Not Only the Last-Layer Features for Spurious Correlations: All Layer Deep Feature Reweighting

Humza Wajid Hameed

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Computer Science (Computer Science) at

Concordia University

Montréal, Québec, Canada

April 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Humza Wajid Hameed**

Entitled: **Not Only the Last-Layer Features for Spurious Correlations: All Layer Deep Feature Reweighting**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Charalambos Poullis

_____ Examiner
Dr. Ching Yee Suen

_____ Supervisor
Dr. Eugene Belilovsky

Approved by _____
Joey Paquet, Chair
Department of Computer Science and Software Engineering

_____ 2025

_____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Not Only the Last-Layer Features for Spurious Correlations: All Layer Deep Feature Reweighting

Humza Wajid Hameed

Spurious correlations are a major source of errors for machine learning models, in particular when aiming for group-level fairness. It has been recently shown that a powerful approach to combat spurious correlations is to re-train the last layer on a balanced validation dataset, isolating robust features for the predictor. However, key attributes can sometimes be discarded by neural networks towards the last layer. In this work, we thus consider retraining a classifier on a set of features derived from all layers. We utilize a recently proposed feature selection strategy to select unbiased features from all the layers. We observe this approach gives significant improvements in worst-group accuracy on several standard benchmarks. Another pain point in transfer learning is with out-of-distribution tasks having large distribution shifts relative to the source task. Full finetuning suffers in performance as it disturbs backbone parameter weights during the starting few optimization steps and is forced to make drastic adaptations to correct for large losses initially observed in training. Linear tuning is another approach shown to improve model generalization capabilities and is especially effective for transfer learning on out-of-distribution downstream tasks. We further evaluate the usefulness of intermediate layer information by incorporating it with a linear tuning approach. Results over datasets from a common visual task adaptation benchmark show that the empirical benefits from simply leveraging intermediate layers are similar to the proposed method and there is no noticeable gain in accuracy from incorporating a linear tuning step.

Acknowledgments

With the completion of my thesis, I would like to thank the people who made this possible: My family for always pushing me to continue my studies despite some challenging times, Geraldin Nanfack for taking the time to guide me through technical and theoretical parts that seemed out of my level of expertise and of course my supervisor Prof. Eugene Belilovsky for his patience, understanding of my difficulties outside academia and support throughout my studies. Coming from a theoretical mathematics background, a master of computer science seemed like an out of reach goal but Prof. Eugene Belilovsky always provided me with guidance by advising on courses to take and connecting me with individuals who could help fill any gaps in knowledge that I had. I would also like to thank Concordia University for all the resources that helped me in my journey. Being the place where I completed both my undergraduate studies as well as this master of computer science, I will surely remember my time here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Thanks to their performance capability, deep learning is increasingly applied across diverse domains including healthcare. However, when trained with empirical risk minimization (ERM), deep learning models may fail to learn stable features, which are those that hold across data distributions collected at different times and places Shah, Tamuly, Raghunathan, Jain, and Netrapalli (2020). For instance, it has been observed the tendency of convolutional neural networks (convnets) to often prioritize image texture over more informative features such as shapes, which may be better predictors Geirhos et al. (2020); Hermann, Chen, and Kornblith (2020). This tendency arises from models' ability to exploit any shortcuts or spurious correlation present in training data which may be sufficient to correctly predict training data, but may not hold in unseen test data. As a result, this exposure to learning spurious correlation or any shortcut in data makes them vulnerable to a potential drop in predictive performance.

Addressing the challenge of learning in the presence of such spurious correlations has motivated several approaches in the literature. It is widely adopted the notion of *groups* (defined in Sec. 2), in which to correctly classify instances from certain groups, it is expected that models should be robust enough to spurious correlation. Assuming the presence of group annotations in training data, a well-known alternative to ERM is group distributionally robust optimization (group DRO) Sagawa, Koh, Hashimoto, and Liang (2020a), which minimizes the empirical worst-group risk. There exist methods that do not assume group information during training such as just train twice (JTT) E. Z. Liu

et al. (2021), which divides the training time into two phases. The first phase trains the model with ERM, while the second phase continues by up-weighing the loss of misclassified instances of the first phase. Recently, it has been shown that the last-layer representations of ERM-trained models already exhibit both robust and non-robust (to spurious correlation) features Izmailov, Kirichenko, Gruver, and Wilson (2022); Kirichenko, Izmailov, and Wilson (2022). As a remedy, to decrease the impact on non-robust features, deep feature reweighting (DFR) Kirichenko et al. (2022) has proven effective, which only retrains the classifier with a balanced-group validation set.

DFR can be viewed as an instantiation of transfer learning, where there is a desire to exploit robust and generalizing features from the source domain to build a good predictive model on the target domain Tan et al. (2018). Here, the goal of the target domain is a dataset balanced according to groups. In transfer learning literature, more advanced methods exist beyond retraining the classifier. Indeed during supervised learning a network can learn to discard certain robust features present from earlier layers to make the final prediction in the last layer, thus losing potentially useful features. This paper considers a simple yet efficient transfer learning method, called Head2Toe Evci et al. (2022). Unlike last-layer retraining, Head2Toe leverages all layer features, not just the last one, to find a sparse network with the most transferable features. Therefore, we leverage Head2Toe to complement DFR and aim to get the most transferable features while also decreasing the impact of non-robust (to spurious correlation) features.

1.2 Contributions

For this thesis, we look to provide a robust model able to extract features from intermediate layers of a neural network while also reducing its spurious feature bias to be performant on any target task. Our contributions are summarized as follows

(1) We show how an efficient transfer learning method (**Head2Toe** here) can be incorporated in a pipeline of a state-of-the-art method in spurious correlation learning (**Deep Feature Reweighting**).

(2) We demonstrate that this incorporation can yield better performance on standard evaluation benchmarks

(3) We combine **Head2Toe** with ideas from another study regarding efficient transfer learning and explain the shortcomings of this method

The contributions of Chapter 3 have been presented at the ECCV 2024 *Fairness and ethics towards transparent AI: facing the chalLEnge through model Debiasing (FAILED)* workshop on July 31, 2024: *Not Only the Last-Layer Features for Spurious Correlations: All Layer Deep Feature Reweighting* by Humza Wajid Hameed, Geraldin Nanfack, Eugene Belilovsky. (Hameed, Nanfack, and Belilovsky (2024))

Chapter 2

Background and Related Work

2.1 Background

This chapter serves as a reference to detail spurious correlations, transfer learning, as well as relevant works that were studied and considered throughout our efforts. Several approaches have been developed to address the challenge of learning amidst spurious correlations, alongside various methods aimed at efficient transfer learning. All of these methods align with the unique goal of achieving robustness by employing techniques that lead to a model that can generalize over varying and unseen environments.

2.1.1 Convolutional Neural Networks

A fundamental component of this study are Convolutional Neural Networks (CNNs), a subset of neural networks in deep learning often leveraged for computer vision and image processing tasks. The ability of CNNs to effectively learn spatial patterns or features in images is one of the reasons they are frequently at the core of many vision architectures. Traditional CNN architectures consist of convolutional layers that apply filters to inputs, creating feature (or activation) maps that capture various aspects of the image; pooling layers that reduce the dimensionality of the input image as it passes through different layers, thereby reducing the number of parameters and decreasing computation load; and fully-connected layers that process the learned features from previous layers into a prediction (O'Shea and Nash (2015)). CNNs are renowned for their ability to extract

relevant features from data, making them a critical component of this study. Our proposed method emphasizes the importance of feature selection, further motivating a deeper exploration of CNNs. To illustrate the strength of CNNs, Alabsi, Anbar, and Rihan (2023) presents a *CNN-CNN* method to detect Internet of Things (IoT) attacks. Given the vast number of connected devices, detecting attacks from network traffic is a significant challenge addressed by this paper through the use of two CNN models. Following thorough preprocessing, the first CNN model extracts relevant features, which are then fed into a second model to determine IoT attacks from network traffic. This approach highlights the importance of feature extraction and demonstrates the effectiveness of CNNs in independently identifying critical details in data. By leveraging the *CNN-CNN* method, the study showcases how CNNs can autonomously learn and select pertinent features. It provides validation of using CNNs as an intermediary feature selection step (similar to our proposed method having a separate feature selection step) in improving the performance of deep learning models, particularly in complex scenarios involving large-scale data. CNNs directly relevant to this study are residual networks, a residual learning framework presented in He, Zhang, Ren, and Sun (2015). The paper explains that following the discovery of the value of depth when constructing network architectures, one of the problems encountered was oversaturation and degradation of accuracy. Their *deep residual learning framework* addresses this through *shortcut connections*. By using blocks consisting of stacked layers, the original input is added to the block output. Instead of fitting the mapping $M(x)$, it fits the residuals and adds back the input, which they explain is easier to optimize:

$$R(x) = M(x) - x \tag{1}$$

For instance, the ResNet-50 (He, Zhang, Ren, and Sun (2016)) model employed in our method, is a residual network consisting of 50 varying layers including convolution, batch normalization, activation, pooling and fully connected layers. They further emphasize that this framework can easily be extended by increasing depth for improved accuracy.

