TOWARDS ROBUST TRANSFER LEARNING: PREVENTING THE TRANSFER OF SPURIOUS FEATURES WITH TASK INTERPOLATION

KESHAVA KATTI [KATTIKES@SEAS], SIMING HE [SIMINGHE@SEAS], BRIAN LEE [WBLEE@WHARTON]

ABSTRACT. Pre-training and transferring deep neural networks is becoming increasingly prevalent due to the seeming ease with which such models can be fine-tuned to different target tasks. Yet fine-tuning remains relatively unprincipled: it heuristically relies on structures shared across image classes, as well as neural networks' ability to quickly adjust outputs after seeing a small number of images from a new task. Naturally, we ask whether fine-tuning allows "biases" from the source task to creep into the transferred model. A concrete example of bias is a spurious feature, which has high information value in the source task but is meaningless or outright misleading in the target task. We show that fine-tuning indeed allows spurious features to seep into the transferred model, but also that a simple task interpolation-based transfer algorithm can largely prevent the transfer of spurious features. We conjecture that information geometry, from which the task interpolation algorithm is drawn, presents a promising new direction in the study of bias transfer.

1. INTRODUCTION

Adapting neural networks that have been pre-trained on a source task to a target task via fine-tuning, i.e., running a few epochs of stochastic gradient descent (SGD) using a small target training set, is a convenient and increasingly popular transfer learning strategy. However, fine-tuning is frequently unable to eliminate the transfer of spurious features – features that have high information value in the source task, but are meaningless or outright misleading in the target task. For example, Salman, et al. (2022) [6] show that a model that has been pre-trained to classify dogs and cats using a training set in which dogs appear disproportionately often alongside humans remains sensitive to the presence of humans even after being fine-tuned on target images that do not include them.

Drawing from an information geometric model of transfer learning, we show that the transfer of spurious features can be largely prevented by a simple task interpolation-based transfer algorithm that requires no more target images nor epochs than canonical fine-tuning. Specifically, we adapt the uncoupled transfer algorithm from Gao & Chaudhari (2021) [2].

1.1. Contributions.

We recognize the theoretical connection between the transfer of spurious features and (the lack of) task interpolation. Our experiments show that (1) fine-tuning and uncoupled transfer achieve comparable loss and accuracy benchmarks on simple, well-behaved tasks, but (2) when the source and target tasks grow further apart, e.g., images are perturbed by spurious features, uncoupled transfer significantly outperforms fine-tuning. This suggests that uncoupled transfer encourages neural networks to "unlearn" spurious features.

2. BACKGROUND

Gao & Chaudhari (2021) develop an information geometric model of transfer learning to formally define the "distance" between tasks, and introduce the coupled transfer (CT) algorithm to compute this distance. Because the weights of the neural network being transferred serve as coordinates in the space of tasks, CT must run a subroutine that performs the transfer of interest before the task distance can be computed. Theoretical and experimental results suggest that this transfer subroutine, which involves the neural network gradually learning interpolations of the source and target tasks, is a more robust way to transfer models than ad hoc fine-tuning.

2.1. Information Geometry of Transfer Learning.

Gao & Chaudhari (2021) represent tasks as joint distributions $p(x, y)$ over data x and labels y; together, tasks live in an information space. Specifically, consider source task $p^s(x, y)$ and target task $p^t(x, y)$. Living between these tasks are interpolated tasks $p^\tau(x, y)$, created by "reshaping" $p^s(x, y)$ to resemble $p^t(x, y)$ with τ controlling how closely we replicate the latter.

Also assume we have access to a neural network parameterized by weights $w \in \mathbb{R}^p$ and trained with cross-entropy loss. Let w^s, w^t, w^τ be the weights that this neural network would have if it was trained directly on the source, target, and interpolated tasks, respectively. Then we can write $p^s(x, y) = p^s_{w^s}(y|x)p^s(x)$, $p^t(x, y) = p^t_{w^t}(y|x)p^t(x)$, and $p^\tau(x, y) = p^\tau_{w^\tau}(y|x)p^\tau(x)$, where the first factors are conditional label distributions that our neural network learns.

