object classification (Russakovsky et al., 2015; Dosovitskiy et al., 2021; Pham et al., 2021). Transfer learning is broadly concerned with how to best reuse the knowledge acquired in such powerful models in other tasks.

A well-known strategy that illustrates the transfer-based learning protocol is that of pretraining and finetuning: a model is trained on some task that is usually large, varied, and highly informative, for example ImageNet (Deng et al., 2009). This is called the pretraining stage. Then, because labeled data may be scarce or non-existent in some other task, new models are initialized with the pretrained model parameters, and subsequent learning of the target task (the finetuning stage) adapts the model parameters $\theta \in \Theta$ (or a subset of them).

Finetuning is an established concept (Girshick et al., 2014) that is often examined in empirical studies that assess its suitability in different applications (Kornblith et al., 2019; Raghu et al., 2019). A recent trend is to propose more flexible treatments of finetuning, e.g. dynamically routing samples to pretrained or finetuned model parameters (Guo et al., 2019b), aligning pretraining and finetuning by comparing label distributions (Tran et al., 2019), or estimating transferability of models in advance (Nguyen et al., 2020). Another recent study goes one step further and investigates ways to combine parameters from a collection of models, in a bid to leverage the increased availability of such large, pretrained architectures (Shu et al., 2021).

It is important to note that knowledge transfer between tasks can have detrimental effects (Zhang et al., 2020), and performance can be negatively impacted due to so-called negative transfer, a term rooted in behavioral psychology (Postman and Stark, 1969). In particular this is known to occur when source and target tasks are too dissimilar, e.g. predictions have been shown to degrade when transferring models between very dissimilar subpopulations (Rosenstein et al., 2005).

While Zamir et al. (2018) have conducted a rigorous survey that investigates the relationship between different tasks, what constitutes a positive or negative relationship, and how this depends on the underlying model class or optimization choices, remains a largely unanswered question. This is an important aspect in latent domain learning as well (introduced in Section 3.2), where one aims to learn models that can both leverage synergies between domains, or separate them from one another.

Several additional challenges, such as catastrophic forgetting (Kirkpatrick et al., 2017), complicate transfer between tasks. Solutions to this problem have been developed in the context of continual learning (discussed in Section 3.1.5), and catastrophic forgetting also is an important element in Chapter 4, where new transfer-based anomaly detection methods are introduced that counter this.

Transfer learning is a fundamental machine learning problem, with early work going back to (Caruana, 1997; Thrun and Pratt, 1998). Besides the commonly used pretrain-

finetune sequence, there exist many additional transfer learning settings, each of which is associated with subtle differences and individual methods. Next, Sections 3.1.2 to 3.1.7 introduce the most important subcategories of transfer learning.

3.1.2 Inductive and Transductive Transfer Learning

Transfer always occurs between the so-called source and the target. Both of these are associated with domains $\mathcal{D}_S$ and $\mathcal{D}_T$, respectively. According to Pan and Yang (2009) these are defined by some marginal distribution, e.g. $\mathbb{P}(X_S)$, over a random variable X_S that resides in some space $\mathcal{X}_S$. Source and target are additionally associated with tasks $\mathcal{T}_S$ and $\mathcal{T}_T$, such as regression or depth estimation. One important condition for transfer-based learning is some relatedness between source and target, a relationship that is however usually implicitly defined (Pan and Yang, 2009), or approximated in some fashion (Tran et al., 2019).

Subcategories of transfer learning can be categorized by equivalence relations between the distributions associated with source and target (Pan and Yang, 2009). In inductive transfer learning the marginal distributions are equivalent and $\mathbb{P}(X_S) = \mathbb{P}(X_T)$, but the tasks between source and target differ, i.e. $\mathcal{T}_S \neq \mathcal{T}_T$. For example, the source task may consist of classification, while the target task is to carry out semantic segmentation. In transductive transfer learning the source and target asks are the same (say, classification), but the marginal distributions differ. Moreover, while in inductive transfer learning there is labeled data from the target domain, in the transductive setting this is typically not the case (Arnold et al., 2007).

3.1.3 Domain Adaptation

In recent years the narrow scope of datasets has been widely questioned (Torralba and Efros, 2011; Tommasi et al., 2017; Recht et al., 2019). Addressing some of these limitations has become an active area of research: one theme that formulates broader

learning criteria is domain adaptation (DA) (Ganin et al., 2016; Tzeng et al., 2017; Xu et al., 2018; Peng et al., 2019a; Sun et al., 2019b), which aims to develop methods that can efficiently carry over representations learned in one dataset to another.

DA has a long history in machine learning. Notable works in the classical literature studied how to best adapt models trained on news articles to handling biomedical documents instead (Daumé III, 2007), or how to adapt spam models to new users (Ben-David et al., 2007).

In DA, the source and target tasks are equivalent ($\mathcal{T}_S = \mathcal{T}_T$), but the underlying domains differ ($\mathcal{D}_S \neq \mathcal{D}_T$). For instance while the posteriors may be equivalent ($p_S(y|x) = p_T(y|x)$), differences could occur due to a change in distribution of the covariates, i.e. $p_S(x) \neq p_T(x)$, a phenomenon called covariate shift (Shimodaira, 2000; Bickel et al., 2009).

Beyond covariate shift DA may include additional types of shifts, such as differences between training and testing distributions due to label shift (Li et al., 2019c; Tachet des Combes et al., 2020), or because of some underlying change in measurement between datasets, e.g. images taken with different cameras and specifications (Storkey, 2009).

For example in the SVHN-to-MNIST benchmark popular in the literature (French et al., 2018; Hoffman et al., 2018) models are first trained on SVHN (Netzer et al., 2011) as the source distribution, and then adapted to MNIST digits (LeCun, 1998). While these two datasets show the same ground-truth objects, their images look very different due to having been were measured/captured differently. Note while the categories in both datasets are equivalent (digits between zero and nine), the number of examples available per class changes due to label shifts (Li et al., 2019a). Liang et al. (2020) address more extreme cases of this, e.g. some classes only appearing in the source domain, through adversarial alignment. A related line of work, where new classes appear in the target problem, is open set recognition (Bendale and Boult, 2016; Geng et al., 2020; Fang et al., 2021).

The problem of DA can be differentiated additionally by the availability of labeled data in the target domain. Unsupervised DA assumes no labels are available for the target domain, and has been tackled by encouraging domain-invariance at the gradient level (Ganin and Lempitsky, 2015), by generating features that can minimize the discrepancy between classifiers (Saito et al., 2018), or by introducing semantic losses over explicit (Xie et al., 2018) or implicit (Jiang et al., 2020) pseudo-labels. Semi-supervised DA (Donahue et al., 2013; Yao et al., 2015) introduces a subset of labeled examples, and has recently been approached through conditional entropy minimization (Saito et al., 2019). In supervised DA all examples from the target domain are labeled (Motiian et al., 2017).

Besides single-source DA, for example the SVHN-to-MNIST benchmark mentioned above, another recent trend is to focus on the problem of multi-source DA (Mansour et al., 2008; Sun et al., 2015; Zhao et al., 2018; Carlucci et al., 2020). In this setting multiple (labeled) domains with index $d = 1, \ldots, D$ constitute the input distribution as $\mathbb{P}_S = \sum_d \pi_d \mathbb{P}_{S,d}$, where samples from each domain appear with an associated relative share $\pi_d \in (0, 1)$, and $\sum_d \pi_d = 1$.

3.1.4 Multi-Task Learning

Multi-task learning is concerned with the development of machine learning methods that simultaneously generalize well over several tasks (Caruana, 1997; Zhang and Yang, 2018). In multi-task learning, tasks $\mathcal{T}_k$ with task index $k = 1, \ldots, K$ are optimized jointly, resulting in a minimization over a risk that contains multiple losses associated with each:

$$R_N[f_{\theta_1}, \ldots, f_{\theta_K}] = \frac{1}{N} \sum_k \sum_n \lambda_k L_k(f_{\theta_k}(x_n), y_{nk}), \tag{3.1}$$

where the task weights $\lambda_k \in \mathbb{R}_+$ control the relative importance of each task, and task-specific parameters $\theta_k \in \Theta_k$ are introduced (although many approaches share between subsets of the θ_k, see next paragraph). Note a different loss function L_k is associated with each task $\mathcal{T}_k$, an important differentiation from multi-domain learning discussed in Section 3.1.6.

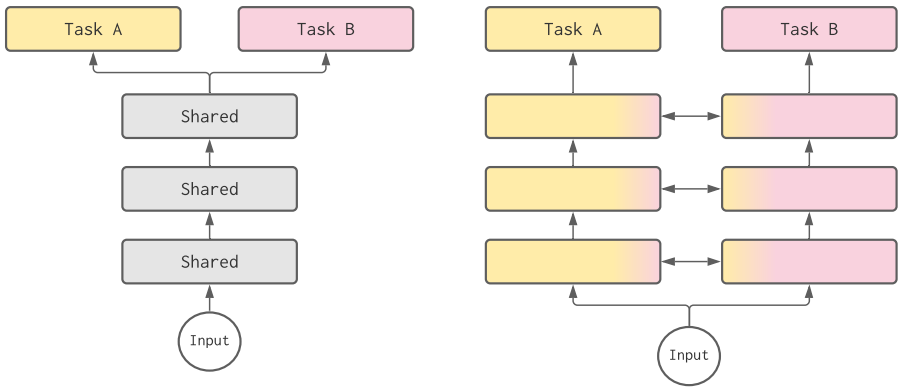


Figure 3.1: Hard and soft sharing in multi-task learning. In hard sharing (left), a subset of layers is shared across all tasks, followed by task-specific heads. Soft-sharing (right) reserves individual layers to each task, with some cross-talk mechanism that exchanges information between them.

A common theme in multi-task learning is to share some parameters, and dedicating others to a specific task. This can be achieved via hard sharing, in which the majority of layers are shared, but individual paths appear during the final stages of the network (Bilen and Vedaldi, 2016; Ranjan et al., 2017; Kokkinos, 2017; Dvornik et al., 2017).

A second option is soft sharing, where constraints enforce exchange between individual per-task layers (Misra et al., 2016; Ruder et al., 2019; Liu et al., 2019a). Recently adaptive sharing schemes have been proposed that combine both ideas (Sun et al., 2019a). The difference between hard vs. soft sharing is illustrated in Figure 3.1.

One line of work in multi-task learning is concerned with designing new optimization routines specific to the multi-task setting, in particular how to scale between tasks via λ_k in eq. (3.1). Such approaches are typically model-free, and include the idea of balancing losses in an automated fashion (Kendall et al., 2018), normalizing or adapting gradients (Chen et al., 2017), or prioritizing tasks dynamically by favoring hard tasks over easy ones (Guo et al., 2018).

While certain tasks are known to synergize and multi-task learning tends to work well for these, for example detection and classification (Girshick, 2015; Ren et al., 2015),

sometimes tasks are not particularly beneficial for each other (Zamir et al., 2018; Standley et al., 2020), which can result in multi-task models being outperformed by task-specific ones (He et al., 2017). To overcome this limitation, Kang et al. (2011) investigated automated learning rules for automatically selecting tasks that benefit one another.

Multi-task learning is focused on the goal of learning tasks jointly. Other learning problems require that tasks are learned in sequence, in particular online settings like continual learning (Hoi et al., 2018), which is introduced in the next section.

3.1.5 Continual Learning

When adapting models to new target domains or tasks without any form of regularization, then this does not necessarily preserve performance on the source task, a phenomenon called "catastrophic forgetting" (Kirkpatrick et al., 2017). This problem surfaces in continual or lifelong learning (Parisi et al., 2019), where one learns over sequences of tasks and observations associated with each.

A practical example can be found in robotics (Lesort et al., 2020), where performance may degrade on old tasks (e.g. forgetting how to pick up an object) due to having learned a new one (say, pulling a lever). Moreover, catastrophic forgetting also plays a central role in Chapter 4, where it is shown that the detection of semantic anomalies can benefit from limiting forgetting.

Lopez-Paz and Ranzato (2017) approach continual learning by restricting gradient steps that oppose gradients for past examples stored in a small episodic memory. This ensures that inner products between current and past gradients stay aligned, but requires solving a quadratic problem that scales in the number of tasks at every iteration of the optimizer. Chaudhry et al. (2018) show that this can be simplified by instead sampling a single batch from a memory bank to align gradient updates, which still results in strong performance in many continual tasks, while being significantly more efficient.

A second important contribution in Lopez-Paz and Ranzato (2017) was the introduction of specific metrics for the continual setting, such as backward and forward transfer. Custom metrics are also relevant in latent domain learning, and are introduced in Section 3.2.2 of this thesis.

Other notable works introduce modular networks for continual learning (Veniat et al., 2021), or use meta-learning (Hospedales et al., 2021) to encourage gradients that will positively align in the future, thereby making negative interference less likely (Riemer et al., 2019). Yet another popular strategy is to reformulate reinforcement learning techniques, such as experience replay (Rolnick et al., 2019; van de Ven et al., 2020).

3.1.6 Multi-Domain Learning

Multi-domain techniques focus on learning a single set of domain-agnostic representations that generalize across multiple domains. While multi-domain learning is closely related to the multi-task scenario, it can be disambiguated in two main regards: in multi-task learning, the nature of underlying tasks $\mathcal{T}_k$ and $k = 1, \ldots, K$ can inherently differ, such that learning an individual model f_{θ_k} is associated with an individual loss function L_k (for example, one task may be object classification, the other semantic segmentation, or depth estimation). In multi-domain learning on the other hand, all losses are associated with an equivalent problem type (e.g. classification).

This implies a second difference in the availability of the labels: in multi-task learning one usually has multiple labels per example, e.g. y_1 for classification, y_2 for semantic segmentation, etc., up to y_K. In multi-domain learning, each image has a distinct label y, but images can vary substantially depending on the visual characteristics of each domain they are associated with. As further differentiation, multi-task learning is contrasted against multi-domain learning in Figure 3.2.

The central assumption in multi-domain learning is that data $(x_1, \ldots, x_N) \in \mathcal{X}$ is sampled i.i.d. from some mixture of distributions that constitute the data-generating

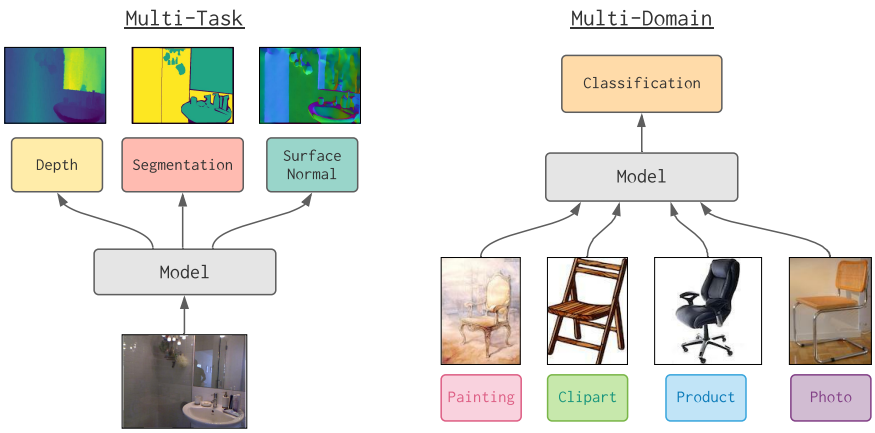


Figure 3.2: A visual comparison between multi-task learning, which includes multiple tasks per image, and multi-domain learning, which tends to focus on one task, but over image data with different visual characteristics.

distribution as $\mathbb{P} = \sum_d \pi_d \mathbb{P}_d$ with domain indices $d = 1, \ldots, D$, where each domain is associated with a relative share $\pi_d = N_d / (N_1 + \cdots + N_D)$, and N_d denotes the number of examples belonging to the d'th domain.

Two multi-domain settings exist. The first setting includes mutually exclusive classes and a disjoint label space $\mathcal{Y}_1 \cup \cdots \cup \mathcal{Y}_D$ that encompasses all domains. A popular benchmark that falls into this category is Visual Decathlon (Rebuffi et al., 2017), which combines different datasets into one. This setting also appears in the evaluation of MN in Section 2.4.1.

Opposed to this, in some settings label spaces are shared and $\mathcal{Y}_d = \mathcal{Y}_{d'}$ for all domains, for example cars driving in different conditions (Alberti et al., 2020), or elephants that are depicted as photos or paintings (Li et al., 2017). Section 3.4 studies this setting in latent domain experiments.

The problem of multi-domain learning is closely connected to the hypothesis of universal representations (Bilen and Vedaldi, 2017), which posits that models that have learned a sufficiently complex semantic representation, obtained e.g. on ImageNet (Deng et al.,

2009), should be able to generalize to new distributions easily via simple but adequate low-capacity transformations.

In their paper, Bilen and Vedaldi (2017) aim to learn classifiers over very different image data, such as the real-world imagery of Caltech-256 (Griffin et al., 2007), the Daimler mono pedestrian classification benchmark (Munder and Gavrila, 2006), or Omniglot which contains hand-written symbols from different alphabets (Lake et al., 2015). They investigate different sharing schemes for this, and recommend tackling such joint learning problem via adequate normalization, an idea that also influenced MN proposed in Chapter 2 of this thesis.

How to share parameters as efficiently as possible is a crucial question in multi-domain learning (and multi-task learning, for that matter). For this, Rebuffi et al. (2017) proposed new dedicated per-domain model parameters that serve as corrections, inserted sequentially at every layer of the network. This builds on-top of residual networks (He et al., 2016), and the concept was extended from sequential to parallel corrections later on (Rebuffi et al., 2018). While slightly more expensive than e.g. separate batch normalization layers for each domain, the authors show that this is highly effective at learning multiple visual domains jointly, and that such layer-wise adaptations constitute an improved and efficient finetuning strategies, in particular for small datasets.

Subsequent works tended to follow the same principle of sharing some parameters, while reserving others to specific domains. For example Berriel et al. (2019) propose a budget-aware adaptation strategy that increases or reduces the amount of domain-specific parameters depending on the computational resources available.

Another notable approach is that of Guo et al. (2019a), who equip models with depth-wise separable convolutions (Chollet, 2017; Howard et al., 2017; Sandler et al., 2018). This decomposes the traditional convolution (LeCun et al., 1998a) into a per-channel, depth-wise component (that leaves the channel number intact), followed by a point-wise convolution across channels that is applied across all channels (and may be used to modulate the number of channels within the network). Under the assumption that there

exist domain-specific spatial correlations while cross-channel correlations are shared, Guo et al. (2019a) recommend using an individual depth-wise convolution for each domain, while sharing the point-wise transformation.

Other work for multi-domain problems makes use of task-specific attention mechanisms (Liu et al., 2019a), employs masking strategies (Mancini et al., 2020b), normalizes the covariance of feature maps (Li and Vasconcelos, 2019), or extends the principle of universal representations via self-training (Tamaazousti et al., 2019). There has also been some interest in applying such concepts in the realm of language models, where Stickland and Murray (2019) recently proposed a multi-domain extension of BERT (Devlin et al., 2019) to obtain efficient models for related language tasks.

Note that in existing multi-domain literature it is typically assumed that domain labels are available for all samples (Nam and Han, 2016; Rebuffi et al., 2017, 2018; Bulat et al., 2019; Guo et al., 2019a). This central assumption is not fulfilled in latent domain learning and differentiates the settings, see Section 3.2.

3.1.7 Domain Generalization

In domain generalization (DG) models are learned on a multi-source mixture $\mathbb{P} = \sum_d \pi_d \mathbb{P}_d$ of domains with relative share $\pi_d \in (0, 1)$ and $d = 1, \ldots, D$. The main task is to generalize on the distribution associated with a new domain $\mathbb{P}_{D+1}$, without the use of samples from this unseen domain. As such, DG can be considered an extreme case of domain adaptation (Section 3.1.3), without access to *any* data from the target domain.

One set of methods in DG focuses on the idea of learning domain-invariant representations. Muandet et al. (2013) propose kernel-based measures to encourage domain-invariance, while Ghifary et al. (2015) employ autoencoding setups for this. Ganin et al. (2016) follow a different approach and learn feature representations that match across domains via domain discriminators, an idea that was extended via accuracy constraints in Akuzawa et al. (2019).

Other works use maximum mean discrepancy to tackle DG (Li et al., 2018b), match the per-domain conditional distribution (Li et al., 2018d,e), build on meta-learning (Balaji et al., 2018; Li et al., 2018a), use self-supervision (Carlucci et al., 2019), or propose new augmentation strategies (Shankar et al., 2019; Zhou et al., 2020; Borlino et al., 2020).