2.1.2 Learning with Spurious Correlation

To better illustrate concepts around learning in the presence of *Spurious Correlation*, we consider the Waterbirds dataset (Sagawa et al. (2020b)). Given images of waterbirds and landbirds, an evident observation is that waterbirds, in a large portion of the dataset, appear on water whereas landbirds are seen on land. This relationship, although prevalent throughout the dataset, is not deterministic of the bird class. *Spurious correlations* refer to associations between elements consistently observed along with the variable of interest. In the case of the Waterbirds dataset, the spurious attribute is the background, and the target variable would be the class of bird. Images of landbirds on water and waterbirds on land are scarcely present in the dataset and are considered minority groups. Without consideration of spurious features, a model may build a bias on a water background to recognize waterbirds and a land background to recognize landbirds instead of identifying the bird class itself. Without consideration of spurious attributes, learning models may build a bias towards unstable features and struggle to correctly classify images in minority groups, as well as in completely different environments.

(a) Cardinal in a forest

(b) Cardinal on a beach

Figure 2.1: **Left**: A cardinal (landbird) from the *waterbirds* dataset (Sagawa et al. (2020b)), often observed by the model in its natural land habitat is correctly classified as a landbird. **Right**: A cardinal with a beach background, scarcely observed in such settings is incorrectly classified as a waterbird when the spurious feature (background) is disregarded during training

Data augmentation techniques appear to be the standard approach to fight against minority groups Agarwal, Shetty, and Fritz (2020); Chen et al. (2020); Shetty, Schiele, and Fritz (2019).

For example, Plumb, Ribeiro, and Talwalkar (2022) introduces a data augmentation technique by generating counterfactual data, which adds or removes object parts responsible for identified spurious patterns. There exist methods that analyze representations throughout the training dynamics to understand how the bias to spurious features arises. For instance, Dreyer, Pahde, Anders, Samek, and Lapuschkin (2023) propose a method to reduce model reliance on spurious features by penalizing the gradient in directions of spurious features. Several other works have been done to analyze stochastic gradient descent (SGD) directions, leading to modified versions of SGD or loss functions Nagarajan, Andreassen, and Neyshabur (2020); Pezeshki et al. (2021); Rahaman et al. (2019). Although most of these methods address the spurious correlation problem, it has been recently shown the ability of ERM to competitively learn robust-to-spurious-correlation features Kirichenko et al. (2022); Zong, Yang, and Hospedales (2022). Despite the use of data with labeled spurious features in DFR, it differs from the above works by directly using balanced data where each grouping of class label and spurious feature are equally represented, resulting in a decrease in bias toward spurious or unstable features.

2.1.3 Efficient Transfer Learning

Transfer learning leverages a model initially trained in a specific setting and later repurposed for a new but related task of interest. Although the issue of spurious correlations in machine learning models has been addressed, a crucial element in this study is the ability to achieve efficient transfer learning. While spurious correlations can be mitigated to reduce bias towards non-predictive elements, there remains a need to finetune the model's hyperparameters and backbone to optimize performance on new, unseen tasks. This necessitates the development of sophisticated transfer learning techniques that can generalize knowledge from one domain to another, thereby improving the model's adaptability and robustness.

Transfer learning still has its own challenges that are addressed across various works, and there are differing techniques established to improve downstream task performance. Yazdanpanah et al. (2022) presents a method that consists of training affine parameters in batch normalization layers. It explains how training these shift and scale parameters used in the normalization step can have a noticeable impact, especially in scarce data settings. Downstream tasks where there is minimal

divergence in distribution from the pre-trained model seldom have difficulty learning on a new dataset when employing finetuning. This assumes that the pre-trained model is otherwise sufficient for the downstream task and improvements can be attained through leveraging batch normalizations alone. Kumar, Raghunathan, Jones, Ma, and Liang (2021) explains that these in-distribution tasks struggle most when exposed to data with significant deviations in distribution from the source task. It explains that with a randomly initialized linear head, finetuning results in larger fluctuations in parameter weights during the earlier training steps as the whole network needs to adjust to what it considers an unusual dataset relative to the source task. First keeping the pretrained backbone frozen and training the linear head before unfreezing and finetuning allows the model to better adapt and transfer to out-of-distribution downstream tasks. Here, instead of drawing focus to just one layer, the assumption is that a better initialization of the classifier will lead to improved finetuning on the downstream task. Qiu, Potapczynski, Izmailov, and Wilson (2023) deals with the out-of-distribution transfer learning problem through the use of a weighted loss function. It proposes Automatic Feature Reweighting (AFR), a method that reweighs the loss among groups with fewer examples to push the model to better adjust and adapt for these minority group examples. This method has the additional benefit of not relying on spurious features. This approach leans similar to the DFR method by rebalancing groups to avoid unstable features from dominating the learning process although it does it through a loss function instead of group counts. With these different approaches to efficient transfer learning, the motivation for our method is derived from the use of Head2Toe's unique feature selection process. Unlike the above-mentioned methods, Head2Toe focuses on searching for information found throughout the network instead of relying solely on the penultimate layer. Head2Toe extracts useful intermediate layer features Evci et al. (2022) and concatenates them to create a linear layer. This newly initialized linear layer, combined with Deep Feature Reweighting (DFR) Kirichenko et al. (2022), is trained on an unseen balanced dataset under our approach. With Head2Toe improving feature extraction by incorporating early layer information and DFR reducing unstable feature bias through balanced data training, our method is motivated to address challenges in efficient transfer learning.

2.1.4 Finetuning

Finetuning (**FT**) methods, a subset of transfer learning, involve adjusting parameter weights throughout a pre-trained neural network to effectively adapt to a new domain. This process is predicated on the assumption that a model can be efficiently tuned and adapted to various different but related tasks. As discussed previously, existing methods often face difficulties in adapting to downstream tasks when the target task distribution sufficiently diverges from the source data used to pre-train the network. Although **FT** can provide improvements in results over time, the cost of fully tuning a model with the classical approach remains significant. This issue is particularly pronounced with large models, such as language models. As shown in Zhai et al. (2020), full finetuning generally outperforms the simpler transfer learning method, referred in this thesis as **NaiveTL**, where only the output head weights are adapted to the target task domain using some pretrained model. Despite this notable improvement in performance, finetuning requires storing and updating the weights through the entire network at every iteration during training, making this a costly solution for larger models. To address the high computation costs observed in classical FT, Lv et al. (2024) introduces LOwMemory Optimization (LOMO). Through an improved optimizer, gradient computations and parameter updates are done simultaneously to reduce memory costs, distinguishing it from traditional methods by not requiring all parameter gradients to be kept in memory at once.