More generally, define $\mathcal{M} = \{p_w(y|x) : w \in \mathbb{R}^p\}$ as the manifold of conditional label distributions parameterized by w. Clearly, $p^s_{w^s}(y|x)$ and $p^t_{w^t}(y|x)$ are two points on $\mathcal{M}$, and $p^\tau_{w^\tau}(y|x)$ lies on some path connecting them. Then transfer learning can be modeled as a neural network with weights w_s, i.e., pre-trained on the source task, updating its weights to transport $p^s_{w^s}(y|x)$ to $p^t_{w^t}(y|x)$ along $\mathcal{M}$ by way of various choices of $p^\tau_{w^\tau}(y|x)$.

2.2. **Coupled Transfer.** Gao & Chaudhari (2021) define the distance between tasks as the minimum distance that weights must travel along $\mathcal{M}$ during the transfer process, and introduce the coupled transfer (CT) algorithm to compute this distance. Defining task distance in terms of weights means CT must perform the transfer of interest and determine the weight trajectory before the task distance can be computed. Thus, CT is divided into two subroutines: the transfer subroutine and the distance computation subroutine. We focus on the former.

The neural network needs access to interpolated training sets to update weights, so we must pin down how to construct interpolated tasks. Given source dataset $D_s = \{(x^i_s, y^i_s)\}_{i=1}^{N_s}$ and target dataset $D_t = \{(x^i_t, y^i_t)\}_{i=1}^{N_t}$, construct empirical marginal input distributions $\hat{p}_s(x) = \frac{1}{N_s}\sum_{i=1}^{N_s} \delta_{x^i_s}(x)$ and $\hat{p}_t(x) = \frac{1}{N_t}\sum_{i=1}^{N_t} \delta_{x^i_t}(x)$, where the Dirac delta is an ordinary indicator function. Gao & Chaudhari (2021) draw from optimal transport literature and suggest that marginal interpolated input distributions take the form $\hat{p}^\tau(x) = \sum_{i,j} \Gamma^*_{ij}\delta_{\tau x^i_s + (1-\tau)x^j_t}(x)$, where Γ^*_{ij} is an entry of the "optimal transport coupling" matrix included to ensure that $\hat{p}_s(x)$ takes the optimal route to $\hat{p}_t(x)$. Put simply, the marginal interpolated input distribution is imagined to be supported on convex combinations of x^i_s and x^i_t (see Gao & Chaudhari (2021) Appendix D for theoretical justifications). Similarly, we can cook up pseudo-labels for these interpolated data as $y^i_\tau = \tau y^i_s + (1-\tau)y^i_t$. This allows us to generate an interpolated training set with labels and train a neural network on an interpolated task.

2.3. **Robust Transfer by Task Interpolation.** Now suppose we have a grid of $\tau \in [0, 1]$. Training the source model on the sequence of interpolated tasks parameterized by this grid yields a complete transfer process. Interestingly, Gao & Chaudhari (2021) show that this CT-based transfer process minimizes the generalization gap, which suggests the transfer subroutine of CT – and task interpolation, more broadly – can serve as a more "robust" way to transfer a model than ad hoc fine-tuning.

This project takes a step towards demonstrating this intuition empirically by testing whether robust transfer via interpolated tasks can eliminate spurious features better than ad hoc fine-tuning. Given constraints on time and computational resources, we will use the transfer subroutine from the uncoupled transfer (UT) algorithm, which works identically to CT except that optimal transport coupling is not calculated. That is, $p^\tau(x, y) = \tau p^s(x, y) + (1-\tau)p^t(x, y)$. Because CT can only do better than UT by calculating the optimal transport couplings, UT's transfer performance can be interpreted as a lower bound on CT's transfer performance.