It is important to note that finding representations that generalize over unseen data is a highly difficult problem, and Gulrajani and Lopez-Paz (2020) questioned some of the recent progress in DG calling for stricter model selection criteria, as standard classifiers trained via empirical risk minimization were shown to outperform many existing DG methods under such standard criteria.

While latent domain learning is more closely connected to the multi-domain setting (see Section 3.1.6), an interesting aspect in recent DG research has been an investigation into the presence of unannotated domains by Matsuura and Harada (2020). This is reviewed in additional detail in Section 3.2.1, which summarizes and discusses literature related to latent domains.

3.2 Latent Domain Learning

While there exists no natural definition for what exactly a visual domain is, previous works in multi-domain learning assume that different subsets of data exist, with some defining characteristic that allows separating them from one another. Each subset, indexed by $d = 1, \ldots, D$, is then assigned to a pre-defined visual domain and vice-versa, multi-domain methods use such domain associations to parameterize their representations and learn some $p_\theta(y|x, d)$.

The assumption that domain labels are always available has been widely adopted in multi-domain learning (Rebuffi et al., 2017, 2018; Liu et al., 2019a; Guo et al., 2019a), however this assumption is not without difficulty.

For one, unless existing datasets are combined as in e.g. Rebuffi et al. (2017), their

Setting	Domain Labels	Section	Training Data	Evaluation Data
Unsupervised DA	Yes No	3.1.3 3.2.1	$S_1, \ldots, S_D, U_{D+1}$ $S_{\text{mixture}}, U_{D+1}$	U'_{D+1}
Semi-supervised DA	Yes	3.1.3	$S_1, \ldots, S_D, P_{D+1}$	U_{D+1}
Domain Generalization	Yes No	3.1.7 3.2.1	$S_1, \ldots, S_D$ S_{mixture}	U_{D+1}
Multi-Domain Learning	Yes	3.1.6	$S_1, \ldots, S_D$	$U_1, \ldots, U_D$
Latent Domain Learning	No	3.3.1–3.3.3	S_{mixture}	$U_1, \ldots, U_D$

Table 3.1: A comparison of unsupervised and semi-supervised DA, domain generalization, and multi-domain learning versus latent domain learning. S_d denotes a labeled dataset from the d'th domain, P and U correspond to partially labeled and unlabeled samples, respectively.

manual collection, labeling and curation can be very laborious. And, as below examples demonstrate, even if definitions are fixed as datasets are curated, it is unclear whether the chosen criteria for d are optimal.

In some cases domains are intuitive and their annotation straightforward. Consider a problem where images have little visual relationship, for example joint learning of Omniglot handwritten symbols (Lake et al., 2015) and CIFAR-10 objects (Krizhevsky and Hinton, 2009). In this case, it is safe to assume that encoding an explicit domain-specific identifier into p_θ is a good idea, and results in the multi-domain literature provide clear evidence that it is highly beneficial to do so (Rebuffi et al., 2018; Guo et al., 2019a; Mancini et al., 2020b).

The same is also true for other data sources: some datasets contain only professional photographs (Saenko et al., 2010), whereas others capture sketches or paintings (Li et al., 2017); some focus on entire scenes (Cordts et al., 2016), others focus on single objects (Venkateswara et al., 2017); some datasets contain images captured at different times, during day or night (Sultani et al., 2018). In each of these cases, labeling individual domains by their semantic context is more or less straightforward.

Consider a different example however, in which multiple domains are created from subsets of the same dataset (say, MNIST (LeCun, 1998)). In this counterfactual setup,

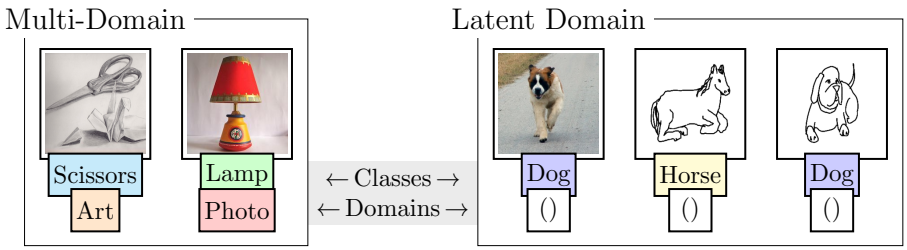


Figure 3.3: In multi-domain learning every sample has a domain label. Latent domain learning studies learning without this information.

learning individual parameters for each "domain" with no native sharing between them will result in sub-par performances for each.

While this setup may appear artificial, the work of Bouchacourt et al. (2018) considers semantic groupings of data: they show that when dividing data by subcategories, such as size, shape, etc., and incorporating this information into the model, then this benefits performance. Should one therefore also encode the number of objects into domains, or their color, shape, and so on?

Given the relatively loose requirement that domains are supposed to be different while related in some sense (Pan and Yang, 2009), these examples hint at the difficulty of deciding whether domains are needed, and – if the answer to that is yes – what the optimal domain criteria are. And note that even if such assignments are made very carefully for some problem, nothing guarantees that they will transfer effectively to some other task.

The remainder of this chapter investigates this ambiguity and studies two main questions: first, is learning separate parameters always the preferred strategy for multi-domain learning, regardless of the domains to be learned? And second, how can models best be learned without such labels while still allowing them to generalize well over visually diverse, multimodal data?

The associated setting is called *latent domain learning* in this thesis, and differs

fundamentally from existing learning settings over multiple domains. A comparison to the related problems introduced in Sections 3.1.1 to 3.1.7 is shown in Table 3.1. The difference between latent domain learning and multi-domain learning, the learning setting most closely related, is illustrated in Figure 3.3.

3.2.1 Related Work

While learning over latent domains was subject to some discussion in the classical literature, for example in Hoffman et al. (2012), in the deep learning era this aspect has received less attention. This is slowly starting to change, in particular in the context of multi-source DA (Mancini et al., 2018, 2019) and DG (Matsuura and Harada, 2020).

The earliest studies around latent domains appear in DA, and focus on recovering absent annotations, for example through hierarchical (Hoffman et al., 2012) or kernel-based clustering (Gong et al., 2013). Other works discover latent domains via exemplar SVMs (Xu et al., 2014), or by clustering through mutual information (Xiong et al., 2014). Once these are recovered, they can again be used to explicitly parametrize p_θ.

More recent studies around latent domains targeted DA, where Mancini et al. (2018) assume that a partial set of domain labels is available in the data, and use this to modify the normalization for better adaptation strategies over multiple source domains. This idea was later generalized to multi-source targets, alongside a new loss that stabilized mode-collapsing of domains into a single branch (Mancini et al., 2019). Another study investigates the transfer from multiple source domains (with domain labels) to target domains (containing latent domains) (Peng et al., 2019b).

A notable work that questioned the availability of domain labels in DG is that of Matsuura and Harada (2020), in which internal feature representations are extracted from a backbone and then used to assign pseudo-labels via clustering. By coupling their loss function with a domain-adversarial (Ganin and Lempitsky, 2015) term they encourage domain-invariance, learning more domain-agnostic representations, a common goal in

the DG literature (Muandet et al., 2013; Ghifary et al., 2015; Ganin et al., 2016).

While domain-invariance was shown to be desirable in the DG setting, in other scenarios seeking out domain-invariance can be overly restrictive, for example when there is a dissimilar domain in data that benefits from being treated separately. In fact Wang et al. (2020) found that enforcing domain-invariance often times confuses multi-domain classifiers and recommended against it.

Because of its reliance on domain annotations to separate out individual model parameters, multi-domain learning firmly depends on the existence of ground-truth domain labels. While above works have begun to address the problem of latent domains in other settings, such strategies have so far not been studied for multi-domain learning. Before introducing methods that aim to fill this gap in Section 3.3, the next section discusses useful metrics to assess multi-domain performance when no annotations are present.

3.2.2 Metrics

Crucially in latent domain learning there exists no explicit association between latent domains and the ground-truth domains annotated in some datasets. Instead latent domain models are optimized to produce the lowest training error, and do not necessarily recover ground-truth domain labels.

Bearing that in mind, if small or underrepresented domains can be identified, then one would still want to prevent outcomes in which severe performance losses occur on them. Traditional metrics often fail to capture these. Consider the *observed accuracy* when sampling i.i.d. from $\mathbb{P} = \pi_a \mathbb{P}_{d_a} + \pi_b \mathbb{P}_{d_b}$:

$$\mathrm{OAcc}[f] = \mathbb{E}_{(x_n, y_n)\sim\mathbb{P}}[\mathbb{1}_{y_f(x_n)=y_n}], \tag{3.2}$$

where y_f denotes the class assigned to sample x_n by the model f, and y_n its corresponding label for training. The OAcc has a problematic property: if $\mathbb{P}$ consists of two imbalanced

domains such that $\pi_a \geq \pi_b$, then the performance on d_a dominates it. This motivates alternative formulations for latent domain learning, as one should anticipate (and account for) imbalanced domains in real-world data.

To address this shortcoming, this thesis also measures models in *uniform accuracy* which decouples the accuracy from relative ground-truth domain sizes:

$$\mathrm{UAcc}[f] = \frac{1}{D}\sum_{d=1}^{D} \mathbb{E}_{(x_n,y_n)\sim\mathbb{P}_d}[\mathbb{1}_{y_f(x_n)=y_n}]. \tag{3.3}$$

For example if d_a has a 90% overall share, and the model perfectly classifies this domain while reaching 0% accuracy on d_b, then OAcc would still assume 0.9, hiding the underlying damage to domain d_b. Measuring latent domain performance uniformly reveals this damage as $\mathrm{UAcc} = 0.5$.

Note that while domain annotations are required in order to compute uniform accuracy, these should never be involved in training of latent domain models, and only be used for analyzing their performance in terms of UAcc.

The uniform accuracy has some important limitations of its own. The ultimate goal of latent domain learning is the development of methods that can entirely avoid the curation of ground-truth domain labels. These are however required in the computation of UAcc.

Because of this, one should always consider both OAcc and UAcc: the observed accuracy is straightforward to compute and informs how well data is estimated overall, whereas the uniform accuracy gives additional insights into potential failure modes of models, thereby assisting the development of new and robust latent domain methods.

3.3 Methods

While latent domains make for a highly practical problem for computer vision, it poses multiple challenges that have not been previously investigated in the context of

learning over multiple domains. This thesis introduces the following methodological contributions designed for latent domain learning:

- *Latent domain exchange* (Section 3.3.1), a class of augmentations that implicitly interpolates style information between latent domains.
- *Sparse latent adaptation* (Section 3.3.3), which enables models to dynamically adapt to instances from multiple latent domains in the data, without depending on domain annotations.

The proposed approaches can be optimized end-to-end, a property also associated with MN (introduced in Chapter 2), and require no separate clustering stage, a fundamental restriction in existing techniques designed for latent domains in related problems, such as DG (Matsuura and Harada, 2020).

Section 3.4 evaluates these methods in several settings that include latent domains, accompanied by a rigorous qualitative analysis that demonstrates that sparse adaptation partitions latent domains in intuitive ways. Because latent domain learning does not rely on the availability of domain labels, a notable benefit is that it can be applied to other classification settings, such as fairness problems (Section 3.4.2), or learning over imbalanced distributions (Section 3.4.3).

3.3.1 Latent Domain Exchange

Given a simple dataset containing two classes (say, dogs and giraffes) but multiple latent domains (sketches, photos, cartoons, etc.), the goal in latent domain learning is to learn some representation of images that allows for robust predictions irrespective of the underlying latent domain d.

In contrast to enforcing domain-invariance (Ganin et al., 2016) on the level of gradients, which has been reported to hamper performance in multi-domain settings (Wang et al.,

2020), this section proposes a strategy that takes inspiration from works that augment data using style transfer (Borlino et al., 2020; Zhou et al., 2021).

The goal of this augmentation is to decrease the importance of style in images, which tends to differ between domains (consider e.g. the occurrence of color in sketches vs. photos). At the same time, geometrical properties that characterize the objects (such as shape or pose) should remain unchanged.

This requires a mechanism that can disentangle style and content in images, which has previously been achieved through autoencoding (Mathieu et al., 2016; Kotovenko et al., 2019). Note the autoencoder is obtained in a completely separate pretraining step, which never sees data used in subsequent latent domain experiments (see Section 3.4).

To construct a suitable architecture for the encoder f_{enc}, recommendations from the style transfer literature, in particular of Huang and Belongie (2017), are followed: the encoding network is constructed from all layers up to `relu4-1` of VGG-19 (Simonyan and Zisserman, 2015). Training proceeds by randomly sampling pairs of content and style images c, s from COCO (Lin et al., 2014b) and Wikiart (Saleh and Elgammal, 2015) which are both mapped to the latent space of f_{enc} resulting in c^{enc} and s^{enc}. A latent code t^{enc} combining the content of c with the style of s is created by the following exchange mechanism:

$$t^{\text{enc}} = \sigma(s^{\text{enc}}) \frac{c^{\text{enc}} - \mu(c^{\text{enc}})}{\sigma(c^{\text{enc}})} + \mu(s^{\text{enc}}),$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation of latent codes after pooling across height and width. Next, a decoder f_{dec}, which mirrors the encoder (pooling layers are replaced with upsampling layers), converts t^{enc} back into an image t. To ensure that the encoder-decoder pair combines the content of c (say, a giraffe) with the style of s (e.g. a painting), an affine combination of two losses is optimized: a content loss measures the Euclidean distance between c^{enc} and the feature responses of the output image $f_{\text{enc}}(t)$, while the so-called style loss compares the feature similarity of t and s.

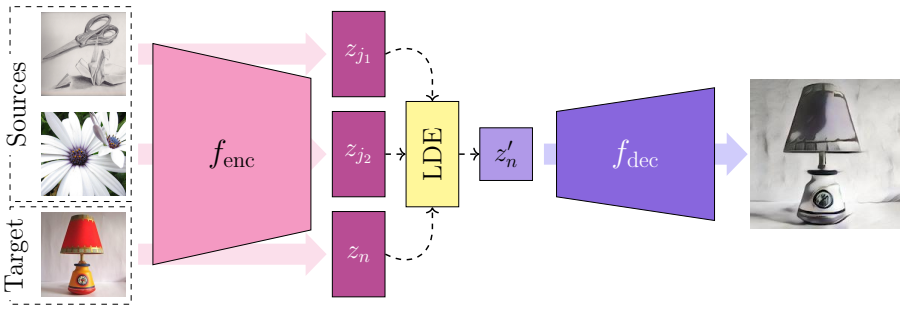


Figure 3.4: LDE augments the target x_n by exchanging style with multiple sources $(x_{j_1}, \ldots, x_{j_M})$. Shown here with $M = 2$. Note the encoder f_{enc} and decoder f_{dec} are pretrained separately in a neural style transfer setup (Huang and Belongie, 2017).

This autoencoding setup was shown to yield a robust pipeline for neural style transfer applications in Huang and Belongie (2017). With some modifications, it can be used as an augmentation: while the encoder f_{enc} and decoder f_{dec} remain fixed after pretraining, the proposed latent domain exchange (LDE) modifies the underlying exchange mechanism in eq. (3.3.1). In the following, two variations are presented: Random-LDE (RLDE), and Cluster-LDE (CLDE). A conceptual overview over these is provided in Figure 3.4.

RLDE Given a minibatch $\{x_n\}_{n=1,\ldots,N}$ during training, samples x_n are first encoded into a latent representation $z_n \in \mathcal{Z}$ using the encoder f_{enc}. RLDE groups each encoded example z_n with $M = 1, \ldots, N-1$ latent codes $z_{j_1}, \ldots, z_{j_M}$ randomly drawn (without replacement) from the same batch. Then the following transformation is applied:

$$\text{RLDE}(z_n) \triangleq \frac{1-\alpha}{M} \sum_{m=1}^{M} \left[\sigma(z_{j_m}) \frac{z_n - \mu(z_n)}{\sigma(z_n)} + \mu(z_{j_m}) \right] + \alpha z_n.$$

where as before $\mu(\cdot)$ and $\sigma(\cdot)$ denote the channel-wise statistics of latent codes after pooling across height and width. A preservation strength $\alpha \in [0, 1]$ is introduced that modulates the strength of the augmentation, and ablations for different values of M are in Table 3.5. RLDE mirrors the feature exchange used in the pretraining of f_{enc} and f_{dec} (c.f. eq. (3.3.1)), but different from before transfers the style information of multiple latent codes $z_{j_1}, \ldots, z_{j_M}$ onto z_n. The image generated through subsequent decoding

$f_{\text{dec}} \circ \text{RLDE}(z_n)$ therefore mixes the styles of $x_{j_1}, \ldots, x_{j_M}$ with content features of x_n.

Similar to how mixing many colors gives gray (thereby removing color), combining different styles from multiple examples as in RDLE should eventually yield some "average style" that promotes geometric object-level information (shapes, poses, etc.), while reducing the importance of the style of individual latent domains in data.

CLDE For some learning problems, such as latent domain learning over differing domains (see PACS in Section 3.4.1), LDE was found to benefit from a clustering ansatz.

For this, all latent codes $z_n \in \mathcal{Z}$ are initially clustered into one of $C = 2, \ldots, N$ clusters. Subsequent style exchange is carried out by randomly picking $C - 1$ codes from every cluster that z_n does *not* reside in. Different clustering mechanisms were explored for this, and CLDE was found to be stable regardless of the precise clustering variant that is chosen (see Table 3.5).

One benefit of CLDE is that it induces stratification between latent domains, particularly benefiting underrepresented ones: common styles will likely form large clusters, whereas unusual ones would belong to smaller ones. By always picking one example from each cluster the likelihood of unusual styles reappearing in other images increases.

3.3.2 Latent Adaptation

A central research question in this chapter is how to modify classification models to allow them to achieve robust performance when learning over data from multiple latent domains, but without domain annotations available in the data.

This section first reviews a strategy for when domain labels are available, which is then extended to cases without labels. Rebuffi et al. (2017, 2018) proposed to modulate networks by constraining the main transformation of residual network blocks (He et al., 2016) $\Phi(x) = x + f(x)$ to allow at most a linear change V_d for each domain from

some pretrained mapping Φ_0 (with f_0 in every layer), whereby $\Phi(x) - \Phi_0(x) = V_d x$. Note the slight abuse of notation here in letting x denote an image's feature activations. Rearranging this yields:

$$\Phi(x, d) = x + f_0(x) + \sum_{d=1}^{D} g_d V_d(x), \tag{3.4}$$

with a domain-supervised switch that assigns corrections to domains, i.e. $g_d = 1$ for d associated with x and 0 otherwise. Each V_d is parametrized through 1x1 convolutions, and f_0 denotes a shared 3x3 convolution obtained e.g. on ImageNet (Deng et al., 2009). This builds on the assumption that models with strong general-purpose representations require minimal changes to adapt to new tasks (Bilen and Vedaldi, 2017), making learning each V_d sufficient, while f_0 remains as is.

In latent domain learning access to d is removed, resulting in two new challenges: there is no a priori information about the right number of corrections, and domain labels cannot be used to decide which of the corrections to apply.

To mitigate the lack of domain labels d, it is assumed that input data is constituted by K *latent* distributions $\mathbb{P}_k$. A mixtures of experts approach (Jacobs et al., 1991; Jordan and Jacobs, 1994; Tresp, 2001) can be used to replace switches g_d with a gating mechanism $g\colon \mathcal{X} \to [0, 1]^K$ that assigns inputs x to latent domains, whereby the dependence on domain annotations is relaxed:

$$\Phi(x) = x + f_0(x) + \sum_{k=1}^{K} [g(x)]_k V_k(x), \tag{3.5}$$

The gates control which correction is applied to which example, and correspond to a categorical variable over K categories, i.e. $0 \leq [g(x)]_k \leq 1$ and $\sum_k [g(x)]_k = 1$ for all x. Note in particular how parametric dependency of Φ on d is removed by the gating mechanism. How to best choose the number of gates K, which replaces this dependency, is discussed in detail in Section 3.4.1.

While eq. (3.5) is motivated from latent domains, there is no guarantee that each V_k will

correspond to an actual visual domain and many additional factors (shape, pose, color, etc.) can enter them as well. Note the broader concept presented here may in principle also be incorporated with other dynamic concepts (Perez et al., 2018; Guo et al., 2019a), adaptation strategies however stand out due to their methodological simplicity.