2.1.5 Parameter-Efficient Fine-Tuning (PEFT)

This then motivates the need for less costly methods that do not require fully tuning a network's weights. An alternative to **FT** is Parameter-Efficient Fine-Tuning (**PEFT**). Classical and modern **PEFT** methods provide an alternative to deal with some of the challenges encountered in the **FT** approaches. **PEFT** methods primarily apply within the context of large language models as they generally require much more computational efficiency. Despite this, these methods can provide insight on approaching vision problems as well. In the context of this paper, a part of the method presented requires training a processed concatenation of all layers, which increases the complexity

and computational requirements. Xu, Xie, Qin, Tao, and Wang (2023) explains the memory efficiency of **PEFT** methods, which significantly reduce the number of parameters to be tuned instead of potentially tuning all the billions of parameters tuned in **FT**. Low-Rank Adaptation (**LoRA**), presented in Hu et al. (2021) is a classical **PEFT** method in which trainable parameters are drastically reduced. **LoRA** is able to achieve results comparable to **FT** by freezing the weight matrix, W_0, and redefining it as:

$$W_0 + B \cdot A \tag{2}$$

with lower-rank matrices B and A as opposed to solely relying on matrix W_0. By rewriting the weight matrix in this manner, W_0 can be updated by training the lower-rank matrices B and A, resulting in reduced computation costs. **Sparse-Tuning**, as shown in T. Liu et al. (2024), is another **PEFT** approach adapted to vision problems. By merging less informative pieces of the input image in each layer, **Sparse-Tuning** eventually produces a layer where uninformative pieces of the image are masked, and the focus is shifter to parameters representing elements essential for image recognition. In the context of the Waterbirds dataset, this approach could involve combining a large portion of the water or land background, thereby effectively reducing the focus on these non-predictive parts of the image. Merging the tokens for those pieces of the image can reduce the amount of computation needed and focus can remain on the bird itself. Xin et al. (2024) explores visual **PEFT** methods and categorizes these into three main categories that are further segmented into sub-categories. These three categories are as such: addition-based approaches that essentially introduces additional trainable elements to achieve parameter efficiency, partial-based methods that focus on training a fraction of parameters while keeping the rest of the network parameters and structures the same, and unified-based methods that combine different tuning techniques into a framework capable of leveraging the different tuning methods into a single architecture. Xin, Du, Wang, Lin, and Yan (2023) is a **PEFT** method adapted to vision tasks by presenting a Vision Multi-Task Adapter (**VMT-Adapter**) that facilitates interactions between different tasks and allows for efficient multi-task adaption. A parameter-sharing scheme is also used to reduce the total trainable parameters and improve computation load. The aforementioned **PEFT** methods all provide insight into potential deviations and directions from our proposed method as the feature selection process

is a computation heavy process.

2.1.6 Domain Generalization

At the core of this study lies the objective of developing a system capable of adapting to diverse, unseen environments. Achieving a robust model necessitates the incorporation of domain generalization concepts. Our method demonstrates the ability to learn in the presence of spurious correlations and generalize across various unseen tasks by minimizing the focus on elements that may be prevalent in the target dataset but should not significantly influence the model's decision-making process. With this in mind, this section details relevant domain generalization methods. Sagawa et al. (2020b) minimizes the worst-case training loss to push the model towards performing even in scarce data groups with their regularized group distributionally robust optimization (**group DRO**) method. **Group DRO** assumes there is a label for each input regarding their spurious attribute (e.g. an image of a waterbird will have target label 'waterbird' and spurious attribute 'water', identifying the background of the target) and balances the learned features weights for robustness during training. **Group DRO** is able to outperform ERM when applying regularization with **group DRO**. The paper explains that **group DRO** on its own is not sufficient to outperform ERM. Using a stronger regularizer is what ultimately produces performing results. With a stronger regularizer, the model no longer converges to a loss near 0 and worst-group accuracies improve significantly under such regularization.

As shown in, Arjovsky, Bottou, Gulrajani, and Lopez-Paz (2020), another way to near domain generalization is generating a predictor invariant of the environment it is exposed to. This is similar to what we observe in learning with the presence of spurious correlations, where the aim is to remove bias towards any spurious components often observed but not predictive of the target itself. The developed method, Invariant Risk Minimization (**IRM**), suggests searching for a representation of the data which, upon training, results in the same classifier across different environments. In essence, by producing such a data representation, regardless of the environment, the optimal predictor will remain consistent. This new representation will always result in a classifier invariant to the environment it is exposed to. By putting emphasis on training with an improved data representation, the model is less likely to form bias towards spurious features and is better able to generalize on

out-of-distribution tasks or unseen environments.

2.1.7 Group-Lasso Regularization

Regularization is a technique used in machine learning to reduce complexity and variance by penalizing and shrinking large parameter weights. It is often needed to prevent overfitting and there are many studies that extend this idea through various methods. *Sparse Feature Selection* involves extracting a subset of relevant features from a high-dimensional dataset. Through regularization techniques, we are able to achieve sparse features by reducing focus towards noisy or unimportant features. Following Scardapane, Comminiello, Hussain, and Uncini (2017), we first introduce the *Standard Lasso* penalty and then cover the more relevant regularizer for our study's feature selection process, *Group-Lasso*.

Standard Lasso minimizes a loss function subject to parameter weights β and shrinks them by applying a penalty. *Group-Lasso* (Yuan and Lin (2006)) is an extension of *Standard Lasso*, as a method to introduce sparsity in the features of the weights matrix. Each feature, represented by a row of the weights matrix is a grouping g. The advantage of *Group-Lasso* being that by grouping weights specific to each row (or feature) in the weights matrix, during training, regularization will penalize entire features (through their weights) and result in a more thorough feature selection process. By penalizing by group instead of individual weights, all weights related to an irrelevant feature will be shrunk together and all weights grouped for relevant features will be kept with more defined weights. The following equations show how *Standard Lasso* (3) aims to minimize weights, regardless of which feature they pertain to, whereas *Group-Lasso* (5) applies group-wise regularization. The selected features in *Group-Lasso* will have more pronounced weights in their parameter weights vector β_g

$$\text{Standard Lasso:} \quad \min_{\beta} \left\{ L(X, y; \beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{3}$$

$$\text{L2-norm:} \quad \|\beta_g\|_2 = \sqrt{\sum_{j \in g} \beta_j^2} \tag{4}$$

$$\text{Group-Lasso: } \min_{\beta} \left\{ L(X_g, y; \beta_g) + \lambda \sum_{g=1}^{G} \sqrt{|\beta_g|} \|\beta_g\|_2 \right\} \tag{5}$$

Where:

- $L(X, y; \beta)$ is the chosen loss function (often cross-entropy loss for classification settings).
- X is the matrix of inputs.
- y is the vector of target variables.
- β is the vector of coefficients (parameter weights).
- λ is the regularization parameter.
- G is the number of groups (or features).
- p is the total number of parameter weights.
- $|\beta_j|$ is the absolute value of the coefficient for the j-th feature
- β_g is the vector of coefficients for the g-th group (feature).
- $|\beta_g|$ is the number of weights for the g-th vector β_g.
- $\|\beta_g\|_2$ is the $L2$-norm of the coefficients for the g-th group.

2.1.8 Fairness

The discussion on learning with spurious correlations, transfer learning and domain generalization, necessitates detailing fairness within the context of AI systems. As detailed in Mehrabi, Morstatter, Saxena, Lerman, and Galstyan (2022), fairness becomes a concern when AI systems exhibit discriminatory behavior due to biases developed during training. The paper explains that many of these tools, developed for real-world applications, may often use data that reflects certain demographics, groups, or individuals. It is crucial to ensure they do not disproportionately disadvantage any particular group.

Zafar, Valera, Gomez-Rodriguez, and Gummadi (2019) provides a framework to limit these biases and discusses some fairness notions that may be useful in building less discriminative systems: *Disparate treatment* necessitates decisions to be independent of sensitive attributes such as race or gender; *Disparate mistreatment* where outcomes may differ because there may be different rates of false positives and false negatives for individuals within a group who have a different sensitive feature.