3. Related Work

3.1. **Iso-classification Process.** Inspired by iso-thermal processes in thermodynamics, in which a slow-evolving system remains in thermal equilibrium with its surroundings, Gao & Chaudhari (2020) [3] develop an "iso-classification process" that ensures that a neural network's classification loss remains constant as the underlying task slowly changes, i.e., from source to target. While this approach involves task interpolation, there are more constraints on how quickly the interpolation parameter is increased. But, in turn, a stronger guarantee about the transferred model's loss is attained.

3.2. **Mixup Regularization.** Zhang, et. al. (2018) [7] find that training neural networks on convex combinations of data and labels from the training set significantly increases their generalization performance, as well as robustness against adversarial attacks. This data augmentation strategy, called mixup, strongly resembles our simple task interpolation scheme, but was not proposed with transfer learning in mind. More recent work by Carratino, et. al. (2022) [1] report that mixup is theoretically equivalent to a combination of regularization strategies, such as label smoothing and dropout.

4. Approach

We implement two experiments to test whether UT can better prevent the transfer of spurious features than ad hoc fine-tuning: one with synthetic data and another with subsets of CIFAR-100.

4.1. **Synthetic Data Experiments.** We designed a synthetic dataset to explore the transfer of spurious features during transfer learning. Each datum has 50 features from a multivariate normal distribution. The means are generated from a uniform distribution between -10 and 10. The covariance matrix is a diagonal matrix with variance generated uniformly between 0 and 10. The function $\sum_{i=0}^{30} \sin x_i + x_i$ is used to create the source task output and $\sum_{i=20}^{50} \cos x_i - x_i$ is used to create the target task output. We used these functions to create $10,000$ source data and $10,000$ target data. The first

20 features are only used to create the source data, so they have no impact on the target data output. We can consider those twenty features as the spurious features of our target tasks. In contrast, the middle 10 features contribute to both the output of the source task and the target task. We do a train-validation data split so that we have $8,000$ training samples for the source task and $2,000$ training samples for the target task.

We use two model-agnostic methods to evaluate the transfer of spurious features. The first method is feeding data with spurious features to the transferred model and checking if the loss changes a lot. The second is the Local Interpretable Model-agnostic Explanations (LIME) method proposed by Ribeiro, et. al. (2016) [5]. LIME is a model-agnostic method since it perturbs the input and analyzes the output – it makes no assumptions about the inner workings of the model. It is also local, meaning we look at data points one by one and explain the model performance around that data point. This method outputs the impact of each feature on the output under perturbation.

4.2. **CIFAR-100 Experiments.**

We began establishing some preliminary results for transfer learning using fine-tuning and UT. In order to benchmark these initial results with those of Gao & Chaudhari (2021), we selected the same 5 datasets from CIFAR-100 as them. Specifically, these datasets were (1) herbivores (camel, cattle, chimpanzee, elephant, kangaroo), (2) carnivores (bear, leopard, lion, tiger, wolf), (3) vehicles-1 (bicycle, bus, motorcycle, pickup truck, train), (4) vehicles-2 (lawn mower, rocket, streetcar, tank, tractor), and (5) flowers (orchid, poppy, rose, sunflower, tulip). See Figures 3a, 3b for examples from dataset (1) and (2), respectively. Each of these 5 datasets consisted of 2,500 images (i.e., 500 from each class), where each image was of size $(3, 32, 32)$.

Fine-tuning distance is defined as $\int_0^1 |\mathrm{d}w|$, the length of the weight trajectory under fine-tuning. Gao & Chaudhari (2021) provide a 5×5 matrix diagram showing the pairwise fine-tuning distance and UT distance between each of the aforementioned 5 datasets. Our preliminary goal was two-fold. First, we wanted to achieve validation accuracy values comparable to those in Gao & Chaudhari (2021). Second, we hoped that the fine-tuning validation accuracy values would be inversely proportional to the fine-tuning distance between the source and target tasks. Similarly, we hoped that the UT validation accuracy values would be inversely proportional to the UT distance between the source and target tasks. Meeting this two-fold goal would give us evidence that we have correctly implemented both fine-tuning and UT models and could move on to different data.