Different options exist for parametrizing the gating function $g\colon \mathcal{X} \to \mathcal{G} \subseteq \mathbb{R}^K$. An ideal gating mechanism for latent domain learning would fulfill two seemingly incompatible properties: be able to filter domains in some layers (implemented via a discrete gate $g(x) \in \{0,1\}^K, \forall x$), but also share parameters between related domains in other layers (requiring smooth gates $g(x) \in [0,1]^K$). The next section proposes to resolve this conflict through sparseness.

3.3.3 Sparse Latent Adapters

The gating function $g\colon \mathcal{X} \to \mathcal{G} \subseteq \mathbb{R}^K$ is parametrized with a small linear transformation $W\colon C \to \mathbb{R}^K$ that constitutes the pre-activation within the gating branch, i.e. $q = W\varphi(x)$, where average pooling $\varphi\colon \mathcal{X} \to C$ is used to project onto the channels.

A crucial choice is whether the activation for $q \in \mathbb{R}^K$ should map to some *discrete* space $\mathcal{G} = \{0,1\}^K$ or a *continuous* $\mathcal{G} = [0,1]^K$ in which the V_k are shared.

A different strategy proposed here lets gates be smooth when appropriate, but a threshold allows discrete outputs $f_\tau(q) = [q - \tau]_+$ with $[\cdot]_+ = \max(0, \cdot)$. Crucially f_τ can be solved in a differentiable manner (Martins and Astudillo, 2016) by sorting $q_1 \geq \cdots \geq q_K$, solving $k^* = \max\{k \mid 1 + kq_k > \sum_{j \leq k} q_j\}$ and computing $\tau = [(\sum_{j \leq k^*} q_j) - 1]/k^*$.

Consider $q = [0.1, 1.0, 0.5]$ for which applying sparse activation results in $f_\tau(q) = [0.0, 0.75, 0.25]$. Compare this to the result of applying a softmax activation, which yields $[0.202, 0.497, 0.301]$. Sparse activation filters out q_1, while sharing between q_2

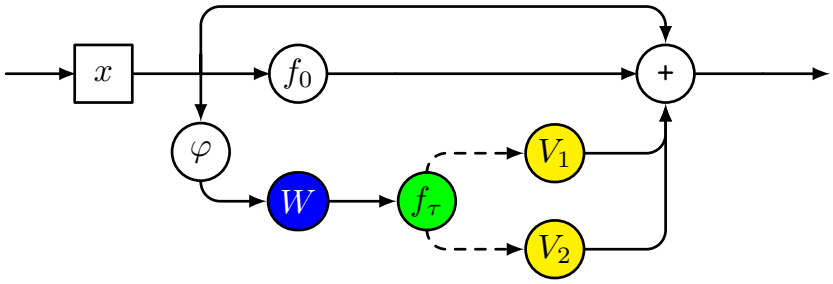


Figure 3.5: In the proposed residual latent domain architecture images x are passed down three streams: an identity function, a convolution f_0, and SLA ($K = 2$) which consists of φ-pooling onto the channels C, followed by a linear W-transformation and sparse activation f_τ before the corrections V_1 and V_2.

and q_3. This is used to define sparse latent adaptation:

$$\mathrm{SLA}(x) \triangleq x + f_0(x) + \sum_{k=1}^{K} \left[f_\tau \circ W \circ \varphi(x) \right]_k V_k(x), \tag{3.6}$$

where $[\cdot]_k$ picks the k'th element of the gating sequence. For an illustration of SLA, see Figure 3.5.

SLA gives rise to a differentiable dynamic architecture, which were studied in the context of reinforcement learning (Pham et al., 2018), Bayesian optimization (Kandasamy et al., 2018), or adapting to new tasks (Mallya et al., 2018; Rosenfeld and Tsotsos, 2018). While dynamic gates are subject to complex interactions such as negative transfer (Rosenbaum et al., 2019), ablations in Table 3.6 clearly show that taking a sparse perspective – which allows the model to mix either continuously or discretely – outperforms the alternative of a priori fixing either a smooth activation such as that used in attention mechanisms (Lin et al., 2017b), or discrete Gumbel-based sampling (Jang et al., 2016). In fact, this choice between discrete (Veit and Belongie, 2018) and continuous mechanisms (Shazeer et al., 2017; Sun et al., 2019a; Wang et al., 2019) delineates previous work that employed differentiable gates. Perhaps surprisingly given its successes in NLP settings (Deng et al., 2017; Peters et al., 2019), sparse activation has not been widely adopted in the computer vision literature.

It should be noted that a softmax-activated model can in principle also learn to suppress individual preactivation components by letting some q_k go to $-\infty$. This however requires either learning extra calibration parameters at every layer, defining a hard cutoff value (Shazeer et al., 2017) (thereby removing differentiability), or very large row-norms within the linear mapping W— a highly unlikely outcome given the several mechanisms found in state-of-the-art models (in particular weight decay, norm-penalties, or BN (Ioffe and Szegedy, 2015)) which act as direct counterforces to this.

3.4 Experiments

The proposed methods are evaluated on two latent domain problems constructed from Office-Home (Venkateswara et al., 2017) and PACS (Li et al., 2017) (examples shown in Figure 3.6 and 3.7). Also examined are a recently proposed fairness benchmark (Section 3.4.2), and long-tailed recognition benchmarks (Section 3.4.3).

3.4.1 Latent Domains

The first experiments use datasets with multiple domains: Office-Home and PACS. The main goal here is not to compare to existing multi-domain or domain adaptation methods that Office-Home or PACS were initially designed for, but to study two central research questions: whether learning separate model parameters is always the preferred option in multi-domain problems, and to what extent methods like SLA can benefit performance when learning multi-domain representations without domain labels.

Once again note domain labels are never used to train latent domain models, and they are only used for analyzing their performance in terms of the domain-normalized metrics introduced in Section 3.2.2.

Figure 3.6: Samples from Office-Home with equivalent class $y_d = y_{d'} \in \mathcal{Y}$ from four different domains. Latent domain learning aims to learn a unified model over these visually different depictions of the same object, but without access to the corresponding domain annotations, which are considered latent.

Optimization

All experiments use a ResNet model (He et al., 2016) pretrained on ImageNet. When SLA is used to replace the residual blocks, then only gates and corrections are learned, while the main convolution of the ResNet backbone remains fixed at its initial parameters, a strategy proposed by Rebuffi et al. (2017) that implicitly regularizes models.

All architectures are trained in exactly the same way with the same default hyperparameters: for 120 epochs using SGD (momentum parameter of 0.9), batch size of 128, weight decay of 10^{-4}, and an initial learning rate of 0.1 (reduced by 1/10 at epochs 80, 100). Official splits are used for each dataset, and average accuracies are reported over five random initializations.

All experiments use standard augmentation techniques: random cropping and flipping, as well as normalization. Prelearnt LDE (for additional details see Section 3.3.1) is applied with $M/C = 2$, preservation strength $\alpha = 0.5$, and randomly to 25% of examples.

Results for different clustering strategies for LDE are reported in Table 3.5. Increasing the number of corrections K within SLA consistently increases performance, however $K = 2$ already represents a strong boost from the baseline of having no sparse adapters, see results in Table 3.2.

	Type	A	C	P	R	OAcc	UAcc
Proportion π_d		*15.57*	*28.01*	*28.48*	*27.95*		
RA (Rebuffi et al., 2018)	MD	48.05	76.12	80.74	67.78	70.73	68.17
Domain-Adv. (Ganin et al., 2016)	MD	55.14	72.85	81.98	68.81	71.57	69.70
4×ResNet26	MD	52.47	79.95	85.02	70.01	74.34	71.86
ResNet26	LD	50.10	76.80	78.83	63.36	69.47	67.27
ResNet56	LD	52.26	78.47	80.80	66.34	71.66	69.47
MN (Chapter 2)	LD	50.54	78.02	78.62	64.30	70.08	67.87
RA (Rebuffi et al., 2018)	LD	58.44	79.15	81.55	72.13	74.65	72.82
MLFN (Chang et al., 2018)	LD	50.72	78.81	81.36	64.56	71.18	68.86
MMLD (Matsuura and Harada, 2020)	LD	59.63	67.89	81.16	74.35	72.19	70.76
Ours ($K = 2$)	LD	63.37	**81.84**	84.85	74.83	77.87	76.22
Ours ($K = 3$)	LD	62.86	80.99	85.47	76.15	78.09	76.37
Ours ($K = 4$)	LD	63.48	80.53	**85.59**	76.32	78.14	76.48
Ours ($K = 5$)	LD	**64.09**	80.64	84.52	**77.81**	**78.38**	**76.77**

Table 3.2: Per-domain performance in percent on Office-Home. Multi-domain (MD) baselines train subsets of parameters for each domain, while for latent domains (LD) models are trained without domain labels so all parameters are shared. Results for SLA+LDE are shown for multiple values of K. Domain-level performances for (A)rt, (C)lipart, (P)roduct, and (R)eal world are reported alongside the (O)bserved (Acc)uracy, and (U)niform (Acc)uracy which summarize performance across all four domains. Best results in bold.

Office-Home

The underlying data contains a variety of objects classes (alarm clock, backpack, etc.) among four domains: *art*, *clipart*, *product*, and *real-world*. Some examples from this dataset with the same object class (chair) are shown in Figure 3.6.

Table 3.2 shows results for d-supervised multi-domain (MD) approaches: RA (Rebuffi et al., 2018), domain-adversarial learning (Ganin et al., 2016) and a baseline of 4×ResNet26, one for each domain. For latent domain (LD) baselines, a single ResNet26 is learned, this time as one joint model over all domains. Next, SLA+RLDE are coupled with the very same ResNet26.

Learning a single ResNet26 over latent domains with no access to d-labels significantly harms performance. This problem is not addressed by simply increasing the depth of the network: while accuracy improves slightly, a ResNet56 exhibits the same performance losses — in particular on the latent domains *product* (P) and *real-world* (R).

While residual adaptation (Rebuffi et al., 2018) was shown to work extremely well in many multi-domain scenarios, performance here is sub-par, regardless of whether it accesses d (in which case there is one V_d per domain) or not (single V). When using annotations, the drop in performance likely originates from having single linear modules V_d for each domain, enabling no native cross-domain sharing of parameters. When d is hidden on the other hand, the model is forced to share a single linear adaptation module V between all four hidden domains, without the flexible gating between them as in SLA.

Learning annotations through clustering and coupling this with domain-adversarial gradient reversal as in MMLD (Matsuura and Harada, 2020) increases performance relative to its d-annotated counterpart (Ganin et al., 2016). The increase is modest however, and in line with reports that enforcing domain-invariance on the gradient level negatively impacts models' abilities to discriminate between classes (Wang et al., 2020).

MN, introduced in Chapter 2, can also be applied to latent domain learning, as it does not require domain annotations. While performance is increased from the ResNet26 baseline, the increase is small when comparing it to dedicated modules for latent domains: SLA contains individual corrections which are more flexible than MN, and the results suggest that this additional capacity considerably improves the processing of hidden domains.

Another related baseline are multi-level factorization nets (MLFN, Chang et al., 2018) which build on ResNeXt (Xie et al., 2017) to define a latent-factor architecture that accounts for multimodality in data. Crucially where SLA is fine-grained and uses gates to modulate corrections at each layer, MLFN instead enables and disables multiple network blocks at once, allowing SLA to outperform it.

The methods proposed in this thesis benefit performance significantly and increase uniform accuracy (UAcc, c.f. Section 3.2.2) by 14.12% relative to ResNet26. Accuracy without LDE consistently drops around 2%, see the ablation in Table 3.3. For Office-Home random pairings as in RLDE were found to be the superior option, whereas when domains exhibit a higher amount of separation CLDE is better suited, as for PACS which is evaluated in the next section. Mixup (Zhang et al., 2018) and MixStyles (Zhou et al.,

Augmentation	None	mixup (Zhang et al., 2018)	MixStyles (Zhou et al., 2021)	RLDE	CLDE
Office-Home	74.35	72.04	74.52	**76.22**	76.06
PACS	93.84	92.43	93.78	94.42	**94.70**

Table 3.3: UAcc (in %) for the proposed RLDE and CLDE (see Section 3.3.1) for two random examples/clusters. Additional results for $M/C > 2$ in Table 3.5.

2021), two alternative augmentations that interpolate between samples, appear to not be equally well suited for latent domain learning. Ablations for the style-transfer based augmentation of Borlino et al. (2020), which can be viewed as a special case of LDE, are shown in Table 3.5.

PACS

The second experiment examines performance on the PACS dataset (Li et al., 2017). Crucially PACS domains (*art, cartoon, photo, sketch*) differ markedly from one another (c.f. examples in Figure 3.7), constituting a latent domain problem with more separable domains than in Office-Home.

Results in Table 3.4 show that the proposed methods improve over existing baselines, even for the more distinct domains found in PACS. The largest gains occur on smaller domains (e.g. *art*), where it can be observed that standard models suppress underrepresented parts of the distribution (see additional discussion on this aspect for long-tailed recognition benchmarks in Section 3.4.3). SLA again surpasses the accuracy of 4×ResNet26, while requiring a fraction of the total model parameters (~ 9.7 mil for $K = 5$ vs. ~ 24.8 mil). Performance continues to increase with larger K in SLA.

The performance increase from using a latent domain-adversarial approach (Matsuura and Harada, 2020) versus using domain-annotations (Ganin et al., 2016) confirms that learning domains alongside the rest of the network can be a better strategy than trusting in annotations. As before MN improves results, but its limited flexibility prevents performance gains beyond those of SLA.

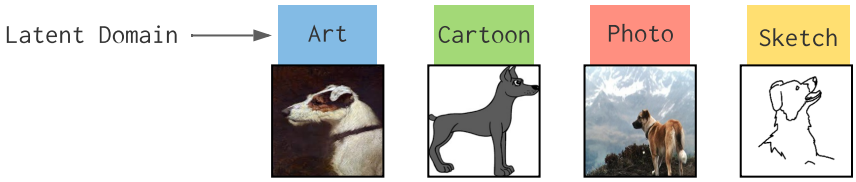


Figure 3.7: PACS samples with equivalent class (dog), but a different latent domain. Coloring of each domain corresponds to that in Table 3.4.

	Type	•A	•C	•P	•S	OAcc	UAcc
Proportion π_d		*0.205*	*0.235*	*0.167*	*0.393*		
RA (Rebuffi et al., 2018)	MD	85.14	92.05	94.50	94.30	91.93	91.50
Domain-Adv. (Ganin et al., 2016)	MD	86.47	93.25	94.17	91.30	91.25	91.30
4×ResNet26	MD	88.41	95.53	94.34	95.71	93.94	93.50
ResNet26	LD	85.27	94.55	93.85	94.98	92.70	92.16
ResNet56	LD	86.96	94.34	95.15	95.34	93.36	92.95
MN (Chapter 2)	LD	85.52	94.68	95.47	94.61	92.91	92.57
RA (Rebuffi et al., 2018)	LD	89.86	93.90	95.56	93.91	93.35	93.31
k-means+4xRA	LD	83.84	92.10	94.75	93.01	91.21	90.93
MLFN (Chang et al., 2018)	LD	78.38	91.29	88.19	92.95	88.78	87.70
MMLD (Matsuura and Harada, 2020)	LD	89.93	92.26	96.25	94.34	93.20	93.27
Ours ($K = 2$)	LD	91.67	96.08	95.95	95.10	94.77	94.70
Ours ($K = 3$)	LD	90.94	95.21	**97.25**	**95.59**	94.82	94.75
Ours ($K = 4$)	LD	90.46	96.19	97.09	94.98	94.69	94.68
Ours ($K = 5$)	LD	**92.87**	**96.62**	96.28	95.28	**95.27**	**95.26**

Table 3.4: Accuracy in percent for individual PACS domains (A)rt painting, (C)artoon, (P)hoto, and (S)ketch, and across domains in terms of UAcc and OAcc. Best overall performance underlined, best latent domain performance bold.

An important baseline to compare SLA against consists of a two-stage approach, whereby examples are first clustered into domains, and subsequently classified. Here k-means (using $D = 4$ centers, applied to the embeddings of images in the final layer of a pretrained ResNet26 feature extractor) is used with subsequent finetuning of residual adapters. At test time, examples from domains are first assigned to a cluster, and then classified with the associated residual adapter.

Results for k-means+4xRA in Table 3.4 show that a two-stage strategy is suboptimal. Similar to domain-supervised RA that uses g_d in Φ of eq. (3.4), this likely results from clustering assigning fixed switches that get used across all residual adaptations of the

Clustering	RLDE					CLDE-k-means				CLDE-GMM			
M/*C*	*1*	*2*	*4*	*6*	*8*	*2*	*4*	*6*	*8*	*2*	*4*	*6*	*8*
Office-Home	76.15	76.22	76.33	76.21	76.15	76.06	76.16	75.43	75.07	75.85	75.30	75.11	74.82
PACS	94.19	94.42	94.62	94.61	94.55	94.70	94.77	94.61	94.55	93.32	93.95	91.84	91.25

Table 3.5: An ablation showing UAcc in percent for Random-LDE and Cluster-LDE for different numbers of random examples M, cluster numbers C, and clustering methods. PACS seems to benefit from a clustering-based augmentation. Note RLDE with $M = 1$ corresponds to Borlino et al. (2020).

Gating mechanism			
SLA	Smooth (Lin et al., 2017b)	Discrete (Jang et al., 2016)	UAcc
✓			76.22
	✓		75.74
		✓	75.21
		`No LDE` »	74.35
		`No SLA` »	72.82

Table 3.6: UAcc in percent on Office-Home for SLA with alternative, non-sparse gating mechanisms. $K = 2$ is fixed across all trials shown here.

model. This is in conflict with the observation that for different visual characteristics, different layers are more relevant than others (Yosinski et al., 2014; Zeiler and Fergus, 2014). Opposed to this, SLA can flexibly share or separate features individually at every layer (c.f. qualitative results in Figure 3.9), synergizing only where appropriate across the depth of the model.

Ablation

Ablations in Table 3.5 show that LDE performs robustly under two modifications: (i.) when using $M > 2$ to increase the number of sources $x_{j_1}, \ldots, x_{j_M}$ whose style is mapped to the target x_n, and (ii.) when coupling CLDE with different clustering mechanisms. When removing LDE altogether performance drops to 74.85% UAcc, and when also withholding SLA this reduces to 72.82% UAcc of the ResNet26 backbone.

Table 3.6 shows that replacing sparse gating within SLA with either smooth or discrete gates registers a drop in performance. Accuracies for soft and straight-through Gumbel-

	ResNet26	RA (Rebuffi et al., 2018)	SLA
CIFAR-10	95.20	95.80	**96.32**
CIFAR-100	77.85	81.01	**82.18**

Table 3.7: Accuracies (in %) for ResNet26, RA, and SLA on single datasets. While these are not typically associated with latent domains, accounting for them raises performance relative to baselines.

softmax sampling (Jang et al., 2016) were on par, and the reported results are for straight-through sampling. In an additional ablation the residual backbone f_0 was not fixed, and instead its parameters were updated throughout model training. In line with what Rebuffi et al. (2017) reported, this lead to overfitting and performance dropped from 76.22% to 73.53% UAcc.

Single Datasets

Because SLA requires no domain annotations it can be used for learning over distributions of single datasets. Table 3.7 contains test accuracies on CIFAR-10 and CIFAR-100 for SLA ($K = 2$). This is compared to standard finetuning of the backbone, and RA (Rebuffi et al., 2018). SLA can be inserted seamlessly into the model, and no changes are made to the optimization settings used in previous sections.

Sparse adaptation outperforms traditional finetuning and RA on both datasets, showing that SLA can be used as a general-purpose module to increase performance on standard benchmarks. Note the larger performance gap on CIFAR-100. In all likelihood SLA has a special advantage there, as CIFAR-100 contains many small modes, which can be associated with latent domains.