In the context of this thesis, fairness can be tied to the spurious correlation problem. One of our goals is to limit inherent bias towards unstable features and use the right metrics to evaluate results, striving towards a fair model. With the spurious attribute being labeled in a dataset, it can be used to create groups with the target variable. For example, in the *Waterbirds* dataset, there were four groups: land birds on land, land birds on water, water birds on land, and water birds on water. The background serves as the spurious label. A large amount of the data consists of birds in their natural habitat, but it also includes samples of birds outside their usual location. These small-sized groups are considered *minority groups* and may be incorrectly classified if models do not consider the spurious attribute and have inherent bias based on the type of background. To improve fairness, we employ *worst-group accuracy (WGA)* as a measure to quantify changes in the accuracy of minority and majority groups.

Chapter 3

Head2Toe-DFR

3.1 Related Works

Here we provide context and discuss works that were directly employed in our proposed method. We consider a classification problem with a training set denoted by $\mathcal{D}_{\text{Tr}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathcal{X}$ is the input and $y_i \in \mathcal{Y}$ is its class label. For each data $\mathbf{x}_i$, there is a spurious attribute value $a_i \in \mathcal{A}$ of $\mathbf{x}_i$, where a_i is non-predictive of y_i, and $\mathcal{A}$ denotes the set of all possible spurious attribute values. We denote by a group the pair $g := (y, a) \in \mathcal{Y} \times \mathcal{A} := G$. Our goal is to learn a parameterized model [1]: $f_\theta : \mathcal{X} \longrightarrow \mathcal{Y}$ that will maximize the expected accuracy and crucially the worst performing group's accuracy while avoiding learning spurious features. The worst-group in terms of accuracy is often one scarcely present in the dataset. This means the model has not been exposed to enough inputs from the group to adapt to its setting and may even have built biases towards non-predictive elements frequently observed in the majority groups (groups with a larger sample size throughout the dataset). This is why by improving upon worst-group accuracies, we produce a more generalized model, capable of adapting to other unseen environments. We consider a neural network model

$$f_\theta(.) = \gamma\left(h(.)\right), \tag{6}$$

where γ denotes its classifier layer and h denotes its feature network. Additionally, we denote by $\mathcal{D}_{\text{Val}}$ and $\mathcal{D}_{\text{Te}}$ the validation and test sets, respectively. Assuming that we have the information of

[1] To simplify, we can omit subscripting parameters θ.

groups on a sample $\mathcal{D}$, we denote by $\mathcal{D}^{\text{RW}}$ a balanced subset of $\mathcal{D}$ in which groups are uniformly distributed, i.e., each group has the same number of examples.

3.1.1 Deep Feature Reweighting

Deep Feature Reweighting (DFR) Kirichenko et al. (2022) is a technique designed to mitigate the influence of spurious features—the previously discussed non-predictive features that are highly correlated with training targets but do not contribute meaningfully to the model's generalizability.

DFR Training Phases.

DFR training involves two phases:

(1) **Initial Training Phase**: Initially, the model undergoes ERM using the unbalanced training data ($\mathcal{D}_{\text{Tr}}$), without information about the spurious attribute. During this phase, the model learns both robust and non-robust features due to spurious correlations in the unbalanced dataset. Since $\mathcal{D}_{\text{Tr}}$ is significantly larger than the balanced validation set ($\mathcal{D}_{\text{Val}}^{\text{RW}}$), it serves as the primary learning dataset, allowing the model backbone to effectively adapt parameter weights to the target task despite developing a bias towards spurious features.

(2) **Feature Reweighting Phase**: In this phase, the feature network (h) is frozen, and the classifier (γ) is retrained on a balanced validation set $\mathcal{D}_{\text{Val}}^{\text{RW}}$. Here, spurious attribute labels are used to create groups ($g := (y, a)$), ensuring each group contains an equal number of samples. Kirichenko et al. (2022) emphasize that retraining the last layer alone on $\mathcal{D}_{\text{Val}}^{\text{RW}}$ is a cost-effective step that achieves results comparable to other techniques addressing spurious learning. While the first phase facilitates the learning of important robust features, the reweighted data training phase reduces reliance on unstable features.

The key takeaway from DFR is that a small validation set with spurious attribute labels is sufficient for retraining purposes to reduce the inherited bias from ERM.

3.1.2 Head2Toe

Head2Toe Evci et al. (2022) is the efficient transfer learning method used as part of our proposed model. It aims to select the most useful features from intermediate layers that better transfer

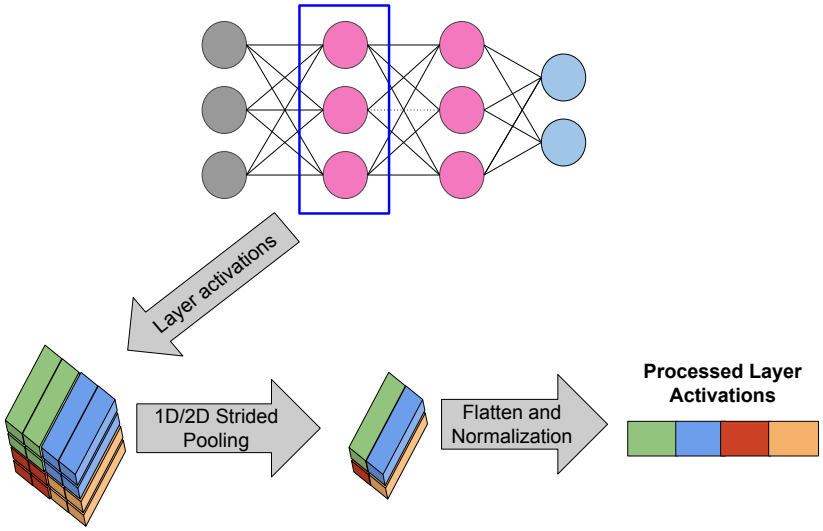

Figure 3.1: **Layer activations processing.** Each layer activations go through 1D/2D strided average pooling, flattening and normalization. The result is then concatenated with similar results from other layers (Evci et al. (2022)

)

to a target or downstream task. Initially, as there is no insight on which features to select, **Head2Toe** concatenates all processed feature maps (activations) from intermediate layers. The processing for the activations in each layer (3.1) is 1D or 2D average strided pooling followed by flattening and normalization to retain feature magnitudes and differences between layers (Evci et al. (2022)). Following this processing for all features, the resulting vectors are concatenated to produce h_{AF}. This serves as a basis of the set of features to filter through until those deemed most important are identified. Denoting the corresponding parameter weights for h_{AF} as W_{AF}, the paper explains that this will be an immense weights matrix which can be quite expensive to tune.

This is where **Head2Toe** levarages the aforementioned *Group-Lasso Regularization* (2.1.7) . Here, the L2-norm (4) will be the score for each row (feature) in W_{AF} and will be the source for ranking features in terms of importance. Using a threshold τ, **Head2Toe** can select the most useful features (highest score rankings) that better transfer to the downstream task. It has been shown that this method for transfer learning usually performs better than only retraining the linear head. **Head2Toe** emphasizes it is especially beneficial for tasks differing significantly from the source

task. These are labeled as low domain affinity tasks.

$$\text{DomainAffinity} = \text{Acc}_{\text{NaiveTL}} - \text{Acc}_{\text{S}} \tag{7}$$

Where:

- $\text{Acc}_{\text{NaiveTL}}$ is the accuracy of training a new output head onto a pretrained model (2.1.4)
- Acc_{S} is the accuracy of training the model from scratch

A low domain affinity would suggest that the target task is noteably out-of-distribution relative to the source task. This implies training the model from scratch is resulting in better results than using a pre-existing feature embedding and training a new output head on it. This then implies that the target task features are largely different than the pretrained ones. The gain in performance from **Head2Toe** in out-of-distribution settings shows the usefulness of searching for intermediate layer features during the feature extraction process. In the following sections, we denote by h^{H2T} the feature network found by the **Head2Toe** method.