To that extent, we set up an experiment to evaluate how spurious features in a source dataset could affect the model's ability to transfer. Using a similar strategy as Salman et al. (2022) [6], we embedded a 4×4 image of a tennis ball at a random location in a 32×32 CIFAR-100 image (Figures 3c, 3d). Specifically, we created a perturbed source dataset in which a single class was "treated" with tennis ball embeddings. For example, in the herbivore dataset, only images from the kangaroo class had a tennis ball embedded in them. We then trained a backbone on the perturbed dataset with the expectation that the network would associate the perturbed class label with tennis balls. In our example, the network would associate kangaroos with tennis balls. We then created a target dataset where all images had tennis balls. We hypothesized that the model would mistakenly believe that all images in this target dataset belonged to a single class (analogously to how it learned that all source images with tennis balls should be labeled kangaroos). We could then use how confused the model became (i.e., how high the loss was on the target set) to quantify how much the model transferred spurious features.

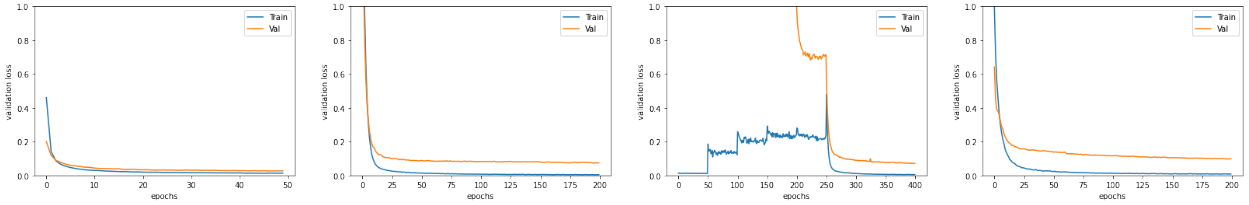
5. Experimental Results

5.1. **Synthetic Data Experiments.**

The model is a 6-layer multi-layer perceptron (MLP) with $92,001$ parameters. We first trained the backbone and got a validation loss of 0.027. We then trained three models for the target task: (1) full-network fine-tuning, (2) UT, and (3) training from scratch. To align our experiment with what happens in real data, we randomly permuted the first 20 features of the target dataset. This manipulation is comparable to removing the "tennis ball" from the target data images, as in Salman et al. (2022) [6]. We used the permuted data for all training on the target task. Figure 1 shows the training and validation results. The full-network fine-tuned model, UT model, and from-scratch model have losses 0.075, 0.072, and 0.099, respectively, on the permuted target validation dataset.

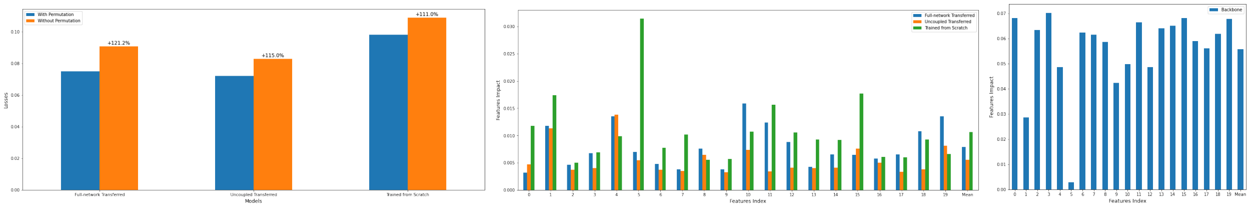
To check if spurious features impacted the performance of the transferred model, we added the spurious features back. Specifically, we used the original first 20 features in another target validation dataset and compared the change of validation loss as shown in Figure 2a. The losses increased for all three models, though the amount of increase is comparable. It follows that the spurious features indeed impacted the output, and the impact exists not only for the transferred models but also for the model trained from scratch. Hence, there is no evidence suggesting that the impact of spurious features is significantly larger for a transferred model.

Moreover, we looked into the details of spurious features' impact through LIME. Figure 2b shows the impact of the first 20 features on the output after perturbation. Again, the perturbation of spurious features did not change the output



(A) Backbone (B) Full-network Fine-tuning (C) Uncoupled Transfer (D) Training from Scratch

FIGURE 1. Losses of models trained on synthetic data.