Memory Requirements

Every layer contains a total of $O(K|C| + K|C|^2)$ parameters to parametrize the gate g and corrections V_k, respectively. This is however a very modest requirement, in particular

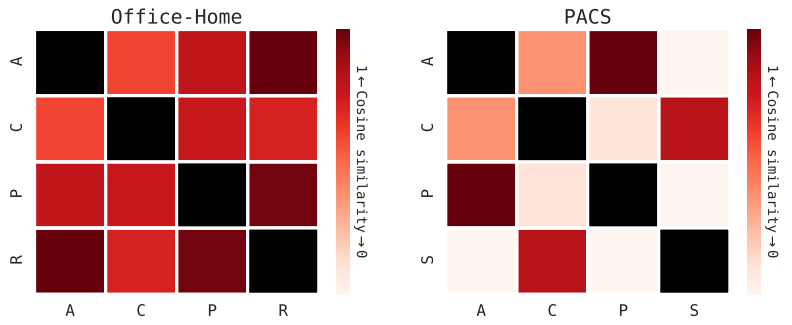


Figure 3.8: Cosine similarity between SLA gates for Office-Home (left) and PACS (right). PACS domains are more dissimilar, but similarities exist, e.g. (A)rt and (P)hoto.

because f_0 stays fixed: while a ResNet26 contains ~ 6.2 mil learnable parameters, even when setting $K = 5$ within SLA this sums to just ~ 3.5 mil free parameters.

Note also that the complexity of solving sparse gates in SLA scales as $O(K \log K)$, a negligible increase given the small K required.

Qualitative Analysis

This section analyzes global gating statistics of Office-Home and PACS domains, as well as feature sharing across different layers of a network that includes SLA, and how sparsity is utilized by the SLA gating mechanism. Moreover, an analysis of the final representation of images in networks that include SLA is presented, alongside evidence that sharing between geometric properties (shape, pose, etc.) occurs in the gates. Here $K = 2$ is fixed to simplify the analysis.

First, Figure 3.8 shows average cosine similarities of per-domain gating vectors $g \in \mathcal{G}^L$ across $l = 1, \ldots, L$ layers of ResNet26. This confirms that Office-Home domains differ less than the domains found in PACS.

Figure 3.9 presents layerwise measurements of $\mathrm{Corr}[g_l(x), g_l(x')]$ for x, x' drawn from

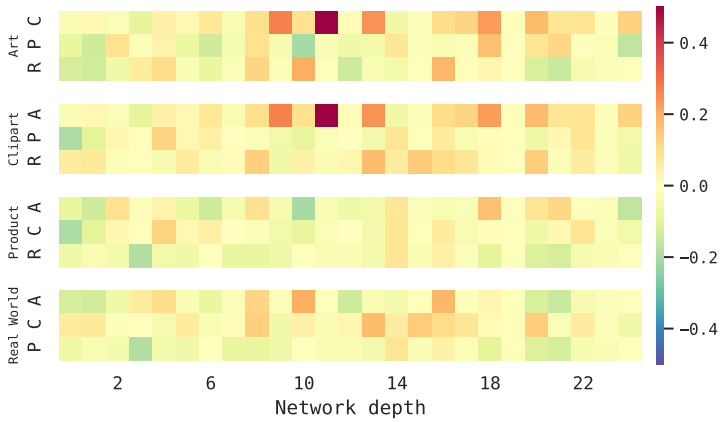


Figure 3.9: Layerwise correlation between intermediate feature representations of SLA convolutions V_k on Office-Home for different ground-truth domains. Most correlations occur in the mid-to-late stages of the model.

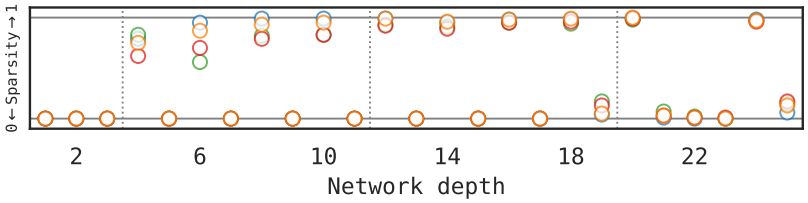


Figure 3.10: SLA sparsity (Office-Home); dotted lines indicate pooling transitions between residual blocks.

differing $d \neq d'$ for Office-Home. If the correlation between domains is high, then similar corrections V_k are responsible for processing samples from a particular combination of domains, indicating a large amount of feature sharing.

Across top layers of the network there is little correlation, presumably as low-level information associated with each domain is processed independently. In the mid to bottom stages correlation increases: these layers are typically associated with higher-order features (Yosinski et al., 2014; Mahendran and Vedaldi, 2016; Asano et al., 2020), and since label spaces are shared between latent domains, similar object-level features are required to classify objects into their respective categories.

The sparse gates used in SLA (c.f. eq. 3.6) have the flexibility to output single activations

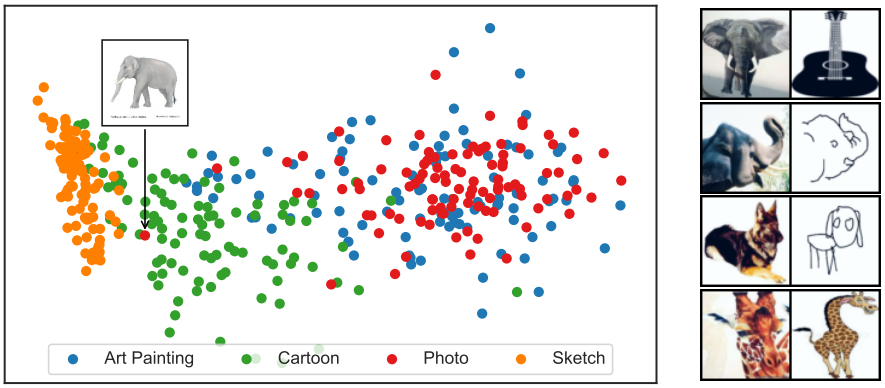


Figure 3.11: **Left**: PCA of samples represented by their SLA gate activations, colored by their ground-truth domain label as assigned in PACS. SLA shares parameters between visually similar domains *art* and *photo* (•,•), while isolating *sketch* (•). The arrow highlights one sample that has been labeled a *photo* in PACS. SLA categorizes it as a *cartoon* instead, a more adequate assignment for this particular image. **Right**: sample pairs from different domains ($d_i \neq d_j$) with matching SLA activations. Note their similar geometric properties (e.g. pose).

(i.e. become fully discrete) or output only non-zero values (a continuous gate). To evaluate the gating behavior, the per-layer sparsity $\mathbb{E}_{x\sim\mathbb{P}_d}[K - \|g_l(x)\|_0]/(K-1)$ can be analyzed, where $\|\cdot\|_0$ counts values that differ from zero.

Figure 3.10 shows that the sparsity of SLA gates varies across model depth. Interestingly after each downsampling operation SLA tends to be relatively sparse, followed by a dense gate, then again a sparse one, and so on. The model thus clearly utilizes the flexibility from sparse gating functions.

Due to PACS domains being relatively distinctive, the dataset is an interesting candidate for additional analysis of how sparse adaptation accounts for different ground-truth domains. Figure 3.11 (left) shows activations for the first gate (collected at all layers) for samples from all four PACS domains, visualized by their principal components.

In SLA visually similar domains *art* and *photo* (•,•) cluster together. The manifold describing *sketches* (•) is arguably more primitive than those of the other domains, and indeed only maps to a small region. *Cartoon* (•) lies somewhere between sketches and

real-world images. This matches intuition: a cartoon is, more or less, a colored sketch.

One image is displayed (highlighted with an arrow) that shows an elephant which SLA places among the *cartoon* domain. This image was however assigned a ground-truth domain label of *photo* in the PACS dataset. The ground-truth label appears to have been assigned in error, but different from approaches that use d-supervision, SLA learns latent domains on-the-fly and is therefore not irritated by this.

Figure 3.11 (right) displays pairs of samples that have similar gate activations across the network, but are from different domains. Pose, color, etc. of the samples are visibly related. Compare in particular the poses of elephants (second row). This similarity indicates that SLA does not only account for latent domains, but also incorporates geometric features into its gating mechanism.

3.4.2 Fairness

From the perspective of algorithmic fairness, a desirable model property is to ensure consistent predictive equality across different identifiable subgroups in data (Zemel et al., 2013; Hardt et al., 2016; Fish et al., 2016). This relates to one of the goals in latent domain learning: to limit implicit model bias towards large domains, and improve robustness on small domains.

Recent work elevated the role of small subgroups in data and examined model fairness on CelebA (Bagdasaryan et al., 2019; Wang et al., 2020; Hooker et al., 2020). Because such subgroups may be interpreted as constituting an individual component $\mathbb{P}_d$, they are an interesting candidate for the evaluation of latent domain models in an applied setting.

This section evaluates a benchmark that contains face images with different attribute labels (e.g. “brown hair”, “glasses”), constructed from the Aligned&Cropped subset of CelebA (Liu et al., 2015) by hiding gender information (Wang et al., 2020). Models are evaluated on all 39 remaining attributes, which experience varying amounts of gender

	ResNet18	ResNet18-SLA	ResNet34	ResNet34-SLA	ResNet50	ResNet50-SLA
mAP (↑)	0.718	0.732 *(+0.015)*	0.713	0.740 *(+0.027)*	0.745	0.750 *(+0.005)*
BA (↓)	0.025	0.014	0.022	0.009	0.012	0.008

Table 3.8: mAP (measured in percent) and bias amplification of SLA on the CelebA fair attribute recognition benchmark (Wang et al., 2020).

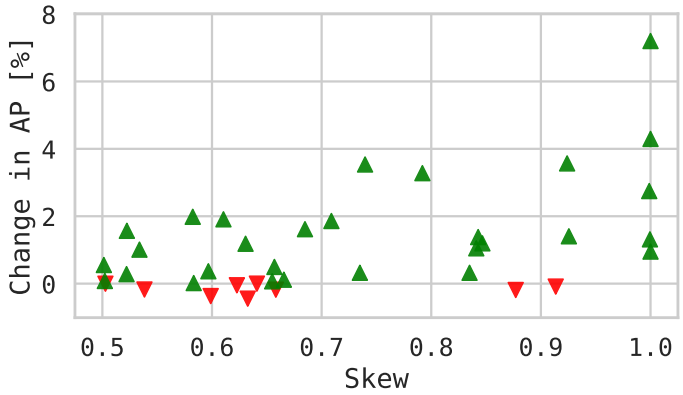


Figure 3.12: Change in AP (in percent) between ResNet18 and ResNet18-SLA for different gender skews in CelebA attributes.

skew. Framed as a latent domain problem one could identify $d = \{\text{female}, \text{male}\}$, but models have no access to this spurious information.

The same optimization settings are used as previously (see Section 3.4.1) to finetune models for 70 epochs with learning rate reductions at epochs 30, 40, and 50. This setup closely follows previous work on empirical fairness (Wang et al., 2020; Ramaswamy et al., 2020), which however – different from the methods presented here – focused on learning models that have access to the gender-attribute d.

Per-attribute accuracy is evaluated using mean average precision (mAP) alongside bias amplification (BA) (Zhao et al., 2017) (see Table 3.8). The latter compares the propensity of a model to make positive predictions (i.e. f exceeds some threshold $t_+ \in [0, 1]$) in the gender g_y^* that appears most frequent within attribute y, compared to the true counted

ratio of positive examples y_+:

$$\text{BA}[f] = \mathbb{E}_{x\sim\mathbb{P}_x}\left[\frac{\mathbb{1}_{f(x)>t_+|g^*_y}}{\mathbb{1}_{f(x)>t_+}}\right] - \mathbb{E}_{y\sim\mathbb{P}_y}\left[\frac{\mathbb{1}_{y=y_+|g^*_y}}{\mathbb{1}_{y=y_+}}\right], \tag{3.7}$$

where t_+ is optimized for on the validation split. For example if 60% of male examples are wearing glasses but under the model this is raised to a total of 65%, then bias is amplified by BA = 0.05.

Performance is compared between ResNet18, ResNet34, and ResNet50, and the same models with SLA inserted ($K = 2$). SLA consistently raises both mAP and reduces bias, indicating that it relies less on spurious correlations between protected attributes and target properties in data to formulate its predictions. In addition, Figure 3.12 compares per-attribute skew towards either female or male (whichever is more frequent) to the gain in performance from ResNet18 to the same model but with SLA inserted. A clear trend is observed here, whereby SLA is able to raise performance the most in those attributes that experience the largest amounts of skew.

3.4.3 Long-Tailed Recognition

Standard models often experience difficulty when some classes are heavily underrepresented. This problem has recently been studied in long-tailed recognition (Liu et al., 2019b; Cao et al., 2019) with benchmarks that modify CIFAR-10 and CIFAR-100 to an imbalanced version by reducing the number of examples for some classes, e.g. 6-10 for CIFAR-10 (Buda et al., 2018). The severity of the imbalance is usually described via the ratio $\rho = n_{\max}/n_{\min}$ between the largest and smallest classes.

Similar to the fairness experiments (Section 3.4.2), long-tailed distributions may be viewed as containing an underrepresented latent component with $\pi_d = 1/(1+\rho)$. Previous results, in particular for PACS (c.f. Section 3.4.1), that fortified small latent domains within $\mathbb{P}$ therefore serve as motivation to evaluate the imbalance setting more closely for latent domain methods.

Imbalanced CIFAR-10						
ρ	ERM	ERM-SLA	Focal	Focal-SLA	LDAM-DRW	LDAM-DRW-SLA
10	86.09	**93.05** *(+6.96)*	86.61	92.14 *(+5.53)*	91.08	92.49 *(+1.41)*
100	68.02	**81.60** *(+13.58)*	67.44	78.82 *(+11.38)*	75.67	80.96 *(+5.29)*
Imbalanced CIFAR-100						
ρ	ERM	ERM-SLA	Focal	Focal-SLA	LDAM-DRW	LDAM-DRW-SLA
10	65.16	**70.76** *(+5.60)*	64.56	70.60 *(+6.04)*	66.08	69.61 *(+3.53)*
100	45.44	48.46 *(+3.02)*	45.19	48.39 *(+3.20)*	51.25	**55.06** *(+3.81)*

Table 3.9: Test accuracy (in %) on imbalanced CIFAR benchmarks (Buda et al., 2018). SLA consistently improves performance for standard ERM and existing long-tail approaches, such as a focal (Lin et al., 2017a) or label distribution aware loss (Cao et al., 2019).

Since SLA is entirely model-based, it can be combined seamlessly with recent state-of-the-art techniques for long-tailed recognition which are loss-based: reducing contributions from well-classified examples as in focal losses (Lin et al., 2017a), or a label-distribution-aware margin loss with deferred reweighting (Cao et al., 2019). Adaptation via sparse gates is found to consistently improve performance of the underlying ResNet26 across different imbalance strategies on long-tail benchmarks (see Table 3.9).

This result highlights the benefit of relaxing the domain concept from its more strict definitions, e.g. found in multi-domain learning, to that of latent domains: SLA appears to have advantages even when one can only define *very* abstract domains as in this section, where d = "y in six to ten".

3.5 Conclusion

This chapter introduced and formalized the problem of learning over distributions that contain multiple latent domains, and showed that this poses a considerable challenge for standard learning methods.

The main motivation was to investigate whether learning separate model parameters

for each domain is always the preferred strategy for multi-domain problems, and what methods to use when domain labels are missing in data. While experiments showed that performance for standard models tends to degrade without domain labels, a new class of augmentations alongside a novel sparse adaptation strategy were proposed which together account for (and often exceed) this lost accuracy — benefiting several challenging problems where some notion of a domain (but no annotation) exists.

Several interesting questions remain: for one, in what circumstances is the use of domain labels preferred, and how can practitioners decide whether explicit domain partitions are sensible. And how can one incorporate partial labels into multi-domain methods? These research questions are touched upon in additional detail in Chapter 5, which concludes this thesis with a discussion of directions for future work.

As this and the previous chapter highlighted, learning problems that contain heterogeneous sources or latent domains benefit from customized solutions. The next chapter investigates the appearance of new modes in data in the context of anomaly detection, and introduces new transfer-based techniques for this setting, in particular ones that counter catastrophic forgetting (c.f. Section 3.1.5).

Chapter 4

Transfer-Based Semantic Anomaly Detection

The detection of anomalies is challenging due to the countless ways in which these may appear in high-dimensional data. Existing methods, reviewed in Section 4.1, focus on enhancing the robustness of networks, for example through self-supervision (Golan and El-Yaniv, 2018; Hendrycks et al., 2019c). While such strategies may be sufficient for modeling simplistic visual anomalies (such as watermarks, or salt and pepper noise), there is no good known way of preparing models for all potential and unseen anomalies that can occur, such as the appearance of new semantic categories in data.

This chapter introduces approaches that improve the detection of visual anomalies using transfer learning. Section 4.2 explains how anomaly detection can benefit from transferring over representations from some large and varied semantic task, enabling the formulation of new transfer-based anomaly detection methods in Section 4.3.

The experiments in Section 4.4 show that transfer-based strategies yield very powerful methods that can be coupled with any modern network architecture, and outperform

previous approaches in the anomaly detection literature on a set of common benchmarks.

Anomalous instances can in principle also originate from previously unseen regions of the input space that can be associated with anomalous modes (e.g. sketches, when previously there were only photos in the training data). While Section 4.4.4 experimentally evaluates this problem, it should be noted that the stark visual differences that characterize visual domains in existing benchmarks such as PACS makes their detection relatively straightforward, such that they may often be detected by hand-crafted features or shallow learning strategies. Due to this, such simpler AD problems have recently been summarized under the term "non-semantic" by Ahmed and Courville (2020). The main focus of this chapter lies on the detection of more high-level (or "semantic") anomalies, in particular the appearance of new object classes, as opposed to low-level anomalies, e.g. texture defects (Bergmann et al., 2019).

4.1 Background

Given a collection of images, it is often interesting to automatically determine which examples in it are representative, or vice versa, which of them are unusual. This fundamental problem in machine learning is usually referred to as outlier, novelty, or anomaly detection (AD), with applications ranging from medicine (Wong et al., 2003; Schlegl et al., 2017), to fault detection (Campbell and Bennett, 2001; Görnitz et al., 2015), and astronomy (Dutta et al., 2007; Collins et al., 2018).

Regardless of the type of anomaly that is to be detected, AD always begins by incorporating a set of non-anomalous examples $S_n \triangleq \{x_i\}_{i=1,\ldots,n} \in \mathcal{X}$ into a model of normality.[3] These examples are assumed to have been sampled from the distribution of the normal data $\mathbb{P}^+$, with associated density p^+. The goal is to use the examples in S_n to learn a one-class model $f_\theta : \mathcal{X} \to [0, 1]$ with parameters $\theta \in \Theta$ that decides whether a previously unseen $x \in \mathcal{X}$ is likely *normal* (s.t. $f_\theta(x)$ assumes small values)

[3]Previous chapters used n to index data. Here this denotes the total number of examples in S_n.

or *anomalous* (large score $f_\theta(x)$). For example, a model could be trained on images of cats. During evaluation, the model should be able to score all cats as normal, while other objects, e.g. dogs, deer, etc., are assigned high scores, i.e. deemed as anomalous. Note that different from classification problems, in AD models do not need to classify examples into their respective object classes, but only score how likely they contain an anomaly. In the context of AD the more general case of a multimodal normal class $\mathbb{P}^+$ is called "semantic AD" (Ahmed and Courville, 2020), and is evaluated in Section 4.4.2.

4.1.1 Traditional AD

Early research on AD goes back as far as Edgeworth (1887). In its long history, AD has been incorporated with many different methodological approaches: generative models, for example, can be applied to detect anomaly via thresholding of the learned model. Given examples, one simply estimates $p_\theta \approx p^+$, and declares anomalies when $p_\theta(x) < \tau$ for some $\tau \in (0, 1)$.

Similar approaches that use thresholding can be used for non-parametric methods, for example kernel density estimation was used for detecting intrusions in Yeung and Chow (2002), mixtures of Gaussians in Pelleg and Moore (2004), and hidden Markov models in Ourston et al. (2003). All these methods fall into the spectrum of traditional AD methods, an overview over which can be found in Chandola et al. (2009) and Emmott et al. (2013).

Another popular traditional AD method are one-class support vector machines (OCSVM, Schölkopf et al., 1999). These were adopted in early work that introduced deep learning into AD (or "deep AD", for short), consisting of a two-stage setup that freezes features after learning representations via an autoencoding setup, and subsequent application of OCSVMs to detect anomalous examples (Erfani et al., 2016). Since then, various approaches for deep AD have been proposed, which are summarized in the next section.