3.2 Method

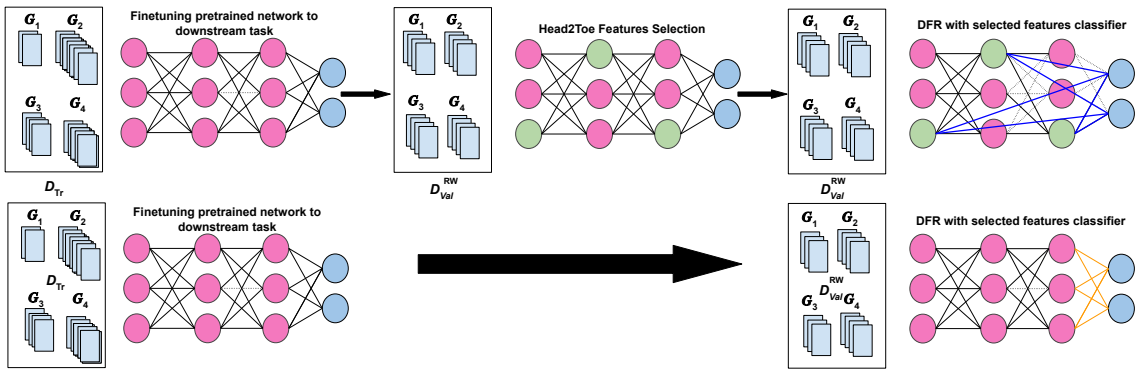

Figure 3.2: **Top Row - H2T-DFR** Illustration of the different training phases for H2T-DFR. A pre-trained network is tuned on a target task using unbalanced data followed by Head2Toe feature selection with balanced training data. Balanced data consists of equal counts of group G_i, where each group is a unique combination of target and spurious features. Lastly, a classifier comprised of selected features is trained on an unseen validation dataset. **Bottom Row - DFR** Illustrates the DFR baseline method in comparison, which excludes any feature selection

Algorithm 1 Spurious Head2Toe (**H2T-DFR**)

Input: training, validation, testing sets: $\mathcal{D}_{\text{Tr}}$, $\mathcal{D}_{\text{Val}}$, $\mathcal{D}_{\text{Te}}$; percentage of selected features τ; number of epochs T_1 and T_2 and T_{DFR}. ***Finetuning for a downstream task***Initialize model with pre-trained model $f = \gamma \circ h$ Finetune γ and h on $\mathcal{D}_{\text{Tr}}$ for T_1 epochs

Feature Selection

Initialize the Head2Toe model f_{H2T} with classifier consisting of all layers Get the set of most important features $\mathcal{H}_{\mathcal{FS}}$ using group lasso Evci et al. (2022) on $\mathcal{D}_{\text{Val}}^{\text{RW}}$ with T_2 epochs (see Section 3.2)

Deep Feature Reweighting: Balanced data retraining

Initialize the f_{H2T} through h_{H2T} with only selected features $\mathcal{H}_{\mathcal{FS}}$ Freeze h_{H2T} and retrain the classifier g_{H2T} on $\mathcal{D}_{\text{Val}}^{\text{RW}}$ for T_{DFR} epochs

This section describes our method H2T-DFR, which is summarized in Figure 3.2 and Algorithm 1. It consists of three phases detailed below: (1) unbalanced training (or finetuning) on all the training data (2) balanced data feature selection and finally using the features for (3) balanced data linear classifier training.

Unbalanced Finetuning. As in the DFR approach our method starts by simple fine-tuning on the entire dataset using ERM on $\mathcal{D}_{\text{Tr}}$, denoting this trained backbone as $h_{\text{pretrained}}$.

Balanced Feature Selection and Classifier Training. We follow an approach similar to Head2Toe: we concatenate all the features from all layers and perform a group lasso feature selection. Using $h_{\text{pretrained}}$, we initialize a Head2Toe model which will have the linear layer g_{H2T}. It should be noted that g_{H2T} in this stage is a linear classifier layer with inputs from all intermediate features in the network as described in the Head2Toe background section. g_{H2T} is then trained on $\mathcal{D}_{\text{Val}}^{\text{RW}}$ using a group-lasso regularization loss. The regularization allows the model to adjust weights in terms of importance, and the resulting weights can be used to calculate *relevance scores* s_i Evci et al. (2022), where s_i are the l_2 norms of the final weights of each feature of the $h_{\text{pretrained}}$ network. The final features selected correspond to the ones with the top τ percent scores, and the corresponding sparse network is denoted by h_{H2T}.

		Worst-group accuracy	Mean group accuracy
Dataset	Method	Mean (%)	Mean (%)
CelebA	NaiveTL	48.44 ± 1.02	82.79 ± 0.30
	DFR	85.99 ± 0.74	91.58 ± 0.15
	Affine-DFR	85.49 ± 0.70	91.55 ± 0.14
	H2T-DFR (ours)	88.59 ± 0.48	91.87 ± 0.17
Waterbirds	NaiveTL	86.23 ± 1.56	92.90 ± 0.46
	DFR	92.76 ± 0.37	94.54 ± 0.21
	Affine-DFR	89.02 ± 0.37	94.13 ± 0.08
	H2T-DFR (ours)	90.85 ± 0.45	93.51 ± 0.06
HAM10000	NaiveTL	50.11 ± 0.54	75.27 ± 0.52
	DFR	67.31 ± 2.61	78.09 ± 0.91
	Affine-DFR	53.63 ± 2.84	76.72 ± 0.68
	H2T-DFR (ours)	69.69 ± 1.84	78.23 ± 0.46

Table 3.1: Mean and worst-group performance over 5 seeds. Mean group accuracy averages over all group accuracies.

With features selected, our final model is initialized using the sparse network h_{H2T}, which contains only selected features instead of all features from the pre-trained network. We denoted by $f_{\text{H2T}} = g_{\text{H2T}} \circ h_{\text{H2T}}$ the full head2Toe model alongside its classifier. It should be noted that the classifier here, g_{H2T}, is a new linear layer with inputs h_{H2T} features selected in Part 1. The backbone is frozen and g_{H2T} trained on $\mathcal{D}_{\text{Val}}^{\text{RW}}$, where each batch has equal group counts and removes the ability for the model to build a bias towards minority groups. The resulting model is what we denote **H2T-DFR**. The difference here being that although $\mathcal{D}_{\text{Val}}^{\text{RW}}$ has been used for feature selection previously and is composed of equal group counts in each batch, more importantly, it has never been introduced to the model for parameter updates. During our experiments, we observed that a balanced dataset alone does not improve results. It is also required that the dataset has never been previously used to train the model as in DFR.

3.3 Experiments

We now discuss our experimental results. We focus our comparison to DFR which has shown to be a powerful technique that can outperform existing methods such as groupDRO Izmailov et al. (2022). We propose another baseline to compare to DFR and H2T-DFR, specifically **Affine-DFR**. This approach mimics DFR and does not have an explicit feature selection phase. Similar to DFR, **Affine-DFR** uses a balanced training phase but adapts only the affine parameters of batch-norm layers.

Experiments for DFR, Affine-DFR and H2T-DFR were run over 5 seeds and table 3.1 shows their respective performance on CelebA Z. Liu, Luo, Wang, and Tang (2015), Waterbirds Sagawa et al. (2020a), and HAM10000 Hermann et al. (2020); Zong, Yang, and Hospedales (2023). The pre-trained model employed is ResNet-50 He et al. (2016). Hyperparameters were selected to reproduce result from the respective papers employed. For spuriousH2T, there is further hyperparameter tuning done starting from the hyperparameters in table C as there is added complexities such as feature selection fraction and layer target size to consider along with the different learning rates across the 3 training phases. Hyperparameters were originally set according to papers referenced for baseline results and further tuned through a hyperparameter search similar to the DFR paper Kirichenko et al. (2022).

Results. Our results shown in Tab. 3.1 are with the use of a balanced dataset for feature selection. We observe improvement in the worst group accuracy for HAM10000 and CelebA, whereas Waterbirds does not show improvement from H2T features, the result being very close to that of the basic DFR. We note however that this dataset is known to be simpler than the others considered. Indeed, for Waterbirds, it was already observed that ImageNet pre-trained models already contain robust features to easily get $\approx 88\%$ of worst-group accuracy by just retraining the classifier Izmailov et al. (2022). CelebA's worst-group accuracy increased by 2.60% from our method relative to the baseline (DFR). The medical dataset HAM10000 shows a 2.38% increase in worst-group accuracy. With all 3 datasets, the mean across group accuracies did not show a noticeable difference between the 3 methods.