(A) Loss increase using non-permuted features (B) LIME on target task models (C) LIME on backbone

FIGURE 2. Evaluation result on synthetic data.

Source Task	Target Task	Fine-Tuning Distance	Full-Network Loss/Accuracy	Fixed-Feature Loss/Accuracy
vehicles-1	vehicles-2	26	0.016 / 90.6%	1.220 / 50.8%
herbivores	flowers	68	0.111 / 73.6%	1.419 / 39.2%
carnivores	flowers	110	0.088 / 67.2%	1.401 / 41.4%

TABLE 1. Fine-tuning results for very similar (top row), moderately similar (middle row), and very different (bottom row) source and target distributions.

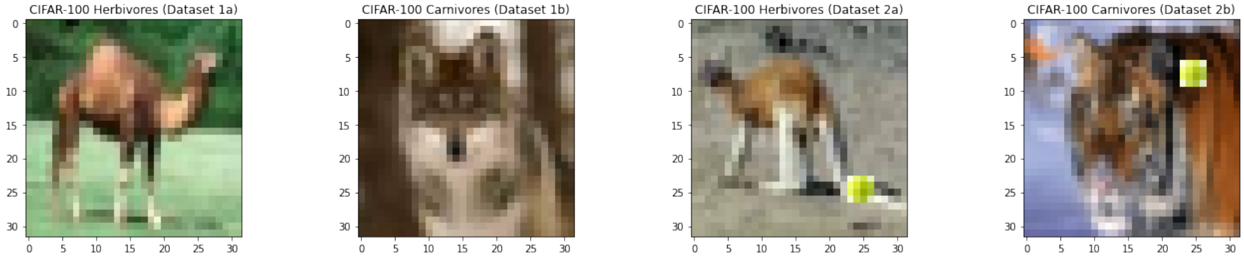
noticeably, and the impact of spurious features on the transferred model was no larger than the model trained from scratch. Compared to Figure 2c, the impact of spurious features all reduced significantly after training on the target dataset. We note that the uncoupled transferred model was more robust to the perturbation of spurious features (i.e., the orange bars in Figure 2b are mostly lower), which suggests that uncoupled transfer learning is more robust to spurious features than ad hoc fine-tuning. However, we need more experiments before drawing a conclusion.

5.2. **CIFAR-100 Experiments.** We used a 10-layer convolutional neural network (CNN) with ReLU non-linearities, dropout, batch normalization, and a final fully-connected layer. We began by pre-training the model on the source tasks using SGD for 50 epochs. Particularly, we used a mini-batch size of 16, learning rate of 10^{-2}, and cross-entropy loss. In total, 5 backbones were trained, one for each source task. For each backbone, 50 epochs of training was sufficient to achieve about 95% training accuracy and 80% validation accuracy.

We executed three different fine-tuning experiments, the first with very similar source and target distributions, the second with moderately similar source and target distributions, and the last with very different source and target distributions. As mentioned before, these notions of similarity and difference are captured by the fine-tuning distance. Table 1 includes the validation loss and accuracy for each of these three experiments under two conditions: (1) re-training the entire network (full-network) and (2) re-training only the last fully-connected layer (fixed-feature). As expected, Table 1 shows a relative decrease in accuracy as fine-tuning distance increases, and the losses published by Gao & Chaudhari (2021) lie between the losses we achieved for the two experimental settings. These two observations suggest that we have properly implemented fine-tuning. Next, we executed an analogous set of experiments for UT, whose results are summarized in Table 2. In this setting, we opted only for full-network tuning. We executed UT for $\tau \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, with 10 epochs per value of τ, totaling to 50 epochs of training. That meant that the total amount of training for fine-tuning and UT were the same. It is important to note that the results of UT were very similar to full-network fine-tuning for simple, unperturbed CIFAR-100 tasks.