4.1.2 Deep AD

With the advent of deep learning, the focus of AD methods has started to shift. Ruff et al. (2018) showed that deep end-to-end representation learning of traditional AD losses is complicated by a phenomenon they termed "hypersphere collapse", whereby all points in S_n tend to get mapped close to zero. This can occur when combing traditional AD objectives, e.g. the loss associated with support vector data descriptions (Tax and Duin, 2004), with a large hypothesis class (in particular deep models). Ruff et al. (2018) addressed this by restricting the network architecture on the level of functional layers, and in particular for image data recommend including no bias terms in convolutions.

Subsequent works in deep AD utilized a wide range of methods, such as scoring anomalies by their reconstruction loss in autoencoder setups (Zhou and Paffenroth, 2017; Zong et al., 2018), further modifications to one-class losses that improve the model's robustness against hypersphere collapse (Sabokrou et al., 2018; Ghafoori and Leckie, 2020; Goyal et al., 2020), or scoring new examples depending on whether they can be generated by a GAN solely trained on examples from $\mathbb{P}^+$ (Goodfellow et al., 2014; Schlegl et al., 2017; Akcay et al., 2018; Deecke et al., 2018; Perera et al., 2019; Ngo et al., 2019; Berg et al., 2020).

A learning setting closely related to AD is out-of-distribution detection (Hendrycks and Gimpel, 2017; Hendrycks et al., 2020), which is used to investigate how models can robustly detect examples from previously unseen datasets, for example using invertible networks (Schirrmeister et al., 2020) or by estimating uncertainty (Burda et al., 2019; Ciosek et al., 2020).

4.1.3 Self-Supervised AD

A recent focus has been on developing auxiliary tasks from the samples in S_n to learn better representations for AD, often following the paradigm of self-supervision. For this,

Golan and El-Yaniv (2018) a priori define a sequence of simple geometric transformations $t_1, \ldots, t_{K-1}$, e.g. flipping of images or rotation. A new dataset is created by applying each transformation to every image in S_n, such that $\tilde{S}_n = \{(t_k(x), k) \,|\, x \in S_n, k = 1, \ldots, K\}$, with t_K the identity mapping.

Next a classifier $h_\theta \colon \mathcal{X} \to [0,1]^K$ is learned over $\tilde{S}_n$, to learn geometrical features for AD. Empirical results, in particular of Hendrycks et al. (2019a), demonstrate that selecting anomaly scores from such features via the maximum softmax probability (Hendrycks and Gimpel, 2017) as $1 - \max_k h_\theta(x)_k$ results in favorable performance.

While these and related (Bergman and Hoshen, 2020; Tack et al., 2020; Sohn et al., 2021) self-supervised methods use only examples from S_n to learn representations, more recent works in AD have increased performance through the concept of outlier exposure, a form weak supervision. This is introduced in the next section.

4.1.4 Weakly-Supervised AD

Hendrycks et al. (2019b) proposed to enrich AD representations through the concept of outlier exposure (OE), in which the normal class is differentiated against a large unstructured set of image data, which serve as auxiliary outliers. For this, all normal examples receive a negative labeling, i.e. $y = 0, \forall x \in S_n$, and are classified against a second set Q_m that contains positively labeled (i.e. $y = 1$) examples of all sorts of objects. For example, Q_m could consist of all images contained in ImageNet (Deng et al., 2009). A model is then learned via binary classification of the negative versus the positive set:

$$\arg\min_{\theta \in \Theta} \Big\{ \mathcal{L}_{S_n}[f_\theta] + \mathcal{L}_{Q_m}[f_\theta] = \frac{1}{|S_n|+|Q_m|} \Big[\sum_{x \in S_n} \log f_\theta(x) + \sum_{x \in Q_m} \log(1 - f_\theta(x)) \Big] \Big\}, \tag{4.1}$$

At test time f_θ can be applied directly to examples to obtain an anomaly score. Importantly, this amounts to a form of weak supervision via existing resources (Zhou, 2018), and is not equivalent to supervised classification: images from the auxiliary

corpus Q_m are not necessarily true anomalies, and may even contain samples from $\mathbb{P}^+$.

The concept of OE has been quickly adopted in recent AD literature (Hendrycks et al., 2019c; Ruff et al., 2020a; Liznerski et al., 2021), and while the approaches presented in this chapter also leverage large corpora as in OE, they establish inductive biases as a separate crucial element for AD.

4.2 Motivation

For data types that are semantically rich such as images, "unusualness" can be caused by a variety of high-level (or *semantic*) factors, for example the appearance of new objects classes, or unexpected shapes or poses. Because of the large number of factors that can potentially cause an anomaly, there exists no established principal learning objective for deep AD.

While auxiliary methods that use self-supervision (Section 4.1.3) or weak supervision (Section 4.1.4) exist, the relatively ad-hoc nature of these approaches – especially given the semantic richness present in natural images – make it questionable whether one can learn particularly meaningful features from such auxiliary objectives. This is problematic since anomalies can manifest themselves in ways that require a good semantic understanding, for example when anomalies appear in crowded scenes (Mahadevan et al., 2010).

Here a different perspective is proposed. Because it can be difficult to anticipate all potential anomalies from the normal data alone, an alternative is to follow a transfer-based approach that utilizes the semantically rich features obtained from some semantic task associated with a large, varied dataset.

As discussed in Section 3.1, the transfer from rich semantic representations has been shown to boost the performance in many machine learning problems, including image classification Donahue et al. (2014); Guo et al. (2019a), object detection (Girshick et al.,

2014; Girshick, 2015), transferring between large numbers of tasks (Zamir et al., 2018), or from one domain to another (Rebuffi et al., 2017, 2018). In another line of work, a surge of papers has recently elevated the role of pretrained models in natural language processing (Mikolov et al., 2018; Howard and Ruder, 2018; Devlin et al., 2019; Adhikari et al., 2019; Hendrycks et al., 2020).

Transfer-based AD builds on the increased availability and utilization of networks pretrained on semantically rich tasks that incorporate different variations commonly seen in data (edges, color, semantic categories, etc.). The central hypothesis in transfer-based AD is that transferring features from such semantic tasks, for example ILSVRC image classification (Deng et al., 2009), provides very powerful and generic representations for various AD problems, even when the pretraining task is only loosely related to the task of AD.

For AD in particular, it is crucial to preserve variations incorporated during pretraining that, even though they potentially don't exist in the set of normal data S_n, can nonetheless be meaningful for inferring anomalous semantics at test time (Tax and Müller, 2003; Rippel et al., 2020). This requires ensuring that the change in representation from the pretraining task is not excessive, which risks *catastrophic forgetting* (Kirkpatrick et al., 2017) of features relevant to assessing such anomalous variations in unseen examples.

Note however that opposed to mere feature extraction (Bergman et al., 2020), experiments (see Section 4.4) show that it is critical to let the network have *some* flexibility to learn new variations important for AD. Before introducing adequate regularization strategies for the transfer of parameters in Section 4.3, the next section compares different AD paradigms through linear probes.

4.2.1 Linear Probes

To motivate transfer-based AD, the semantic viability of features learned under different auxiliary AD objectives is evaluated in this section. Linear probes (Zhang et al., 2017b)

are employed over a so-called two-stage setup commonly used in AD applications (Erfani et al., 2016; Sohn et al., 2021): after an initial feature extraction phase f over some normal data $x \sim p^+(x)$, a subsequent one-class SVM (Schölkopf et al., 1999) learns to encapsulate the extracted features of the data (associated with one single class) in S_n.

For the experiments in this section a standard AD benchmark (Ruff et al., 2018) is used: single classes from CIFAR-10 (e.g. dogs) constitute the normal class, and the one-class model is learned over all embeddings of the training examples of this class. At test time, it is measured whether the two-stage model can successfully identify the appearance of the remaining object classes (cats, deer, etc.) as anomalous.

The metric reported in Table 4.1 results from repeating this procedure for all ten classes, and recording the area under the ROC curve (AUC) relative to that of a random baseline (AUC of 0.5). Note an important benefit of using AUC is that it does not require selecting a threshold for scoring examples, as the AUC is computed by varying the decision threshold across all possible values in $[0, 1]$, plotting recall as a function of the false positive rate, and integrating over the resulting area.

The initial extraction phase occurs at one of three layers (`conv1-3`) of a LeNet architecture (LeCun et al., 1989) trained via three different paradigms (see next paragraph). The subsequent one-class stage always uses the exact same one-class SVM to assign anomaly scores. Fixing a simple model on top of $f_{\texttt{conv},i}(x)$ allows direct insights into the viability of each layer's features for semantic AD.

The comparison is carried out on top of $f_{\texttt{conv},i}$ after training the extraction model with three different learning paradigms for AD:

(i) self-supervision through geometric transformation as in Hendrycks et al. (2019c);

(ii) weakly supervised classification via outlier exposure (CIFAR-100 as OE dataset) (Hendrycks et al., 2019b);

(iii) transferring from another task (CIFAR-100 classification) and subsequent finetuning through OE.

Layer	(i) Self-sup.	(ii) Weakly-sup.	(iii) Transfer-based
$f_{\texttt{conv},1}$	1.44 *(0.19)*	1.58 *(0.20)*	2.02 *(0.21)*
$f_{\texttt{conv},2}$	4.60 *(0.17)*	3.83 *(0.16)*	5.48 *(0.19)*
$f_{\texttt{conv},3}$	4.63 *(0.15)*	5.12 *(0.12)*	6.72 *(0.15)*

Table 4.1: Percent improvement in AUC relative to a random baseline on CIFAR-10 AD for one-class SVMs on top of features extracted from LeNet layers (`conv1-3`) trained through different learning paradigms (self-supervised, weakly-supervised, transfer-based). Standard deviations in parentheses were computed over five random seeds.

For (i), (ii), and (iii), deeper features always result in performance improvements (see Table 4.1). When extracting at deeper layers – which are typically associated with higher semantic function (Yosinski et al., 2014; Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2016) – there is however a performance gap between paradigms: (i) self-supervised features do not improve from `conv2` to `conv3`, indicating they learn predominantly low-level features, an observation also made by Asano et al. (2020). For (ii) OE-based extraction performance increases a little at every layer, but overall AUCs are most improved by the (iii) transfer-based approach, which raises mean AUC by 6.72% in `conv3`. From this, it can already be observed that transfer-based features can have a favorable impact for robust downstream AD detection performance.

The experiments in Section 4.4 expand this finding and show that, when transferring semantic representations to complex AD tasks, it is crucial to ensure models do not suffer from catastrophic forgetting. The next section introduces ways in which this can be achieved, e.g. through adequate regularization.

4.3 Methods

Components of the proposed transfer-based methods are reviewed in Section 4.3.1, with subsequent introduction of two new methods for semantic AD with an inductive bias: ADIB (Section 4.3.2) and ADRA (Section 4.3.3).

4.3.1 Transfer-Based AD

Following work that investigated the prospects of large pretrained networks (Zamir et al., 2018; Adhikari et al., 2019; He et al., 2019; Hendrycks et al., 2019a, 2020), a recent study proposed carrying out AD through a nearest neighbor search on top of features extracted from a large pretrained residual network (Bergman et al., 2020).

However as can be seen from the experimental results in Table 4.5, simply transferring over fixed representations to an unrelated task seems subpar for semantic AD. Next, it is outlined how parameters obtained from pretraining can be more effectively transferred to the problem of AD.

Pretraining itself follows a simple protocol (c.f. Section 3.1.1): a model's parameters are randomly initialized with some distribution, for example Xavier initialization (Glorot and Bengio, 2010). Optimization of a suitable transfer task $\mathcal{T}$ (e.g. ImageNet object classification) yields a set of general-purpose parameters $\theta_0 \in \Theta$. The so-obtained model f_{θ_0} is then ready to be transferred to some downstream task $\mathcal{B}$.

The traditional methodology for leveraging pretrained models is to continue to optimize the model parameters (or a subset thereof) on $\mathcal{B}$. One crucial limitation of this learning protocol is that when learning on $\mathcal{B}$ isn't carried out carefully through the introduction of some explicit inductive bias (Li et al., 2018c), this risks *catastrophic forgetting* of information previously extracted from $\mathcal{T}$.

To alleviate this issue, a common approach is to use regularization, e.g. as in continual learning (Kirkpatrick et al., 2017; Lopez-Paz and Ranzato, 2017) (c.f. Section 3.1.5). The following two sections introduce two new AD-specific learning methods that prevent catastrophic forgetting in the context of AD.

4.3.2 Anomaly Detection with an Inductive Bias

Transfer-based AD hypothesizes that the best bet for robust semantic AD is to introduce an inductive bias into models. To achieve this, the weakly supervised learning criterion in eq. (4.1) is augmented with an additional regularizer $\Omega\colon \Theta \times \Theta \to \mathbb{R}_+$ that depends on pretrained parameters $\theta_0 \in \Theta$, resulting in the following objective:

$$\arg\min_{\theta \in \Theta} \left\{ \mathcal{L}_{S_n}[f_\theta] + \mathcal{L}_{Q_m}[f_\theta] + \Omega(\theta, \theta_0) \right\}. \tag{4.2}$$

As the ablations in Table 4.4 show, an inductive bias such as L_2 regularization towards the initial pretrained parameters θ_0 is crucial for robust semantic AD performance. Motivated by this finding, in *Anomaly Detection with an Inductive Bias (ADIB)* the regularizer is set to $\Omega(\theta, \theta_0) = \alpha||\theta - \theta_0||^2$ scaled by $\alpha \in \mathbb{R}_+$.

Recent state-of-the-art AD methods have proposed to modify the objective in eq. (4.2) by using radial functions Ruff et al. (2020a), which is in line with the so-called concentration assumption common in AD (Schölkopf and Smola, 2002; Steinwart et al., 2005). Such radial functions are included in the ablations (see Table 4.4), however it is empirically observed that – when paired with an explicit inductive bias – standard classifiers typically perform better.

ADIB is found to outperform previous state-of-the-art AD methods on semantic anomaly benchmarks. For the CIFAR-10 semantic AD benchmark, for example, it raises the state of the art to 74.6 versus 41.6 mean AP (in percent) reported previously by Ahmed and Courville (2020). Moreover, ADIB sets a new state of the art on the widespread one-versus-rest AD benchmark, raising the bar from 96.1 (Ruff et al., 2020a) to 99.1 mean AUC (in percent).

4.3.3 Anomaly Detection with Residual Adaptation

Similar to how this was done for latent adaptation (Section 3.3.3), regularization can also be formulated to bolster parameter efficiency, but without the focus on multiple domains. For this, the transformation of each residual layer $\Phi(x) = x + f(x)$ is constrained to allow at most a linear change from some pretrained mapping Φ_0 with f_0, such that $\Phi(x) - \Phi_0(x) = V(x)$. This is then rearranged to:

$$\Phi(x) = x + f_0(x) + V(x), \tag{4.3}$$

where V linearly corrects from adjacent layers (via a single 1x1 convolution), and f_0 is the residual 3x3 convolution associated with the pretrained Φ_0. As in Chapter 3, only the parameters of the linear correction V are updated, while the pretrained f_0 is left unchanged (full details of the learning process and surrounding hyperparameter choices are in Section 4.4).

This strategy is applied in *Anomaly Detection with Residual Adaptation (ADRA)*. Fixing f_0 and only varying the parameters associated with $V(x)$ in the model makes ADRA comparatively parameter-efficient, and such savings are crucial for applications in which multiple normal datasets exist but memory footprints are restrictive, e.g. federated learning scenarios (Yang et al., 2019; Bhagoji et al., 2019). Notably, as the experiments in Section 4.4 demonstrate, the performance of ADRA is often comparable to that of regularizing all parameters via Ω as in ADIB.

4.4 Experiments

Section 4.4.1 proposes the use of disentanglement datasets (Gondal et al., 2019) to evaluate the semantic detection performance of AD models. The resulting experimental setup allows for a controlled comparison between AD paradigms under semantic intervention (e.g. a change of object color), showing that transfer-based AD preserves meaningful

variations in its representations when it is coupled with appropriate regularizations.

In addition to verifying the suitability of the proposed methods on semantic AD benchmarks in Section 4.4.2, models trained on semantic tasks have been shown to learn elements required for non-semantic decisions in early parts of the network (Zeiler and Fergus, 2014), and experiments in Sections 4.4.3 and 4.4.4 show that transfer-based AD methods are indeed suitable for non-semantic AD tasks.

4.4.1 Examining Models through Interventions

As previous authors have emphasized, curating datasets with semantic anomalies is challenging (Ahmed and Courville, 2020). For gaining better insights into AD methods here it is proposed to borrow from datasets developed in recent research on unsupervised learning of disentangled representations (Kulkarni et al., 2015; Higgins et al., 2017; Bouchacourt et al., 2018; Burgess et al., 2018; Chen et al., 2018; Kim and Mnih, 2018; Kumar et al., 2018; Locatello et al., 2019, 2020).

These disentanglement datasets, in particular high-resolution, realistic ones such as the recently released MPI3D (Gondal et al., 2019), contain underlying ground-truth factors of images. In contrast to previous benchmarks for semantic AD (see Section 4.4.2), for example those that modify CIFAR-10 to such a task (Ahmed and Courville, 2020), interventions on ground-truth factors allow for principled measurements of semantic capabilities of a model, as for example the color of an object can be changed in a systematic fashion.

MPI3D contains joint pairs of latent ground-truth factors z (color, shape, angle, etc.), and corresponding images x_z of a robot arm mounted with an object. The original dataset comes in three styles (photo-realistic, simple, or detailed animation); because the models evaluated use rich deep architectures, the evaluation on simple and animated images (which are useful for simpler models) is skipped, such that the focus lies on the photo-realistic images here.

All models use the same number of parameters, and differ only in which AD loss is optimized:

- DSVDD (Ruff et al., 2018) uses eq. (4.1) without any weak supervision (no $\mathcal{L}_{Q_m}[f_\theta]$ term).
- SAD (Ruff et al., 2020a,b) differs from DSVDD only in that it uses OE.
- ADIB and ADRA combine both OE and an inductive bias, see eq. (4.2).

To ensure a fair comparison, the exact same ResNet26 is used for all methods, and all of them are initialized in exactly the same way, i.e. with the same pretrained weights. Note however the proposed ADRA has less modeling power than DSVDD and SAD, due to having fewer learnable parameters.

For semantic AD experiments on MPI3D, a red cone is fixed as the normal object (chosen arbitrarily), and models are trained on all available views. Anomalies are obtained by interventions on three underlying factors: (i) changing color to blue, (ii) transforming shape to cube, and (iii) increasing size.

Two additional degrees of freedom exist in the dataset: background color and camera height. Interventions on these have an outsized impact on images however, and do not provide any real challenge to a residual network (or any other modern vision architecture, for that matter), which is why they are not considered here.

For weak supervision through OE all remaining images are used that do not belong to neither the normal nor the anomaly class. For example white, green, brown, and olive all appear in the corpus Q_m.

Optimization The underlying model for DSVDD, SAD, ADRA, ADIB is the exact same ResNet26, optimized via SGD (momentum parameter of 0.9, weight decay of 10^{-4}) for a total of 100 epochs, with learning rate reductions by 1/10 after 60 and 80

epochs. The batch size is fixed to 128, and only standard augmentations are used. The regularization term Ω in ADIB is scaled with $\alpha = 10^{-2}$, following the recommendation of Li et al. (2018c).

For all models, parameters are initialized via the same weights θ_0 obtained from pretraining on ImageNet and then trained further on the downstream AD task. As noted in Section 4.3, in ADRA only linear corrections V are learned, while the backbone θ_0 is fixed. All experiments have been implemented with PyTorch (Paszke et al., 2019). Results are averaged over five seeds.

Results AUCs for different interventions are displayed in Figure 4.1. Detecting even the most simple semantic anomaly, such as a change in object color $z_{\text{color}} = \text{red} \rightarrow \text{blue}$ is impossible when learning without any weak supervision, as is the case for DSVDD (11.7 AUC in %).

The proposed intervention protocol confirms that it is beneficial to introduce a concept of differentness via OE. In other words, exposing models to the concept of red being normal, while also showing it examples of other colors (brown, green, etc.) *prepares* the model for *potential* anomalous shifts — although SAD has never seen a blue example, OE enables it to identify it as "*not* red", and hence an anomaly.

To obtain more robust models that can pick up on less obvious interventions such as changing the shape $z_{\text{shape}} = \text{cone} \rightarrow \text{cube}$ or $z_{\text{size}} = \text{small} \rightarrow \text{large}$, adequate forms of regularization appear to be critical. While it has fewer learnable parameters, ADRA improves performance over SAD under all interventions. Some performance gap remains, however, which is likely a consequence of the parameter-efficiency of ADRA, letting it rely more on the weights of the base network which potentially aren't *particularly* well suited for the task.