To gain insight into the feature selection step, we illustrate the features selected depending on

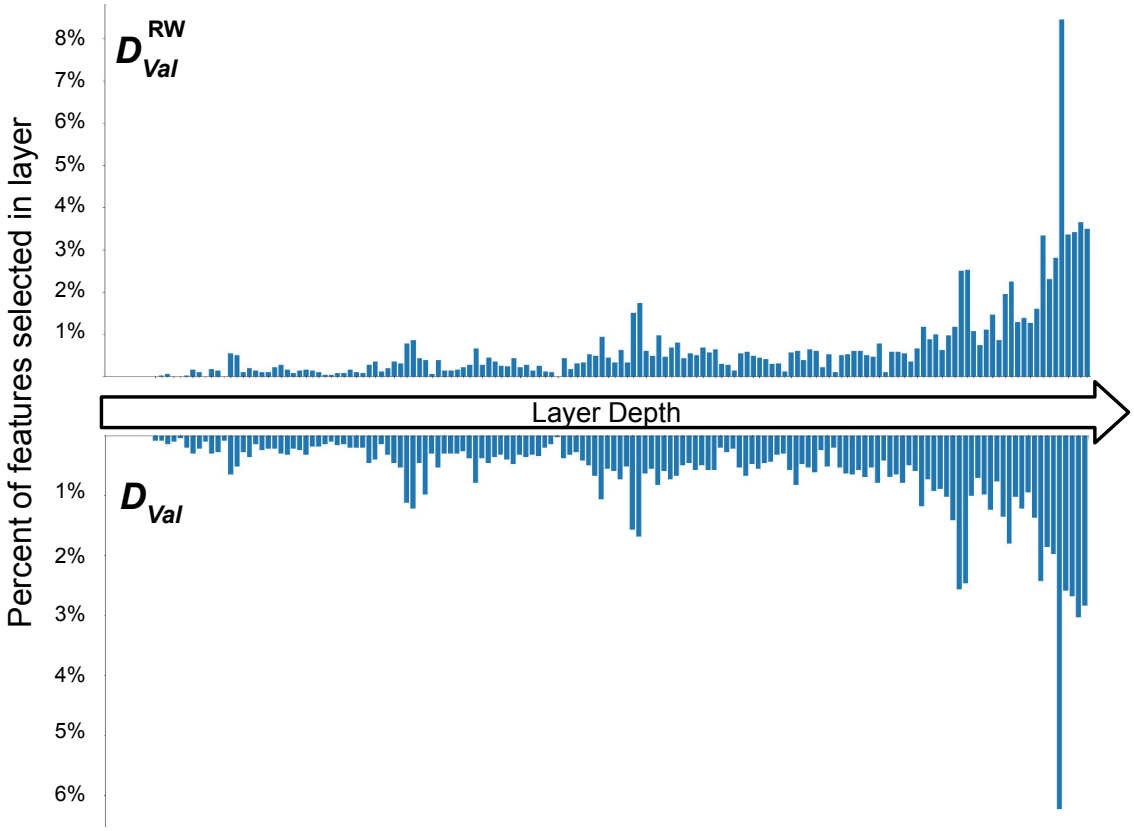

Figure 3.3: **HAM10000 and H2T-DFR**: Layerwise proportion of features extracted among the overall top 5% features selected using balanced **(top)** and unbalanced data **(bottom)** for feature selection. Layer depth displayed on the x-axis illustrates a minimal amount of features selected from early layers while most attributes are retained from deeper layers. **Note:** The y-axis value for each layer represents the portion of features selected from a given layer relative to the total number of features selected

depth for HAM1000 in figure 3.3 (but note similar trends are observed in the other datasets). This illustrates that despite not relying solely on the penultimate layer for feature selection, the most crucial information is found towards the end layers of the network. We also compare to a feature selection done on unbalanced data and observe that this tends to select more features from lower layers, which we attribute to the presence of strong spurious features in lower layers that may be selected for the unbalanced problem setting.

3.3.1 Ablation

Ablation testing was performed to determine whether the improvements in WGA are due to an increase in the number of features being used or if the actual features selected have a significant impact on performance. This was done by running experiments on the CelebA dataset by randomly selecting embeddings from the ResNet-50 backbone during the **Head2Toe** feature extraction phase and running the remainder of the algorithm as described in 1.

	Average WGA (%)
H2T-DFR(Random)	71.47 ± 2.30
H2T-DFR	81.12 ± 0.75

Table 3.2: Ablation testing results on CelebA dataset with τ = 56%. Average worst-group accuracy is over 5 seeds.

We observed a drop in WGA (3.2) of 9.65% relative to our proposed method when using a randomized feature selection step. For the ablation testing, τ was set to 56% as with a larger amount of features selected, the impact of the selection process is more pronounced. We note that along with *H2T-DFR(Random)* under performing, *H2T-DFR* also declined in WGA (compared to 3.1). This could indicate that by forcing the model to select a larger subset of features, the possibility of unstable features being chosen increases, leading to a decline in WGA. We conclude that the feature selection process is an essential step and motivates research into other feature selection methods as discussed in section 5.1. It also suggests that there are in fact unstable features since when random features are selected, the model performance is compromised. This may be due to the model relying on non-predictive elements as they are most likely to be among those selected when arbitrarily

chosen.

Chapter 4

Head2Toe with Linear Tuning

This chapter serves as a deeper look into the effectiveness of **Head2Toe** when combined with other transfer learning methods to improve overall average accuracy (as opposed to WGA in previous sections).

4.1 Background

We performed experiments on combining **Head2Toe** with the ideas presented in Ren, Guo, Bae, and Sutherland (2023). This paper takes a different approach at the transfer learning problem by shifting focus on the output head instead of the network's backbone. It acoomplishes this by first linear tuning and then finetuning to diminish large disruptions to pretrained weights. Using a random initialization of weights in the output head could cause significant disruptions in the pretrained weights during the initial training steps, where the model is forced to drastically adjust to reduce loss in subsequent steps.

For out-of-distribution tasks, full finetuning can destabilize pretrained parameters due to larger losses observed from the initial training steps as the model is forced to make drastic changes to correct for the large distribution shift (Kumar, Raghunathan, Jones, Ma, and Liang (2022)). This paper by Kumar et al. (2022) performs experiments comparing three different approaches on both in-distribution and out-of-distribution. Firstly, it shows that for in-distribution tasks, full finetuning outperforms the **NaiveTL** 2.1.4 method of only training the output head on a frozen backbone.

However, full finetuning suffers in performance for out-of-distribution tasks, where as **NaiveTL** is able to achieve better results as it utilizes a frozen backbone and therefore cannot affect pretrained weights by the large distribution shifts. Their *Linear Probing and Finetuning (**LP-FT**)* method outperforms on in-distribution and out-of-distribution tasks both of the other two methods by having a *Linear Tuning* step, which trains a *customized linear head* roughly adapted to the target task. They are then able to attach this *customized linear head* to a pretrained backbone and perform full finetuning.

With low domain affinity relative to the source task from the pretrained model, the initial shift in the feature extractor weights would be drastic compared to a high domain affinity set. For this reason, a potential solution is as follows:

(1) **Linear Tuning Step**: Initially, train an output head on the chosen pretrained network and stop training at a certain threshold before loss and accuracy fully converge to produce a *custom output head*

(2) **Model Reset**: With pretrained weights set back to their original values(e.g. through a newly initialized model), connect the backbone to the *custom output head*.

(3) **Finetune**: Train the new model until convergence to finetune weights and adapt to downstream task

With this approach, the *Model Reset* will consist of an output head *headed into the right direction*, which in turn will reduce arbitrary fluctuations in parameter weights.

4.2 Method

Linear Tuning followed by full finetuning, as shown in Kumar et al. (2022)has proven to enhance generalization capabilities of models by reducing large disruptions that would otherwise be present if the model was fully tuned on an out-of-distribution task. We aim to see if similar benefits can be found when the linear head is not only derived from the last layer, and includes intermediate layer information as in **Head2Toe**.