Source Task	Target Task	Uncoupled Distance	Full-Network Loss/Accuracy
vehicles-1	vehicles-2	0.3	0.360 / 93.6%
herbivores	flowers	0.28	1.336 / 65.4%
carnivores	flowers	0.32	1.53/ 70.2%

TABLE 2. UT results for same transfer learning tasks. Note that uncoupled distance differs slightly in relative order from fine-tuning distance on the same tasks.


CIFAR-100 Herbivores (Dataset 1a)
CIFAR-100 Carnivores (Dataset 1b)
CIFAR-100 Herbivores (Dataset 2a)
CIFAR-100 Carnivores (Dataset 2b)
0
5
10
15
20
25
30

(A) Camel (Unperturbed) (B) Wolf (Unperturbed) (C) Kangaroo (Perturbed) (D) Tiger (Perturbed)

FIGURE 3. Examples of CIFAR-100 images from herbivores and carnivores datasets with and without spurious tennis ball.

Source Task	Target Task	Average Fine-Tuning Accuracy	Average Uncoupled Distance Accuracy
vehicles-1	vehicles-2	87.5%	90.2%
herbivores	flowers	57.3%	70.2%
carnivores	flowers	20.0%	49.1%

TABLE 3. Transfer learning with UT on data with spurious features (i.e., tennis ball) performs significantly better on average than fine-tuning. Average taken over three trials.

We went on to perturb each of the 5 datasets as described in 4.2. All backbones trained on the perturbed source datasets achieved between $95 - 99\%$ training accuracy but converged to about 20% validation accuracy on the target dataset, which is exactly what we hypothesized in 4.2. In the herbivore example, the model began outputting label “4,” which corresponds to the kangaroo, for all images, leading to the observed 1 in 5 accuracy. We then executed both full-network fine-tuning and full-network UT with these backbones. Next, the backbones, which were trained such that only a single class had the tennis ball, were transferred to a target task where all classes had the tennis ball. The results contained in Table 3 show overwhelming support for UT being the more robust method for preventing the transfer of spurious features. We found that the fine-tuned model would constantly get stuck at 20% accuracy when transferring to the perturbed dataset, meaning, over several attempts, its average accuracy was much lower than the UT model, which rarely got stuck. We hypothesize that task interpolation in UT allows the model to gradually realize that tennis balls should be ignored, while fine-tuning “drops” the model into a setting where it is overwhelmed and confused.

6. DISCUSSION

The synthetic data experiments allow us to conclude that spurious features are not necessarily transferred from the source to target task. In fact, the impact of spurious features is no larger in transferred models compared to models trained from scratch on the target task. Based on the CIFAR experiments, we conclude that fine-tuning and UT perform comparably on simple, well-behaved tasks. Our 5-class subsets of CIFAR-100 were relatively straightforward to learn, and neither approach seemed to outperform the other. But we noticed that as the tasks became more difficult, either due to larger fine-tuning / uncoupled distance or due to back-door adversarial attacks (i.e., tennis ball), UT consistently outperformed fine-tuning. The CIFAR experiments show that UT is the more robust method of transfer learning that prevents the transfer of spurious features. In future work, we would like to use more evaluation metrics to pinpoint the impact of spurious features in the CIFAR experiments, as well as check if the results we got for image classification tasks generalize to other machine learning tasks.

REFERENCES

[1] Carratino, L., et. al. (2022). On Mixup Regularization. arXiv.
[2] Gao, Y. & Chaudhari, P. (2021). An Information-Geometric Distance on the Space of Tasks. arXiv.
[3] Gao, Y. & Chaudhari, P. (2020). A Free-Energy Principle for Representation Learning. arXiv.
[4] Krizhevsky, A. (2009) Learning Multiple Layers of Features from Tiny Images. Technical Report TR-2009, University of Toronto, Toronto.
[5] Ribeiro, M., et. al. (2016). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. ACM.
[6] Salman, H., et. al. (2022). When does Bias Transfer in Transfer Learning? arXiv.
[7] Zhang, H., et. al. (2018). mixup: Beyond Empirical Risk Minimization. ICLR.