ADIB has a higher degree of flexibility, thus allowing for sample-efficient utilization of those features which *are* useful from the pretrained network. While ADIB might be a

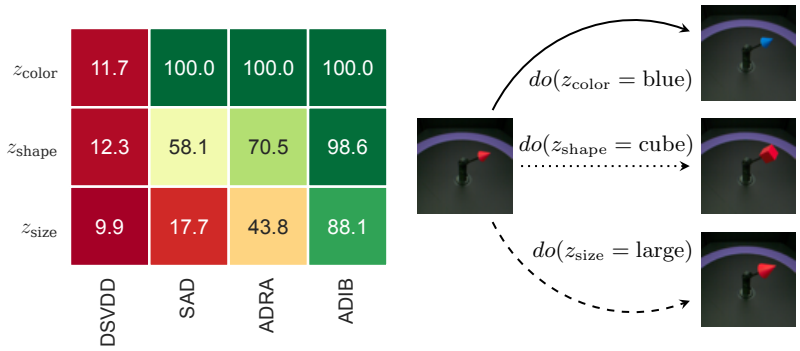


Figure 4.1: Left: Detection performance in AUC (in percentages) under interventions on images in MPI3D after training models on different views of red cones. At testing time, different interventions transform normal image examples (red cone depicted in the center) into an anomaly (e.g. blue cone, top right). DSVDD fails to register any of the interventions. While SAD (Ruff et al., 2020a,b) is sensitive to the change in color, transfer-based models can pick up on more subtle interventions on size and shape.

simple strategy for the transfer of rich semantic features to AD, the performance under all three interventions shows that it can robustly detect semantic anomalies.

Finally, it is noted that weak supervision through OE consistently increased disentanglement in the learned representations. DCI disentanglement (Eastwood and Williams, 2018) almost doubles from 0.068 for DSVDD to 0.103 for SAD, their only distinction being the absence and presence of weak supervision via Q_m, respectively. Locatello et al. (2020) made a similar observation in the context of unsupervised learning, finding that *some* weak supervision is required for disentanglement.

Non-Semantic Shift Recent work examined model robustness towards non-semantic shift, such as the appearance of color not contained in the training data, which can confuse models from their primary objective of detecting semantic categories (Ahmed et al., 2021). In order to examine this setting on MPI3D w.l.o.g. it is fixed *cube* = anomalous and *cone* = normal, while color is considered a non-semantic factor.

The experiment consists of a controlled sequence of trials: a single color (red) is included

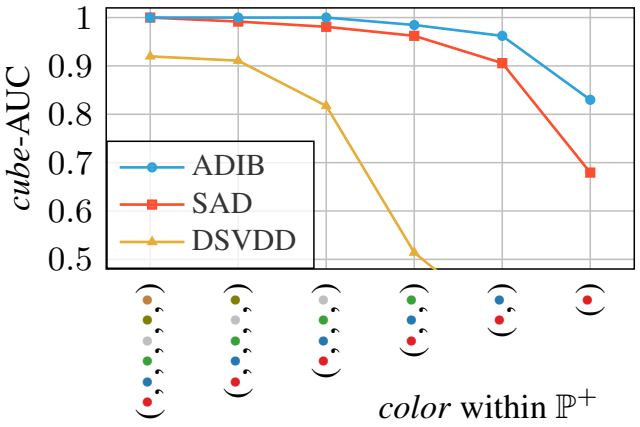


Figure 4.2: Robustness of detecting *cube* under non-semantic color shifts for DSVDD (no OE), SAD (uses OE), and the proposed ADIB. The x-axis indicates colors that are included in the distribution of normality $\mathbb{P}^+$.

in the normal data at first, and the detection performance of models for *cone* vs. *cube* (of any color) is evaluated. Then a second color is picked and added to the normal data (which now contains red and blue), after which models are trained and evaluated again. Repeating this for green, white, etc. yields a sequence of distributions $\mathbb{P}^+_{\text{red}}, \ldots, \mathbb{P}^+_{\text{all}}$ that gives precise control over the degree in which semantic context may be established.

Figure 4.2 shows the extent to which the transfer-based approach improves robustness to non-semantic shifts and underlines the importance of preventing drift from the transfer task. SAD makes use of OE (which in this experiment includes shapes other than *cone* and *cube*, but never additional colors) and enhances performance relative to DSVDD (which does not use OE). A gap remains however when context is established only through OE (SAD vs. ADIB). Especially for few colors in $\mathbb{P}^+$ transfer-based AD appears very useful to manifesting the right semantic context.

4.4.2 Semantic AD

In this section, models are evaluated on recently proposed benchmarks for semantic AD (Ahmed and Courville, 2020). This setup is equivalent to that in the motivation (Section 4.2), but here evaluations focus on a broad range of recent state-of-the-art AD models.

In the CIFAR-10 and STL-10 semantic AD benchmarks 9 out of 10 object classes form the normal data S_n (e.g. all classes except dogs), so that images from multiple classes form a multimodal distribution $\mathbb{P}^+$. The single class that is left out (i.e. dogs) is declared anomalous and never seen during training. At test time, the AD model has to identify the held-out class, i.e. it is measured whether $f_\theta(x) \approx 1$ when x contains a dog. This requires that the AD model has a good semantic understanding of the objects in $\mathbb{P}^+$, and Ahmed and Courville (2020) showed that this setup is more difficult than the popular one-versus-rest AD benchmark (see Section 4.4.3) or problems like detecting unseen domains (Section 4.4.4).

Ahmed and Courville (2020) determine semantic anomalies via MSP (Hendrycks and Gimpel, 2017) and ODIN (Liang et al., 2018) using an auxiliary self-supervised criterion akin to RotNet (Gidaris et al., 2018), while Bergman et al. (2020) use a nearest neighbor search over fixed pretrained features. All existing results are included in Table 4.2.

Optimization As before, the regularization strength is set to $\alpha = 10^{-2}$ following the suggestion of Li et al. (2018c); in elastic weight consolidation (EWC) the Fisher multiplier is fixed to 400, as recommended by Kirkpatrick et al. (2017).

For experiments on CIFAR-10 (Krizhevsky and Hinton, 2009), an inductive bias is introduced by regularizing network weights towards those of ResNet26 trained on ImageNet at 32x32 resolution. The same architecture is used for STL-10 (Coates et al., 2011), but since images have a higher resolution the initial model weights are obtained from training over 96x96 pixels.

	mAUC	mAP
ODIN (Ahmed and Courville, 2020)	—	41.6
GT (Golan and El-Yaniv, 2018)	61.7	—
kNN-AD (Bergman and Hoshen, 2020)	71.7	—
ADRA	95.0 *(0.1)*	72.9 *(0.4)*
ADIB	**95.1** *(0.1)*	**74.6** *(0.3)*

Table 4.2: Mean AUC (mAUC) and AP (mAP) in percent on the CIFAR-10 semantic AD benchmark. Standard deviations computed over five runs. ADRA and ADIB outperform existing AD approaches. Results for GT taken from Bergman and Hoshen (2020).

For eqs. (4.1) and (4.2) the normal set S_n is contrasted against images from an unstructured corpus Q_m. Following previous work that makes use of OE (Hendrycks et al., 2019b; Ruff et al., 2020a), for CIFAR-10 this is fixed to contain all samples from the CIFAR-100 training split. As already emphasized, Q_m equals weak supervision: CIFAR-100 gives a viable surrogate learning signal, however does not contain examples of the anomalous CIFAR-10 categories. STL-10 contains a large unlabeled split, which is used for OE.

Results There are discrepancies in how performance is reported in the semantic AD literature: some authors recommend average precision (AP) (Ahmed and Courville, 2020), while others report AUC (Bergman and Hoshen, 2020). Similar to AUC which measures recall as a function of the false negative rate (c.f. Section 4.2), AP is computed by varying the decision threshold and collecting a finite set of precision and recall values, followed by a weighted summation of these metrics at each threshold (indexed by t here):

$$\text{AP} = \sum_t (R_t - R_{t-1}) P_t \,.$$

Table 4.2 includes both metrics, and AP is reported for STL-10 (see Table 4.3) as this benchmark was so far only evaluated by Ahmed and Courville (2020) who report AP and remark that AUC is overly optimistic for the STL-10 semantic AD benchmark, see also Davis and Goadrich (2006).

On the multimodal CIFAR-10 semantic AD benchmark (see Table 4.2), ADIB out-

Class	ODIN	HSC	ADRA	ADIB
Airplane	23.4	23.1	**49.3** *(8.9)*	41.4 *(7.4)*
Bird	40.1	13.8	18.9 *(8.1)*	**44.0** *(2.9)*
Car	16.9	39.9	**74.6** *(6.5)*	72.2 *(10.5)*
Cat	31.4	18.9	29.6 *(3.4)*	**51.0** *(2.1)*
Deer	29.7	25.3	20.7 *(1.9)*	**43.0** *(5.7)*
Dog	26.1	17.3	26.6 *(3.5)*	**32.2** *(3.1)*
Horse	23.6	30.1	52.5 *(5.9)*	**53.7** *(2.5)*
Monkey	28.3	18.4	23.0 *(2.7)*	**46.6** *(1.9)*
Ship	15.4	49.2	**69.2** *(2.6)*	51.7 *(8.7)*
Truck	16.6	40.7	**64.3** *(2.2)*	58.7 *(3.6)*
mAP	25.1	27.7	42.9 *(1.4)*	**49.5** *(1.2)*

Table 4.3: APs in percent for different models and classes on the STL-10 semantic AD benchmark. Standard deviations over five runs reported in parentheses.

performs previously reported methods by a substantial margin (all in %): 74.6 vs. 41.6 mAP, and 95.1 vs. 71.7 mAUC. Even though it requires a smaller number of learnable parameters ADRA comes very close: 95.0 mAUC, and 72.9 mAP.

As the results confirm, inferring anomalies on STL-10 is significantly harder. In particular, even when using a state-of-the-art HSC classifier (Ruff et al., 2020a) initialized with pretrained θ_0 but without regularization Ω, this does not successfully address the semantic AD task (mAP of 27.7%, Table 4.3). When adding a regularization term performance improves to 35.0% mAP ($\mathtt{a}_3$ in Table 4.4), supporting the assumption that variations that are important to determining anomalies at test time are *forgotten* during training, yielding poorer performance across classes.

ADIB improves performance to 49.5% mAP. While having much fewer effective parameters, ADRA almost matches this performance (42.9% mAP) — interestingly, ADRA outperforms all other AD models on STL-10 man-made objects (cars, trucks, etc.). This is potentially due to there being a minority of examples of human-made objects in CIFAR-10 and ADRA contains many residual bypasses which, as reported in Section 3.4.3, can increase network robustness on smaller modes.

	$\mathcal{L}$	Tr.	Reg.	CIFAR-10	STL-10
$\mathtt{a}_1$	eq. (4.1)	✗	✗	60.3	32.9
$\mathtt{a}_2$	eq. (4.1)	✓	✗	64.9	38.6
$\mathtt{a}_3$	HSC	✓	L_2	68.5	35.0
$\mathtt{a}_4$	DOC	✓	✗	65.8	35.2
$\mathtt{a}_5$	eq. (4.2)	✓	EWC	66.4	39.7
$\mathtt{a}_6$	ADRA			72.9 *(0.3)*	42.9 *(1.4)*
$\mathtt{a}_7$	ADIB			**74.6** *(0.3)*	**49.5** *(1.2)*

Table 4.4: Ablations in terms of mAP (all results in percentages). Tr. indicates absence or presence of transfer learning; Reg. that of regularization. Included are comparisons against hyperspherical classifiers (Ruff et al., 2020a) in $\mathtt{a}_3$, and EWC (Kirkpatrick et al., 2017) in $\mathtt{a}_5$. Standard deviations (in parenthesis) computed over five runs.

Ablation The transfer of features from rich semantic tasks to AD has to be carried out *carefully*. This is examined in an ablation in Table 4.4, for which the exact same model is used in each experiment $\mathtt{a}_1$–$\mathtt{a}_7$, and only individual components are switched on and off: starting from random ($\mathtt{a}_1$) or pretrained models without regularization ($\mathtt{a}_2$) is not sufficient, as also highlighted in the intervention experiments in Section 4.4.1. Using an HSC loss (Ruff et al., 2020a) with the exact same explicit inductive bias through Ω that was used in ADIB reduces performance ($\mathtt{a}_3$ vs. $\mathtt{a}_7$). DOC (Perera and Patel, 2019) is conceptually very similar to HSC, combining a radial compactness loss with a descriptiveness loss that requires ImageNet data. The reported results ($\mathtt{a}_3$ vs. $\mathtt{a}_4$) confirm they also behave very similarly performance-wise.

In $\mathtt{a}_5$ it is found that EWC, a popular strategy for continual learning that regularizes weights via the Fisher information (Kirkpatrick et al., 2017), performs poorly compared to ADIB. This makes perfect sense, as EWC was designed to slow down learning on model weights relevant for the pretraining task: when pretraining on a demanding task like ImageNet, this can restrain capacity. Crucially for the AD setting model capacity is needed to free up and focus on the problem of semantic AD instead. In other words, as one never returns the model to ImageNet classification, there simply is no good reason why one would want to preserve performance for it. This ablation shows that, while they may be simple, the proposed strategies are surprisingly effective for AD.

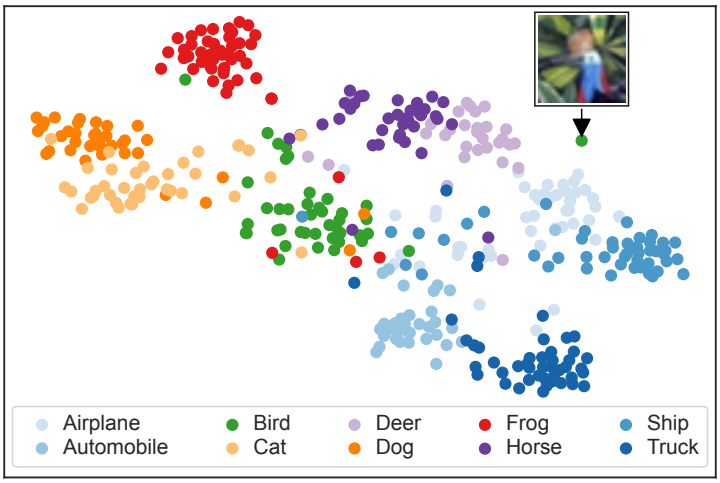
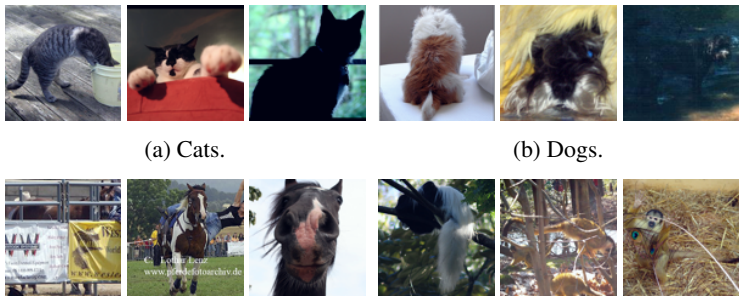


Figure 4.3: Due to its regularization towards weights of the pretrained model, ADIB appears to learn a feature space that preserves some important semantics. For example animal categories (• cats, • dogs, etc.) are separated from man-made ones (•, •, •, •). It also locates examples from the *unseen* bird category (•) nearby other animals. The arrow highlights a bird that gets mapped close to man-made objects, and identifying it as one indeed requires some imagination.

Qualitative Analysis Figure 4.3 presents the high semantic association between real-world object classes and their representation as learned by ADIB. Feature embeddings for samples from CIFAR-10 were computed and mapped to two dimensions using t-SNE (van der Maaten and Hinton, 2008) after learning ADIB on the CIFAR-10 semantic AD setup, i.e. trained on a multimodal $\mathbb{P}^+$ that contains 9 out of 10 classes (• cat, • dog, etc.). At test time the singular anomalous category (• bird) gets revealed to the model.

While ADIB has no direct access to categories or labels, it nonetheless appears able to semantically organize the different objects in $\mathbb{P}^+$. This is likely a consequence of ADIB being regularized towards the weights of an ImageNet-pretrained model, which captures a large amount of the semantics present in natural images.

For example • deer and • horses are similar categories, and also cluster together in ADIB's feature space. The same holds for • cats and • dogs which form a pair, while • frogs are separated from the remaining animal classes. Objects such as • cars, • trucks, etc. are



(a) Cats. (b) Dogs.

(c) Horses. (d) Monkeys.

Figure 4.4: Examples from different object classes in STL-10 that were assigned high anomaly scores by ADIB.

mapped to their own region, away from the animal categories. While the anomalous • bird class is not in $\mathbb{P}^+$ (so is never seen during training), it locates near other animals, a strong indication that information from the transfer task is preserved by regularization via Ω in eq. (4.3.2).

One bird (highlighted by an arrow in Figure 4.3) has a feature representation that differs from those of other animals, and lies closer to man-made objects (cars, trucks, etc.). It is however difficult to identify a bird in the image, explaining its location relatively far away from the •-cluster in feature space.

Figure 4.4 displays examples from STL-10 that have been assigned a high anomaly score by ADIB. The anomalous images are indeed unusual: either because animals appear in an unexpected pose (e.g. cat reaching for camera), because of the presence of captions, or in some cases – such as dogs – because the underlying object class is almost impossible to discern from the image.

4.4.3 Non-Semantic AD

This section evaluates performance of ADIB and ADRA on the standard CIFAR-10 one-versus-rest AD benchmark. This benchmark is reported across large parts of the

AD literature (Ruff et al., 2018; Golan and El-Yaniv, 2018; Hendrycks et al., 2019b; Abati et al., 2019; Hendrycks et al., 2019c; Perera et al., 2019; Ruff et al., 2020a,b) and therefore is still meaningful for comparison of the proposed methods to previous models.

In some sense, this benchmark can be viewed as opposite of semantic AD evaluated in Section 4.4.2: only a single object class is fixed as the normal class — say, dogs. All dogs in the CIFAR-10 training split are collected into S_n (so 5000 out of 50 000 total samples), from which models are trained. Models are then evaluated against the entire CIFAR-10 test split, and performance is measured by checking whether anomaly scores assigned to dogs are lower than scores assigned to all nine remaining non-dog classes.

While the optimization settings remain unchanged from Section 4.4.2, for this benchmark previous works almost exclusively report AUC, and this custom is followed here.

It should be noted that this benchmark constitutes a less complex problem than the semantic AD benchmark of Section 4.4.2. In particular when singling out objects that differ significantly from others in CIFAR-10, for example ships or trucks, shallower feature representations are sufficient to detecting them, which also manifests in relatively large AUCs for such distinct classes. The benchmark has therefore recently been declared a non-semantic problem by Ahmed and Courville (2020).

Results As shown in Table 4.5, ADIB raises the current state of the art to 99.1% mAUC, a marked gap to the previous best method with 96.1% mAUC. As demonstrated by the performance of kNN-AD (Bergman et al., 2020), simply using features from a large pretrained network is inferior when looking to detect anomalies.

These results suggest that favorable inductive biases are critical for utilizing AD models to their full potential. ADRA once again comes very close in terms of performance, while requiring a much smaller number of learnable parameters.

Class	GT	kNN	GT+	HSC	ADRA	ADIB
Airplane	74.7	93.9	90.4	96.7	99.0 *(0.1)*	**99.2** *(0.3)*
Automobile	95.7	97.7	99.3	98.9	99.7 *(0.1)*	**99.8** *(0.1)*
Bird	78.1	85.5	93.7	93.2	97.5 *(0.4)*	**98.6** *(0.2)*
Cat	72.4	85.5	88.1	90.6	96.3 *(0.4)*	**97.0** *(0.7)*
Deer	87.8	93.6	97.4	97.1	98.9 *(0.1)*	**99.3** *(0.1)*
Dog	87.8	91.3	94.3	94.7	97.7 *(0.2)*	**98.2** *(0.3)*
Frog	83.4	94.3	97.1	98.0	**99.6** *(0.1)*	**99.6** *(0.2)*
Horse	95.5	93.6	98.8	97.9	99.6 *(0.1)*	**99.8** *(0.1)*
Ship	93.3	95.1	98.7	98.2	99.5 *(0.1)*	**99.6** *(0.1)*
Truck	91.3	95.3	98.5	97.7	99.4 *(0.1)*	**99.5** *(0.2)*
mAUC	86.0	92.5	95.6	96.3	98.7 *(0.1)*	**99.1** *(0.1)*

Table 4.5: AUCs for different methods on the CIFAR-10 one-versus-rest AD benchmark. Included are geometric transformations (GT) (Golan and El-Yaniv, 2018), kNN-AD (Bergman et al., 2020), self-supervised transformations (GT+) (Hendrycks et al., 2019c), and hyperspherical classifiers (HSC) (Ruff et al., 2020a). Parentheses show standard deviations computed over five runs.