Dataset	Method	Mean Accuracy(%)
EuroSAT	H2T-FT	**95.33**
	H2T-LT (ours)	95.24
Flowers102	H2T-FT	**85.69**
	H2T-LT (ours)	85.56
SVHN	H2T-FT	84.19
	H2T-LT (ours)	**84.41**

Table 4.1: Mean test accuracy over 5 seeds.

4.2.1 Head2Toe with Linear Tuning (H2T-LT)

The proposed method, **H2T-LT**, begins with the **Heda2Toe** steps as described in 3.1.2 until feature selection is complete with a pre-trained ResNet-50 He et al. (2016). Following feature selection, a new **Head2Toe** model is initialized and given the set of selected features to produce its extended linear layer head which will be used for classification. At this point, we perform the *Linear Tuning step* (4.1) to produce our *custom output head*. A new **Head2Toe** model is initialized once again with the same selected features as well as the custom output head weights from the previous phase. It is then fully trained to convergence and produces our final results.

4.3 Results

H2T-LT experiments were performed for the following datasets from Zhai et al. (2020): EuroSAT, Flowers102 and SVHN.

As seen in 4.1, there is a negligible difference in performance between our **H2T-LT** method and regular **Head2Toe** with finetuning (**H2T-FT**).

Results suggest that there is no noticeable gain (or loss) in performance between the two methods but **H2T-LT** does have an extra step which we conclude to be unnecessary (given the results). The shortcomings of this method can be explained through the framework **Head2Toe** relies on. When building a linear output head from intermediate layers of a pre-trained network, the selected features possess parameter weights that are not arbitrary and are in fact heavily tuned on some source task. This eliminates the need to have an linear tuning phase and thus we see there is no

additional gain or loss in performance. With the output head weights being from the pre-trained backbone, the model is less prone to large fluctuations in parameter weights and any changes are simply to adapt to the target task, as opposed to correcting for some random initialization.

Chapter 5

Conclusions and Future Work

In this thesis we studied the problem of efficient transfer learning in the presence of spurious correlation. We propose H2T-DFR, a three-stage method that leverages Head2Toe (an efficient transfer learning method) Evci et al. (2022), and incorporate it in the pipeline of DFR Kirichenko et al. (2022), a state-of-the-art method to fight against spurious correlation. With **Head2Toe** providing a novel way of extracting intermediate layers and DFR reducing spurious feature bias, the combination of these methods proved to be an effective approach. H2T-DFR selects the most transferable features from all layers before applying DFR. Experiments on standard evaluation benchmarks demonstrate that H2T-DFR improves DFR, showing that efficient transfer learning methods can boost the worst-group predictive performance of robust-to-spurious correlation methods.

We further experiment and expand **Head2Toe** by incorporating *Linear Tuning* ideas presented in Ren et al. (2023). This proved to be ineffective and a redundant process as the features extracted from **Head2Toe** were not initialized randomly (set weights sourced from pretrained backbone), rendering the *Linear Tuning step* unnecessary.

5.1 Future Work

Finally, we discuss future direction and areas to expand upon based on the experiments and studied literature.

5.1.1 Extensions to Natural Language Settings

A large part of this thesis consisted of leveraging the novel idea presented in **Head2Toe** of making use of intermediate features for vision problems. This idea can be further explored for natural language models.

Implementing **Head2Toe** with the **BERT** (Bidirectional Encoder Representations from Transformer) model presented by Devlin, Chang, Lee, and Toutanova (2019) would be a sensible direction to explore the idea of leveraging intermediate features. **BERT**, as explained in the paper, is designed to be transferable on various tasks through a new output layer, which means the backbone architecture can remain stable (similar to the RESNET-50 which does not need to change for vision experiments conducted). With differences in architecture and large language models requiring significantly higher computation power, there remain certain points to consider to adapt **Head2Toe** for natural language settings.

(1) **Model size**. With a larger model, the number of parameters at play is quite large in comparison to the RESNET-50 employed for vision problems. The **BERT** base model is composed of 110 million parameters, whereas the large **BERT** model has 340 million parameters (Devlin et al. (2019)). This is in comparison to the approximately 25.6 million parameters in RESNET-50 (He et al. (2016)). The larger parameter count signifies a larger net of features to cover during feature selection, meaning the *Group-Lasso Regularization* (2.1.7 step which requires training with all parameters in a single matrix would be computationally expensive. This motivates the need for a different feature selection approach.

(2) **Feature Selection.** Model size will affect computation capabilities, and one of the ways to reduce this load is through a feature selection approach better suited for large language models. 5.1.2 discusses some advanced methods but an alternative approach would be to make use of a **PEFT** method such as LoRA (2.1.5) that can reduce the number of parameters to tune when considering the large matrix in the *Group-Lasso* step.

After adapting **Head2Toe** for language problems, there remains many viable directions to further improve performance such as combining **Head2Toe**, as per our proposed method, with DFR to reduce effects of spurious correlations.

5.1.2 Advanced Feature Selection

Feature selection in this thesis follows what is proposed in **Head2Toe** and leaves space for further consideration.

Zhou, Jin, and Hoi (2010) provides a stricter form of Group-Lasso regularization, named *Exclusive Lasso*. The added complexity comes from imposing competition within the groups themselves (as opposed to between groups) and adjusts weights within groups so only most important components retain importance. They prove this to be an effective approach for multi-task feature selection and thus could be of interest when applying the ideas in this thesis to such settings. *Two-layer feature reduction method (TLFre) for Sparse-Group Lasso* by Wang and Ye (2014) details a computationally efficient way of removing groups and features deemed insignificant to reduce load during optimization. It directly addresses the problem of computational issues with larger complex systems and could prove to be a powerful way of dealing with feature selection when considering **Head2Toe** for natural language tasks. Zhang et al. (2025) suggests **LLM-Lasso** as a way to leverage context from natural language and provide better direction for the Lasso regularization by tuning the penalties applied to feature parameter weights based on what the LLM deems an important feature.

A Appendix

	Waterbirds	CelebA	HAM10000
Optimizer	SGD	SGD	SGD
Learning Rate	0.003	0.0005	0.0003
Weight Decay	0.0004	0.0001	0.0001
Momentum	0.9	0.9	0.9
DFR Learning Rate	0.0001	0.0001	0.0005
DFR Weight Decay	0.0001	0.0001	0.0004
DFR Momentum	0.9	0.4	0.9
DFR Optimizer	SGD	SGD	SGD
Epochs	20	6	100
DFR Epochs	100	50	500
Batch Size	32	128	128

Table A: DFR - Hyperparameter setting for the 3 datasets presented.

	Waterbirds	CelebA	HAM10000
Optimizer	SGD	SGD	SGD
Learning Rate	0.003	0.0005	0.0003
Weight Decay	0.0004	0.0001	0.0001
Momentum	0.9	0.9	0.9
DFR Learning Rate	0.0001	0.0001	0.0005
DFR Weight Decay	0.0001	0.0001	0.0004
DFR Momentum	0.9	0.4	0.9
DFR Optimizer	SGD	SGD	SGD
Epochs	20	6	100
DFR Epochs	100	50	500
Batch Size	32	128	128

Table B: Affine-DFR - Hyperparameter setting for the 3 datasets presented

	Waterbirds	CelebA	HAM10000
Optimizer	SGD	SGD	SGD
Learning Rate	0.0005	0.0005	0.0003
Weight Decay	0.0004	0.0001	0.0001
Momentum	0.9	0.9	0.9
DFR Learning Rate	0.0005	0.0005	0.0005
DFR Weight Decay	0.0003	0.0003	0.0004
DFR Momentum	0.9	0.9	0.9
DFR Optimizer	SGD	SGD	SGD
Regularization Coefficient	0.00001	0.00001	0.0001
Epochs	70	20	100
DFR Epochs	500	300	500
Batch Size	32	128	128

Table C: H2T-DFR - Hyperparameter setting for the 3 datasets presented

References

Agarwal, V., Shetty, R., & Fritz, M. (2020). Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9690–9698).