OE Dataset	HSC	ADRA	ADIB
SVHN	70.2	75.3 (+5.1)	79.8 (+9.6)
CIFAR-100	96.3	98.7 (+2.4)	99.1 (+2.8)

Table 4.6: Ablations on the CIFAR-10 one-versus-rest AD benchmark for different choices of OE. Results shown in percentages, alongside relative gain (vs. HSC) displayed in parentheses.

Ablation Recent work examined the hierarchical relationship between distributions for out-of-distribution detection (Schirrmeister et al., 2020). Taking inspiration from this study, here the role of CIFAR-100 as OE is critically examined in an ablation that compares it to the use of SVHN as OE.

Results in Table 4.6 make it evident that SVHN is less well suited for CIFAR-10, as performance drops for all methods. HSC, the current state-of-the-art AD method using OE, achieves 70.2% mAUC here. A sizeable drop, but still improving from 64.8% mAUC for DSVDD, the mathematical equivalent to using no OE.

ADIB obtains 79.8% mAUC when coupled with SVHN, a gain of +9.6% over HSC. This

is a considerably larger difference than that for CIFAR-10 coupled with CIFAR-100 of +2.8% reported in Table 4.5 (ADIB: 99.1% vs. HSC: 96.3% mAUC), indicating that transfer-based AD benefits performance more when not using CIFAR-100 as OE (albeit it is better suited overall). In other words, the importance of the transfer task *increases* as the suitability of OE decreases.

4.4.4 Anomalous Domains

Another AD setting that can be considered a non-semantic problem is the detection of anomalous domains, in particular when these differ considerably as is the case for PACS (c.f. Figure 3.7 which shows examples from each domain).

For this experiment, the normal class encompasses all training examples from three out of four domains, e.g. includes *art painting*, *cartoon*, and *sketch*, and it is checked whether models assign higher scores to examples from the anomalous domain (*photo*) at test time. ImageNet (Deng et al., 2009) is used as OE, with no changes to the optimization settings used in previous sections.

While shallow methods can in principle be used to detect the more obvious domains (*sketch* in particular), transfer-based methods registered strong performance on more difficult domains (see Table 4.7).

Due to the presence of latent domains in $\mathbb{P}^+$, the methods proposed in other chapters of this thesis benefit this problem directly: performance is increased by 2.87% mAUC when coupling SAD (Ruff et al., 2020b), which uses BN, with MN instead (introduced in Chapter 2).

Extending the adaptation mechanism in ADRA with SLA (Chapter 3) boosts performance for all domains except *photo*, in all likelihood because this domain is relatively similar to the pretraining task, so that the extra flexibility of SLA does more harm than good. ADIB once again delivers very strong performance, and obtains 80.50% mAUC here.

	A	C	P	S	mAUC
SAD (Ruff et al., 2020b)	23.29 *(3.48)*	44.39 *(3.67)*	56.89 *(2.09)*	99.97 *(0.10)*	56.14 *(2.50)*
MN (Chapter 2)	32.50 *(1.64)*	45.54 *(2.23)*	58.02 *(1.38)*	99.96 *(0.20)*	59.01 *(1.24)*
ADRA	55.23 *(3.47)*	56.91 *(3.18)*	71.12 *(2.01)*	99.98 *(0.22)*	70.81 *(2.08)*
SLA (Chapter 3)	60.28 *(3.80)*	**81.12** *(3.41)*	67.28 *(2.42)*	99.68 *(0.53)*	77.09 *(2.43)*
ADIB	**72.56** *(4.12)*	72.54 *(1.93)*	**76.92** *(1.80)*	**99.99** *(0.11)*	**80.50** *(2.08)*

Table 4.7: AUCs in percent for anomalous domain detection on PACS, where the column heads indicate which domain was set aside during training. MN is inserted into SAD, and SLA used to replace the adaptation mechanism in ADRA. Standard deviations (in parentheses) were measured over five runs.

4.4.5 Robustness to Small Modes

An ideal AD model has the ability to incorporate information from normal examples even if they form only a minor mode of $\mathbb{P}^+$, in the sense that only few samples from this class are contained in S_n — for example a rare dog breed. Since AD is concerned with low-probability events, the ability to robustly incorporate such small modes from few examples is of special importance.

To measure AD robustness, the following experiment lets the normal class be constituted by samples associated with two classes (y_a, y_b), such that $S_n \sim \frac{1}{r+1}\mathbb{P}_{y_a} + \frac{r}{r+1}\mathbb{P}_{y_b}$, where the minor mode amplitude $r \in [0, 1]$ controls the number of examples from y_b in the normal data.

For a robust AD model, even as S_n is relaxed to contain only examples from y_a, its ability to identify the smaller category y_b as non-anomalous would remain intact.

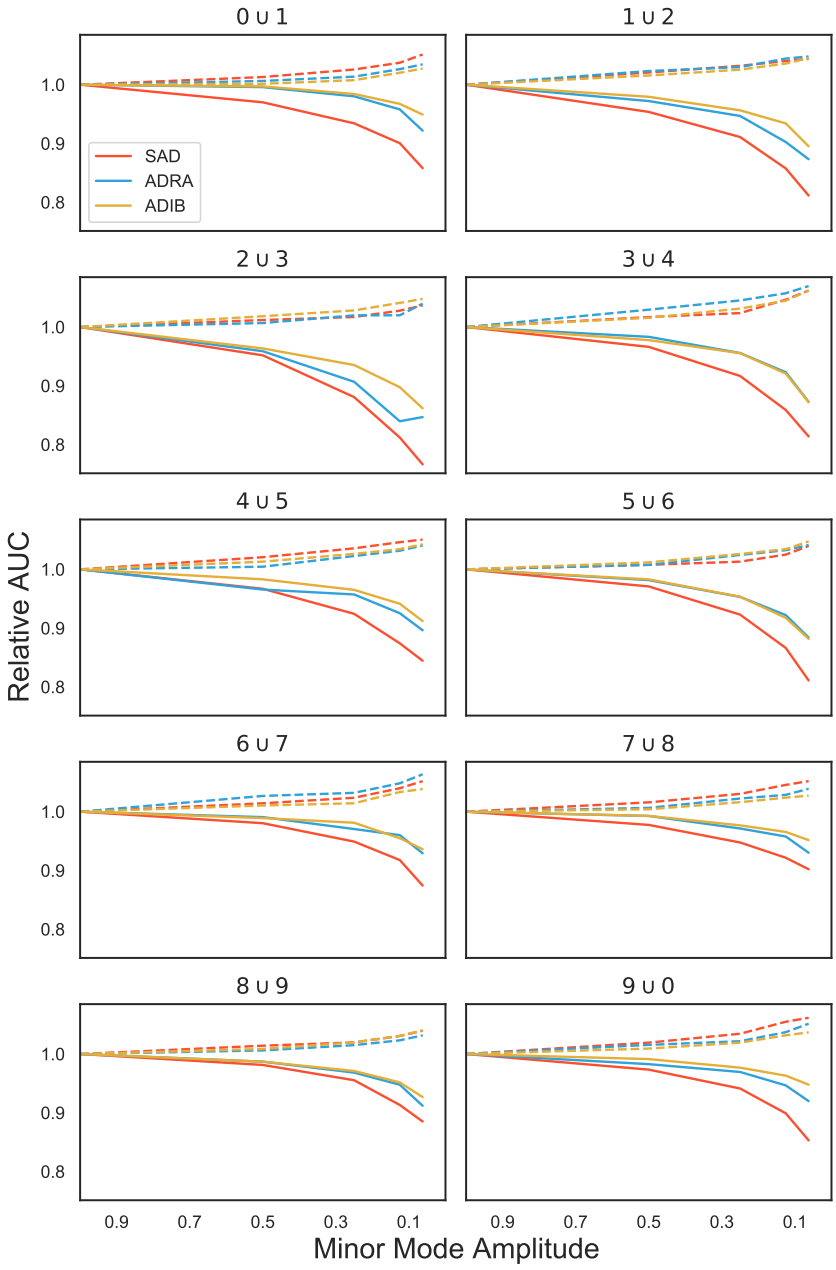


Figure 4.5: Relative AUCs for secondary object categories, with dashed curves displaying primary class performance. In the top-left figure (y_a, y_b) = ("airplane", "automobile"), for example. A faster drop for the secondary class signals a less robust AD model.

CIFAR-10 is used here, and primary and secondary AUCs are reported as a function of r for different class pairings, e.g. y_a = "ship" and y_b = "truck". ADIB and ADRA are compared to to SAD (Ruff et al., 2020b) with pretrained weights, which corresponds to ADIB with $\alpha = 0$, i.e. without a regularization term Ω.

As Figure 4.5 shows, for SAD performance for the secondary class decreases much faster than for ADIB or ADRA, a trend that was found to be consistent across class pairings, indicating that adequate transfer-based regularization as in ADIB and ADRA is important to robustly incorporating small modes of data in the normal class.

4.5 Conclusion

Detecting semantic anomalies is a difficult task, due to the infinite and complex ways these can manifest in data. This chapter proposed two new transfer-based methods to account for such complexities: ADIB sets a new state of the art in semantic AD tasks and ADRA provides a highly efficient, yet surprisingly effective learning protocol.

Interventions were used to examine different AD paradigms, and it was shown that transfer-based AD can detect subtle semantic anomalies. An interesting question for future research is whether detecting semantic anomalies requires disentanglement, and if it can benefit from the ongoing development of disentangled representations. The next chapter discusses more aspects reserved for future work.

Chapter 5

Conclusion

Learning from multimodal data presents several challenges for standard computer vision models. This thesis presented new methods to improve performance in such learning scenarios, proposing a new multimodal normalization, as well as novel approaches for latent domain learning and semantic anomaly detection.

In the following, Section 5.1 discusses existing limitations and highlights future research directions for normalization, latent domain learning, and anomaly detection. This is followed by a discussion in Section 5.2 that aims to assess the broader impact of the methods and ideas proposed in this thesis.

5.1 Future Work

Normalization One limitation of existing normalization methods is that they require large batch sizes for accurate estimates of the population statistics μ and σ (Wu and He, 2018). This issue also affects MN, since mode-level estimators become less precise as the number of normalizations is increased, i.e. as K grows.

While methods such as GN and MGN do not share this restriction, for many learning problems, e.g. classification, their performance simply does not match that of BN, which therefore remains the default option in modern network architectures. This calls for new normalization methods that robustly perform when the batch size is reduced.

Given that all other parameters in deep learning are found by optimizing the empirical risk, c.f. eq. (2.1), the direct estimation of μ, σ from the mini-batch, as done in BN and MN, is a somewhat unusual strategy. An interesting alternative is to carry out normalization by determining μ, σ via maximum likelihood estimation.

The main benefit of such normalizations is that they would function with any batch size, even $N = 1$. In addition, it is unclear how to integrate BN with more advanced gradient-based optimization routines, for example gradient episodic memory (Lopez-Paz and Ranzato, 2017) which constrains gradients to counter forgetting in continual tasks (c.f. Section 3.1.5). This becomes straightforward when the normalization is learned from gradients as well, alongside the rest of the network.

Another open question is the amount of flexibility required in dynamic gating. While MN needed gates to be light-weight and therefore used a relatively simple parametrization, more advanced gating mechanisms should be able to pick up on additional signals to better align modes in data. Solutions of this kind could become computationally feasible if they were combined with strategies for dynamically detecting where to insert such layers in networks, and which regions can do without, which is an interesting research problem from the perspective of neural architecture search (Zoph and Le, 2017; Liu et al., 2018; Mellor et al., 2021).

Latent Domain Learning The modeling solutions presented in Chapter 3 are geared towards learning unified representations over visually diverse data. Rekindling modules originally devised for traditional (annotated) multi-domain problems to latent domains required the replacing of domain-level assignments, such that – where adequate – multimodal data with different visual characteristics may still be processed separately.

This can be done for more methods besides residual adaptation (the basis for SLA). For example, a recent study decomposed convolutions for multiple domains (Guo et al., 2019a). The associated convolutions (one per domain) could be substituted, modifying layers from containing D of them to a preassigned number targeted by gates.

A largely unexplored but realistic scenario locates itself between latent domain learning and multi-domain learning, i.e. when domains are annotated for a subset of examples $\{(x_1, d_1), \ldots, (x_M, d_M)\}$, whereas others $\{x_{M+1}, \ldots, x_N\}$ are not. This problem has been investigated by Mancini et al. (2018, 2019) for DA, but arguably also deserves attention in the context of multi-domain learning: for a small number of examples annotating domains, e.g. photo vs. clipart, appears relatively straightforward. Annotating an entire database of images however, as required in most multi-domain methods, is not.

This scenario bears many interesting research problems: for example, which share M/N should ideally be labeled, and what is the dependence on the shape of $\mathbb{P}$? Which problems can do without any domain labels at all, i.e. when $M = 0$ as in latent domain learning? And how should domain information be incorporated into models in this case: globally, as in many multi-domain strategies, or locally, as in SLA? And perhaps model-free alternatives are more desirable, for example gradient-based ones? The problem of learning over partially annotated domains also extends into the purview of active learning (Sinha et al., 2019; Requeima et al., 2019): given a domain-labeled subset to start with, which additional examples would benefit most from getting annotated?

Anomaly Detection Anomalies can appear in data for many different reasons, and transfer-based strategies, proposed in Chapter 4, showed it is important to prepare models adequately for their variety. It should be noted however that, just like the majority of existing works in AD, this thesis assumed that S_n is clean, so does not contain any outliers. The contamination of the normal class was studied in the classical literature (Kim and Scott, 2012) and for autoencoders (Zhou and Paffenroth, 2017), but in the context of deep AD this aspect seems to not have received the attention it arguably deserves. Transfer-based approaches are an interesting candidate for such problems,

because its initial representations may allow spotting outliers *before* incorporation of the normal class, which otherwise has to occur in tandem (Liu et al., 2014).

Another interesting extension is to study model robustness for semantic anomaly detection as data is corrupted (Hendrycks and Dietterich, 2019). For example, such a study could be constructed by embedding new object categories into previously unseen modes, merging research aspects of AD and DG. According to Mancini et al. (2020a) instance-level mixing (Zhang et al., 2018) under a curriculum (Bengio et al., 2009) are promising tools for such research, both of which have so far not been adapted to deep AD.

Yet another challenging problem in AD is the susceptibility to adversarial attacks. These small perturbations, invisible to the human eye, drastically influence a model's perception of the object displayed in an image (Papernot et al., 2017; Madry et al., 2018). Devising methods that can dodge such confusions is important, and results that showed a positive link between adversarial robustness and transfer learning (Shafahi et al., 2020; Salman et al., 2020) make this a promising direction for future (transfer-based) AD research.

5.2 Broader Impact

Normalization is ubiquitous in deep learning, contributing both to its beneficial applications – medicine (Esteva et al., 2019), cosmology (Ishida, 2019), or pharmaceutics (Aliper et al., 2016) – as much as the disputable, e.g. mass surveillance.

Like with other deep normalization techniques, while MN itself can be viewed neutrally, this does not encompass its application: when used in questionable settings the better accounting for different subpopulations in data may have negative consequences for them. At the same time, improving the performance of e.g. cell classification (Chen et al., 2016) for different cell types, or achieving more robust performance across diverse groups in visual data (Wang et al., 2020), are both very desirable.

Latent Domain Learning In most cases, a sufficiently complex distribution will have multiple subregions that are of interest. In latent domain learning, this property is assumed a priori, allowing the decomposition of global accuracy metrics into their individual components. As the experiments in Section 3.4 highlighted, performance gains can often be traced back to networks focusing their modeling power onto the larger modes in data. In turn, this means information from less densely sampled subregions of the distribution gets suppressed.

A limitation of SLA is that, compared to more conventional mixture models' clustering assumption, its latent structure is less interpretable, which makes validating and establishing trust in its learned representations harder. One can be cautiously optimistic however that latent domain learning can be a useful tool in better understanding of how deep learning models may be prevented from fitting some regions in $\mathbb{P}$ at the cost of others, with potential benefits for model fairness (Hardt et al., 2016; Fish et al., 2016; Corbett-Davies et al., 2017) in the future.

Anomaly Detection has many applications. While progress for medical imaging, astronomy, or fraud detection will likely be of broader benefit, its uses in e.g. monitoring and surveillance should be viewed more critically.

An important algorithmic limitation of transfer-based AD is that, like most deep approaches, it suffers an implicit bias towards the predominant modes in data. Second, ADIB and ADRA are learned end-to-end, starting from a complex semantic representation. Therefore a precise understanding of its learned representations is challenging due to an associated lack of interpretability.

On the positive side, transfer-based AD gives rise to powerful yet simple models for AD, with ADRA in particular being highly parameter-efficient. As their performance shows, this doesn't necessarily mean a constraint on performance, which will hopefully encourage new AD models that also keep an eye on efficiency. Lastly, transfer-based AD improves the utilization of pretrained networks, allowing practitioners who do not

have access to large stores of data or computational resources to nonetheless produce high-performance machine learning tools.

Bibliography

D. Abati, A. Porrello, S. Calderara, and R. Cucchiara. Latent space autoregression for novelty detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2019.

M. Abdollahzadeh, T. Malekzadeh, and N.-M. M. Cheung. Revisit multimodal meta-learning through the lens of multi-task learning. *Advances in Neural Information Processing Systems*, 2021.

A. Adhikari, A. Ram, R. Tang, and J. Lin. DocBERT: BERT for document classification. *arXiv preprint arXiv:1904.08398*, 2019.

F. Ahmed and A. Courville. Detecting semantic anomalies. In *AAAI Conference on Artificial Intelligence*, pages 3154–3162, 2020.

F. Ahmed, Y. Bengio, H. van Seijen, and A. Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2021.

S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. GANomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conference on Computer Vision*, pages 622–637. Springer, 2018.

K. Akuzawa, Y. Iwasawa, and Y. Matsuo. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 315–331, 2019.

E. Alberti, A. Tavera, C. Masone, and B. Caputo. IDDA: a large-scale multi-domain dataset for autonomous driving. *IEEE Robotics and Automation Letters*, 5(4):5526–5533, 2020.

A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov. Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data. *Molecular Pharmaceutics*, 13(7):2524–2530, 2016.

A. Arnold, R. Nallapati, and W. W. Cohen. A comparative study of methods for transductive transfer learning. In *IEEE International Conference on Data Mining Workshops*, pages 77–82, 2007.

D. Arpit, Y. Zhou, B. Kota, and V. Govindaraju. Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. In *International Conference on Machine Learning*, pages 1168–1176, 2016.

Y. M. Asano, C. Rupprecht, and A. Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2020.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

E. Bagdasaryan, O. Poursaeed, and V. Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems*, pages 15479–15488, 2019.

Y. Balaji, S. Sankaranarayanan, and R. Chellappa. MetaReg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 998–1008, 2018.

T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (2):423–443, 2018.

S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, et al. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 137–144, 2007.

A. Bendale and T. E. Boult. Towards open set deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

E. Bengio, P.-L. Bacon, J. Pineau, and D. Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *International Conference on Machine Learning*, pages 41–48, 2009.

A. Berg, M. Felsberg, and J. Ahlberg. Unsupervised adversarial learning of anomaly detection in the wild. In *European Conference on Artificial Intelligence*, pages 1002–1008, 2020.

L. Bergman and Y. Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020.

L. Bergman, N. Cohen, and Y. Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint arXiv:2002.10445*, 2020.

P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019.

R. Berriel, S. Lathuillere, M. Nabi, T. Klein, T. Oliveira-Santos, N. Sebe, and E. Ricci. Budget-aware adapters for multi-domain learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 382–391, 2019.

A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643, 2019.

S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9):2137–2155, 2009.

H. Bilen and A. Vedaldi. Integrated perception with recurrent multi-task neural networks. In *Advances in Neural Information Processing Systems*, pages 235–243, 2016.