Alabsi, B. A., Anbar, M., & Rihan, S. D. A. (2023). Cnn-cnn: Dual convolutional neural network approach for feature selection and attack detection on internet of things networks. *Sensors*, *23*(14). Retrieved from https://www.mdpi.com/1424-8220/23/14/6507 doi: 10.3390/s23146507

Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2020). *Invariant risk minimization.* Retrieved from https://arxiv.org/abs/1907.02893

Chen, L., Yan, X., Xiao, J., Zhang, H., Pu, S., & Zhuang, Y. (2020). Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10800–10809).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding.* Retrieved from https://arxiv.org/abs/1810.04805

Dreyer, M., Pahde, F., Anders, C. J., Samek, W., & Lapuschkin, S. (2023). From hope to safety: Unlearning biases of deep models by enforcing the right reasons in latent space. *arXiv preprint arXiv:2308.09437*.

Evci, U., Dumoulin, V., Larochelle, H., & Mozer, M. C. (2022). Head2toe: Utilizing intermediate representations for better transfer learning. In *International conference on machine learning* (pp. 6009–6033).

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, *2*(11), 665–673.

Hameed, H. W., Nanfack, G., & Belilovsky, E. (2024). *Not only the last-layer features for spurious correlations: All layer deep feature reweighting.* Retrieved from https://arxiv.org/abs/2409.14637

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition.* Retrieved from https://arxiv.org/abs/1512.03385

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

Hermann, K. L., Chen, T., & Kornblith, S. (2020). *The origins and prevalence of texture bias in convolutional neural networks.* Retrieved from https://arxiv.org/abs/1911.09071

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... Chen, W. (2021). *Lora: Low-rank adaptation of large language models.* Retrieved from https://arxiv.org/abs/2106.09685

Izmailov, P., Kirichenko, P., Gruver, N., & Wilson, A. G. (2022). On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*, *35*, 38516–38532.

Kirichenko, P., Izmailov, P., & Wilson, A. G. (2022). Last layer re-training is sufficient for robustness to spurious correlations. In *The eleventh international conference on learning representations.*

Kumar, A., Raghunathan, A., Jones, R., Ma, T., & Liang, P. (2022). *Fine-tuning can distort pretrained features and underperform out-of-distribution.* Retrieved from https://arxiv.org/abs/2202.10054

Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., & Liang, P. (2021). Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International conference on learning representations.*

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., ... Finn, C.

(2021). Just train twice: Improving group robustness without training group information. In *International conference on machine learning* (pp. 6781–6792).

Liu, T., Liu, X., Huang, S., Shi, L., Xu, Z., Xin, Y., ... Liu, X. (2024). *Sparse-tuning: Adapting vision transformers with efficient fine-tuning and inference.* Retrieved from https://arxiv.org/abs/2405.14700

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the ieee international conference on computer vision* (pp. 3730–3738).

Lv, K., Yang, Y., Liu, T., Gao, Q., Guo, Q., & Qiu, X. (2024). *Full parameter fine-tuning for large language models with limited resources.* Retrieved from https://arxiv.org/abs/2306.09782

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2022). *A survey on bias and fairness in machine learning.* Retrieved from https://arxiv.org/abs/1908.09635

Nagarajan, V., Andreassen, A., & Neyshabur, B. (2020). Understanding the failure modes of out-of-distribution generalization. In *International conference on learning representations.*

O'Shea, K., & Nash, R. (2015). *An introduction to convolutional neural networks.* Retrieved from https://arxiv.org/abs/1511.08458

Pezeshki, M., Kaba, O., Bengio, Y., Courville, A. C., Precup, D., & Lajoie, G. (2021). Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, *34*, 1256–1272.

Plumb, G., Ribeiro, M. T., & Talwalkar, A. (2022). *Finding and fixing spurious patterns with explanations.* Retrieved from https://arxiv.org/abs/2106.02112

Qiu, S., Potapczynski, A., Izmailov, P., & Wilson, A. G. (2023). Simple and fast group robustness by automatic feature reweighting. In *International conference on machine learning* (pp. 28448–28467).

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., ... Courville, A. (2019). On the spectral bias of neural networks. In *International conference on machine learning* (pp. 5301–5310).

Ren, Y., Guo, S., Bae, W., & Sutherland, D. J. (2023). *How to prepare your task head for finetuning.*

Retrieved from https://arxiv.org/abs/2302.05779

Sagawa, S., Koh, P. W., Hashimoto, T. B., & Liang, P. (2020a). Distributionally robust neural networks. In *International conference on learning representations.*

Sagawa, S., Koh, P. W., Hashimoto, T. B., & Liang, P. (2020b). *Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization.* Retrieved from https://arxiv.org/abs/1911.08731

Scardapane, S., Comminiello, D., Hussain, A., & Uncini, A. (2017, June). Group sparse regularization for deep neural networks. *Neurocomputing*, *241*, 81–89. Retrieved from http://dx.doi.org/10.1016/j.neucom.2017.02.029 doi: 10.1016/j.neucom.2017.02.029

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., & Netrapalli, P. (2020). *The pitfalls of simplicity bias in neural networks.* Retrieved from https://arxiv.org/abs/2006.07710

Shetty, R., Schiele, B., & Fritz, M. (2019). Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 8218–8226).

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In *Artificial neural networks and machine learning–icann 2018: 27th international conference on artificial neural networks, rhodes, greece, october 4-7, 2018, proceedings, part iii 27* (pp. 270–279).

Wang, J., & Ye, J. (2014). Two-layer feature reduction for sparse-group lasso via decomposition of convex sets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 27). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2014/file/dbea76ed2c0d0e42a285ac78fbe8e57e-Paper.pdf

Xin, Y., Du, J., Wang, Q., Lin, Z., & Yan, K. (2023). *Vmt-adapter: Parameter-efficient transfer learning for multi-task dense scene understanding.* Retrieved from https://arxiv.org/abs/2312.08733

Xin, Y., Luo, S., Zhou, H., Du, J., Liu, X., Fan, Y., ... Du, Y. (2024). *Parameter-efficient fine-tuning for pre-trained vision models: A survey.* Retrieved from https://arxiv.org/

abs/2402.02242

Xu, L., Xie, H., Qin, S.-Z. J., Tao, X., & Wang, F. L. (2023). *Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment.* Retrieved from https://arxiv.org/abs/2312.12148

Yazdanpanah, M., Rahman, A. A., Chaudhary, M., Desrosiers, C., Havaei, M., Belilovsky, E., & Kahou, S. E. (2022). Revisiting learnable affines for batch norm in few-shot transfer learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9109–9118).

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49-67. Retrieved from https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00532.x doi: https://doi.org/10.1111/j.1467-9868.2005.00532.x

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., & Gummadi, K. P. (2019). Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, *20*(75), 1–42. Retrieved from http://jmlr.org/papers/v20/18-262.html

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., ... Houlsby, N. (2020). *A large-scale study of representation learning with the visual task adaptation benchmark.* Retrieved from https://arxiv.org/abs/1910.04867

Zhang, E., Goto, R., Sagan, N., Mutter, J., Phillips, N., Alizadeh, A., ... Tibshirani, R. (2025). *Llm-lasso: A robust framework for domain-informed feature selection and regularization.* Retrieved from https://arxiv.org/abs/2502.10648

Zhou, Y., Jin, R., & Hoi, S. C. (2010, 13–15 May). Exclusive lasso for multi-task feature selection. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (Vol. 9, pp. 988–995). Chia Laguna Resort, Sardinia, Italy: PMLR. Retrieved from https://proceedings.mlr.press/v9/zhou10a.html

Zong, Y., Yang, Y., & Hospedales, T. (2022). Medfair: Benchmarking fairness for medical imaging. In *The eleventh international conference on learning representations.*

Zong, Y., Yang, Y., & Hospedales, T. (2023). Medfair: Benchmarking fairness for medical imaging.

In *The eleventh international conference on learning representations.*