H. Bilen and A. Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv:1701.07275*, 2017.

F. C. Borlino, A. D'Innocente, and T. Tommasi. Rethinking domain generalization baselines. In *International Conference on Pattern Recognition*, pages 9227–9233. IEEE, 2020.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. Springer, 2010.

D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI Conference on Artificial Intelligence*, pages 2095–2102, 2018.

M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

A. Bulat, J. Kossaifi, G. Tzimiropoulos, and M. Pantic. Incremental multi-domain learning with network latent tensor factorization. *AAAI Conference on Artificial Intelligence*, pages 10470–10477, 2019.

Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

C. Campbell and K. P. Bennett. A linear programming approach to novelty detection. In *Advances in Neural Information Processing Systems*, pages 395–401, 2001.

K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pages 1567–1578, 2019.

O. Cappé and E. Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo. MultiDIAL: Domain alignment layers for (multisource) unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

X. Chang, T. M. Hospedales, and T. Xiang. Multi-level factorisation net for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2109–2118, 2018.

A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2018.

C. L. Chen, A. Mahjoubfar, L.-C. Tai, I. K. Blaby, A. Huang, K. R. Niazi, and B. Jalali. Deep learning in label-free cell classification. *Scientific Reports*, 6(1):1–16, 2016.

T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257*, 2017.

F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

K. Ciosek, V. Fortuin, R. Tomioka, K. Hofmann, and R. Turner. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2020.

A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

J. Collins, K. Howe, and B. Nachman. Anomaly detection for resonant new physics with machine learning. *Physical Review Letters*, 121(24):241803, 2018.

R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. In *Advances in Neural Information Processing Systems*, pages 633–640, 2002.

S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

H. Daumé III. Frustratingly easy domain adaptation. In *Annual Meeting of the Association of Computational Linguistics*, pages 256–263, 2007.

J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *International Conference on Machine Learning*, pages 233–240, 2006.

L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Image anomaly detection with generative adversarial networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–17. Springer, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning*, pages 980–989, 2017.

G. Desjardins, K. Simonyan, R. Pascanu, et al. Natural neural networks. In *Advances in Neural Information Processing Systems*, pages 2071–2079, 2015.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell. Semi-supervised domain adaptation with instance constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 668–675, 2013.

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: a deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, pages 647–655, 2014.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

H. Dutta, C. Giannella, K. Borne, and H. Kargupta. Distributed top-k outlier detection from astronomy catalogs using the DEMAC system. In *SIAM International Conference on Data Mining*, pages 473–478, 2007.

N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid. Blitznet: A real-time deep network for scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4154–4162, 2017.

C. Eastwood and C. K. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

F. Edgeworth. On discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(5):364–375, 1887.

D. Eigen, M. Ranzato, and I. Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong. Systematic construction of anomaly detection benchmarks from real data. In *ACM SIGKDD Workshop on Outlier Detection and Description*, pages 16–21. ACM, 2013.

S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019.

Z. Fang, J. Lu, A. Liu, F. Liu, and G. Zhang. Learning bounds for open-set learning. In *International Conference on Machine Learning*, pages 3122–3132, 2021.

B. Fish, J. Kun, and Á. D. Lelkes. A confidence-based approach for balancing fairness and accuracy. In *SIAM International Conference on Data Mining*, 2016.

G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1), 2016.

L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

C. Geng, S.-j. Huang, and S. Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Z. Ghafoori and C. Leckie. Deep multi-sphere support vector data description. In *SIAM International Conference on Data Mining*, pages 109–117, 2020.

M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *International Conference on Computer Vision*, pages 2551–2559, 2015.

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

R. Girshick. Fast R-CNN. In *International Conference on Computer Vision*, pages 1440–1448, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018.

M. W. Gondal, M. Wuthrich, D. Miladinovic, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *Advances in Neural Information Processing Systems*, pages 15714–15725, 2019.

B. Gong, K. Grauman, and F. Sha. Reshaping visual datasets for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1286–1294, 2013.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

N. Görnitz, M. Braun, and M. Kloft. Hidden markov anomaly detection. In *International Conference on Machine Learning*, pages 1833–1842, 2015.

S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain. DROCC: Deep robust one-class classification. In *International Conference on Machine Learning*, pages 11335–11345, 2020.

A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471, 2015.

G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei. Dynamic task prioritization for multitask learning. In *European Conference on Computer Vision*, pages 270–287, 2018.

Y. Guo, Y. Li, L. Wang, and T. Rosing. Depthwise convolution is all you need for learning multiple visual domains. In *AAAI Conference on Artificial Intelligence*, pages 8368–8375, 2019a.

Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris. Spottune: transfer learning through adaptive fine-tuning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4805–4814, 2019b.

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision*, pages 1026–1034, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2969, 2017.

K. He, R. Girshick, and P. Dollár. Rethinking ImageNet pre-training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4918–4927, 2019.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721, 2019a.

D. Hendrycks, M. Mazeika, and T. G. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019b.

D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15637–15648, 2019c.

D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*, 2020.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 1991.

J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *European Conference on Computer Vision*, 2012.

J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1989–1998, 2018.

S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao. Online learning: A comprehensive survey. *arXiv preprint arXiv:1802.02871*, 2018.

S. Hooker, N. Moorosi, G. Clark, S. Bengio, and E. Denton. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao. Normalization techniques in training DNNs: Methodology, analysis and application. *arXiv preprint arXiv:2009.12836*, 2020.

X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *International Conference on Computer Vision*, pages 1510–1519, 2017.

S. Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Advances in Neural Information Processing Systems*, pages 1942–1950, 2017.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

E. E. Ishida. Machine learning and the future of supernova cosmology. *Nature Astronomy*, 3(8):680–682, 2019.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, pages 79–87, 1991.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

K. Jarrett, K. Kavukcuoglu, Y. LeCun, et al. What is the best multi-stage architecture for object recognition? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2146–2153, 2009.

X. Jiang, Q. Lao, S. Matwin, and M. Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 4816–4827, 2020.

M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

M. M. Kalayeh and M. Shah. Training faster by separating modes of variation in batch-normalized models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1483–1500, 2019.

K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing. Neural architecture search with Bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems*, pages 2016–2025, 2018.

Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*, pages 521–528, 2011.

A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Learning Representations*, 2018.

J. Kim and C. D. Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(1):2529–2565, 2012.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

B. Kleinberg, Y. Li, and Y. Yuan. An alternative view: When does SGD escape local minima? In *International Conference on Machine Learning*, pages 2698–2707, 2018.

J. Kohler, H. Daneshmand, A. Lucchi, M. Zhou, K. Neymeyr, and T. Hofmann. Towards a theoretical understanding of batch normalization. *arXiv preprint arXiv:1805.10694*, 2018.

I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6129–6138, 2017.

S. Kornblith, J. Shlens, and Q. V. Le. Do better ImageNet models transfer better? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019.

D. Kotovenko, A. Sanakoyeu, S. Lang, and B. Ommer. Content and style disentanglement for artistic style transfer. In *International Conference on Computer Vision*, pages 4422–4431, 2019.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Y. LeCun. The MNIST database of handwritten digits, 1998.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten ZIP code recognition. *Neural Computation*, 1(4):541–551, 1989.

Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995, 1995.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998a.

Y. LeCun, L. Bottou, G. Orr, and K.-R. Müller. Efficient backprop in neural networks: Tricks of the trade. *Lecture Notes in Computer Science*, 1524, 1998b.

T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020.

D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *International Conference on Computer Vision*, pages 5542–5550, 2017.

D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, pages 3490–3497, 2018a.

D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales. Episodic training for domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1455, 2019a.

H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018b.

X. Li, Y. Grandvalet, and F. Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834, 2018c.

X. Li, S. Chen, X. Hu, and J. Yang. Understanding the disharmony between dropout and batch normalization by variance shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2682–2690, 2019b.

X. Li, W. Sun, and T. Wu. Attentive normalization. In *European Conference on Computer Vision*, pages 70–87, 2020.

Y. Li and N. Vasconcelos. Efficient multi-domain learning by covariance normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5424–5433, 2019.

Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao. Domain generalization via conditional invariant representations. In *AAAI Conference on Artificial Intelligence*, pages 3579–3587, 2018d.

Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *European Conference on Computer Vision*, pages 624–639, 2018e.

Y. Li, M. Murias, S. Major, G. Dawson, and D. Carlson. On target shift in adversarial domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 616–625, 2019c.

J. Liang, Y. Wang, D. Hu, R. He, and J. Feng. A balanced and uncertainty-aware approach for partial domain adaptation. In *European Conference on Computer Vision*, pages 123–140, 2020.

P. Liang and D. Klein. Online EM for unsupervised models. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages 611–619, 2009.

S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

M. Lin, Q. Chen, and S. Yan. Network in network. *International Conference on Learning Representations*, 2014a.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014b.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2980–2988, 2017a.

Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. 2017b.

A. Liu, S. Tang, X. Liu, X. Chen, L. Huang, Z. Tu, D. Song, and D. Tao. Towards defending multiple adversarial perturbations via gated batch normalization. *arXiv preprint arXiv:2012.01654*, 2020a.

C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *European Conference on Computer Vision*, pages 19–34, 2018.

S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019a.

S. Liu, D. Papailiopoulos, and D. Achlioptas. Bad global minima exist and SGD can reach them. In *Advances in Neural Information Processing Systems*, pages 8543–8552, 2020b.

W. Liu, G. Hua, and J. R. Smith. Unsupervised one-class learning for automatic outlier removal. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3826–3833, 2014.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, pages 3730–3738, 2015.

Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019b.

P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K.-R. Müller. Explainable deep one-class classification. In *International Conference on Learning Representations*, 2021.

F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124, 2019.

F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359, 2020.

D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.

E. Lubana, R. Dick, and H. Tanaka. Beyond BatchNorm: towards a unified understanding of normalization in deep learning. *Advances in Neural Information Processing Systems*, 34, 2021.

S. Lyu and E. P. Simoncelli. Nonlinear image representation using divisive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1975–1981, 2010.

A. Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.

P. Maini, E. Wong, and Z. Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pages 6640–6650, 2020.

A. Mallya, D. Davis, and S. Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *European Conference on Computer Vision*, pages 67–82, 2018.

M. Mancini, L. Porzi, S. Rota Bulò, B. Caputo, and E. Ricci. Boosting domain adaptation by discovering latent domains. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3771–3780, 2018.

M. Mancini, L. Porzi, S. R. Bulo, B. Caputo, and E. Ricci. Inferring latent domains for unsupervised deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

M. Mancini, Z. Akata, E. Ricci, and B. Caputo. Towards recognizing unseen categories in unseen domains. In *European Conference on Computer Vision*, pages 466–483, 2020a.

M. Mancini, E. Ricci, B. Caputo, and S. R. Bulo. Boosting binary masks for multi-domain learning through affine transformations. *Machine Vision and Applications*, 31 (6):1–14, 2020b.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, pages 1041–1048, 2008.

A. Martins and R. Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623, 2016.

M. Mathieu, J. Zhao, P. Sprechmann, A. Ramesh, and Y. LeCun. Disentangling factors of variation in deep representations using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016.

T. Matsuura and T. Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI Conference on Artificial Intelligence*, pages 11749–11756, 2020.

J. Mellor, J. Turner, A. Storkey, and E. J. Crowley. Neural architecture search without training. In *International Conference on Machine Learning*, pages 7588–7598, 2021.

T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin. Advances in pre-training distributed word representations. In *International Conference on Language Resources and Evaluation*, 2018.

I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *International Conference on Computer Vision*, pages 5715–5725, 2017.

K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013.

S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1863–1868, 2006.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning*, pages 807–814, 2010.

H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning*, pages 689–696, 2011.

P. C. Ngo, A. A. Winarto, C. K. L. Kou, S. Park, F. Akram, and H. K. Lee. Fence GAN: towards better anomaly detection. In *IEEE International Conference on Tools with Artificial Intelligence*, pages 141–148, 2019.

C. Nguyen, T. Hassner, M. Seeger, and C. Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pages 7294–7305, 2020.

D. Ourston, S. Matzner, W. Stump, and B. Hopkins. Applications of hidden markov models to detecting multi-stage network attacks. In *International Conference on System Sciences*. IEEE, 2003.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *ACM Asia Conference on Computer and Communications Security*, pages 506–519, 2017.

G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019.

D. Pelleg and A. Moore. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2004.

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

X. Peng, Z. Huang, X. Sun, and K. Saenko. Domain agnostic learning with disentangled representations. *arXiv preprint arXiv:1904.12347*, 2019b.

P. Perera and V. M. Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019.

P. Perera, R. Nallapati, and B. Xiang. OCGAN: One-class novelty detection using GANs with constrained latent representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019.

E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, pages 3942–3951, 2018.

B. Peters, V. Niculae, and A. F. Martins. Sparse sequence-to-sequence models. In *Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, 2019.

H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

H. Pham, Z. Dai, Q. Xie, and Q. V. Le. Meta pseudo labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021.

M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander. Canadian adverse driving conditions dataset. *The International Journal of Robotics Research*, 40(4-5):681–690, 2021.

L. Postman and K. Stark. Role of response availability in transfer and interference. *Journal of Experimental Psychology*, 79:168, 1969.

M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in Neural Information Processing Systems*, pages 3347–3357, 2019.

V. V. Ramaswamy, S. S. Kim, and O. Russakovsky. Fair attribute classification through latent space de-biasing. *arXiv preprint arXiv:2012.01469*, 2020.

R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1): 121–135, 2017.

S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018.

B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning*, pages 5389–5400, 2019.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, and R. E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems*, pages 7959–7970, 2019.

M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

O. Rippel, P. Mertens, and D. Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *International Conference on Pattern Recognition*, 2020.

D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 350–360, 2019.

C. Rosenbaum, I. Cases, M. Riemer, and T. Klinger. Routing networks and the challenges of modular and compositional computation. *arXiv preprint arXiv:1904.12774*, 2019.

F. Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Laboratory, 1961.

A. Rosenfeld and J. K. Tsotsos. Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich. To transfer or not to transfer. In *NeurIPS 2005 Workshop on Transfer Learning*, volume 898, pages 1–4, 2005.

S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard. Latent multi-task architecture learning. In *AAAI Conference on Artificial Intelligence*, pages 4822–4829, 2019.

L. Ruff, R. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4393–4402, 2018.

L. Ruff, R. A. Vandermeulen, B. J. Franks, K.-R. Müller, and M. Kloft. Rethinking assumptions in deep anomaly detection. *arXiv preprint arXiv:2006.00339*, 2020a.

L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020b.

L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, 2010.

K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. Semi-supervised domain adaptation via minimax entropy. In *International Conference on Computer Vision*, pages 8050–8058, 2019.

B. Saleh and A. Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015.

H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust ImageNet models transfer better? In *Advances in Neural Information Processing Systems*, pages 3533–3545, 2020.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2488–2498, 2018.

R. T. Schirrmeister, Y. Zhou, T. Ball, and D. Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. In *Advances in Neural Information Processing Systems*, pages 21038–21049, 2020.

T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

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

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. Technical Report MSR-TR-99-87, Microsoft Research, 1999.

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations*, 2014.

A. Shafahi, P. Saadatpanah, C. Zhu, A. Ghiasi, C. Studer, D. Jacobs, and T. Goldstein. Adversarially robust transfer learning. In *International Conference on Learning Representations*, 2020.

S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2019.

N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

A. Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

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

Y. Shu, Z. Kou, Z. Cao, J. Wang, and M. Long. Zoo-tuning: Adaptive transfer from a zoo of models. In *International Conference on Machine Learning*, pages 9626–9637, 2021.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

S. Sinha, S. Ebrahimi, and T. Darrell. Variational adversarial active learning. In *International Conference on Computer Vision*, pages 5972–5981, 2019.

K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister. Learning and evaluating representations for deep one-class classification. In *International Conference on Learning Representations*, 2021.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 2014.

T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132, 2020.

I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(Feb):211–232, 2005.

A. C. Stickland and I. Murray. BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning*, pages 5986–5995, 2019.

A. Storkey. When training and test sets are different: characterizing learning transfer. *Dataset Shift in Machine Learning*, 30:3–28, 2009.

W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018.

S. Sun, H. Shi, and Y. Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92, 2015.

X. Sun, R. Panda, and R. Feris. AdaShare: Learning what to share for efficient deep multi-task learning. *arXiv preprint arXiv:1911.12423*, 2019a.

Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019b.

R. Tachet des Combes, H. Zhao, Y.-X. Wang, and G. J. Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33:19276–19289, 2020.

J. Tack, S. Mo, J. Jeong, and J. Shin. CSI: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, pages 11839–11852, 2020.

Y. Tamaazousti, H. Le Borgne, C. Hudelot, M. E. A. Seddik, and M. Tamaazousti. Learning more universal representations for transfer-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

D. M. Tax and R. P. Duin. Support vector data description. *Machine Learning*, 54(1): 45–66, 2004.

D. M. Tax and K.-R. Müller. Feature extraction for one-class classification. In *Artificial Neural Networks and Neural Information Processing*, pages 342–349. Springer, 2003.

S. Thrun and L. Pratt. Learning to learn, 1998.

T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pages 37–55. Springer, 2017.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011.

F. Tramer and D. Boneh. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems*, pages 5866–5876, 2019.

A. T. Tran, C. V. Nguyen, and T. Hassner. Transferability and hardness of supervised classification tasks. In *International Conference on Computer Vision*, pages 1395–1405, 2019.

V. Tresp. Mixtures of Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 654–660, 2001.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *International Conference on Machine Learning*, pages 1349–1357, 2016a.

D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016b.

D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017.

G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11(1):1–14, 2020.

L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool. Multi-task learning for dense prediction tasks: A survey. *arXiv preprint arXiv:2004.13379*, 2020.

V. Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, pages 831–838, 1992.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

A. Veit and S. Belongie. Convolutional networks with adaptive inference graphs. In *European Conference on Computer Vision*, 2018.

T. Veniat, L. Denoyer, and M. Ranzato. Efficient continual learning with modular networks and task-driven priors. In *International Conference on Learning Representations*, 2021.

H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pages 1–12, 2019.

M. Wang, Y. Panagakis, P. Snape, and S. P. Zafeiriou. Disentangling the modes of variation in unlabelled data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2682–2695, 2017.

X. Wang, Z. Cai, D. Gao, and N. Vasconcelos. Towards universal object detection by domain attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7289–7298, 2019.

Z. Wang, K. Qinami, I. C. Karakozis, K. Genova, P. Nair, K. Hata, and O. Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8919–8928, 2020.

W.-K. Wong, A. W. Moore, G. F. Cooper, and M. M. Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In *International Conference on Machine Learning*, pages 808–815, 2003.

Y. Wu and K. He. Group normalization. In *European Conference on Computer Vision*, pages 3–19, 2018.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*, pages 418–434, 2018.

S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

S. Xie, Z. Zheng, L. Chen, and C. Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 5423–5432, 2018.

C. Xiong, S. McCloskey, S.-H. Hsieh, and J. J. Corso. Latent domains modeling for visual domain adaptation. In *AAAI Conference on Artificial Intelligence*, pages 2860–2866, 2014.

L. Xu, M. I. Jordan, and G. E. Hinton. An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems*, pages 633–640, 1994.

L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems*, pages 1537–1544, 2005.

R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *European Conference on Computer Vision*, 2014.

Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2142–2150, 2015.

D.-Y. Yeung and C. Chow. Parzen-window network intrusion detectors. In *Object recognition supported by user interaction for service robots*, volume 4, pages 385–388. IEEE, 2002.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014.

R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017a.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017b.

W. Zhang, L. Deng, L. Zhang, and D. Wu. Overcoming negative transfer: A survey. *arXiv preprint arXiv:2009.00909*, 2020.

Y. Zhang and Q. Yang. An overview of multi-task learning. *National Science Review*, 5 (1):30–43, 2018.

H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, pages 8559–8570, 2018.

J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *Empirical Methods in Natural Language Processing*, 2017.

C. Zhou and R. C. Paffenroth. Anomaly detection with robust deep autoencoders. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.

K. Zhou, Y. Yang, T. Hospedales, and T. Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI Conference on Artificial Intelligence*, pages 13025–13032, 2020.

K. Zhou, Y. Yang, Y. Qiao, and T. Xiang. Domain generalization with MixStyle. In *International Conference on Learning Representations*, 2021.

Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu. Traffic-sign detection and classification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2110–2118, 2016.

B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.