TOPOLOGICAL VANILLA TRANSFER LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper we investigate the connection of topological similarity between source and target tasks with the efficiency of vanilla transfer learning (i.e., transfer learning without retraining) between them. We discuss that while it is necessary to have strong topological similarity between the source and target tasks, the other direction does not hold (i.e., it is not a sufficient condition). To this extent, we further investigate what can be done in order guarantee efficient feature representation transfer that is needed for such vanilla transfer learning. To answer this, we provide a matrix transformation based homeomorphism (i.e., topology preserving mapping) that significantly improves the transferability measures while keeping the topological properties of the source and target models intact. We prove that while finding such optimal matrix transformation is typically APX-hard, there exists an efficient randomised algorithm that achieves probably correct approximation guarantees. To demonstrate the effectiveness of our approach, we run a number of experiments on transferring features between ImageNet and a number of other datasets (CIFAR-10, CIFAR-100, MNIST, and ISIC 2019) with a variety of pretrained models (ResNet50, EfficientNetB3, and InceptionV3). These numerical results show that our matrix transformation can increase the performance (measured by F-score) by up to 3-fold.

1 INTRODUCTION

Transfer learning is a subarea of machine learning where the main goal is to identify the most efficient ways to reuse pretrained models for new tasks for new tasks, typically in new domains (Pratt, 1996; Daume III & Marcu, 2006; Goodfellow et al., 2016). With the significant increase in both the size of datasets and models, transfer learning methods are becoming more essential, due to the fact that they can significantly reduce the cost of model training for the new task, e.g., by using the learnt parameters of the pretrained model as an initial starting point for the training process on the new task (Zamir et al., 2018; Achille et al., 2019; Bao et al., 2019; Nguyen et al., 2020). Therefore, it is no surprise that much work has been conducted on transferability estimation, that is, to quantitatively estimate how efficient it is to transfer *a priori* learned knowledge from one task to another (Ben-David et al., 2007; Mansour et al., 2009; Tran et al., 2019; Nguyen et al., 2020). Typically, theoretical results regarding transferability estimation involve the divergence between the input distributions associated with each domain. More specifically, the more alike both distributions are in terms of statistical distance, the better the generalisation bound. As a result, much research has focused on learning feature transformations of the target domain so that both input distributions are closer in terms of statistical divergence (Ben-David et al., 2007; Mansour et al., 2009).

However, work within this line of research, as well as within the broader transfer learning field in general, typically assume that there is also a retraining phase during the transfer process. The main reason for this is that even with a good source task with high transferability measures, reusing it without retraining often results in poor performance, as these measures only provide theoretical generalisation bounds. Therefore, these transferability measures should only be used as an indicator whether the learnt feature representation of a source task can be a good initialisation for the (re)training process for the target task (Tran et al., 2019; Nguyen et al., 2020).

In this paper we consider a slightly different problem of vanilla transfer learning, where retraining is not possible in the transfer process, due to either lack of resources such as compute or time. This setting occurs in many real-world applications, ranging from edge computing where classification tasks are run on local and computationally limited devices, to rescue drones in disaster response scenarios where there is no time to retrain the model to adopt to the new situations, or to exploration robotics

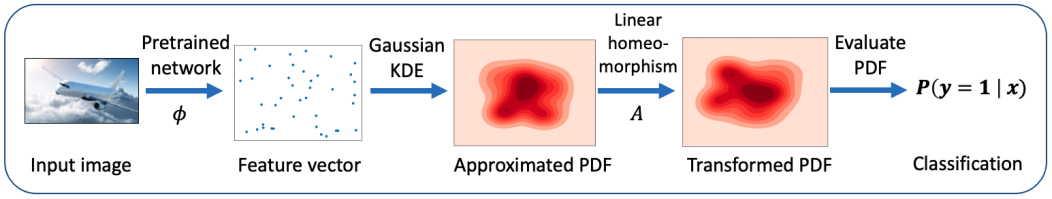


Figure 1: The integration of RMMS, our proposed linear homeomorphism, into the vanilla transfer learning pipeline.

where both compute and time are sparse resources. As existing techniques do not provide good solutions for transfer learning without retraining, or do optimise theoretical measures that are not computable in practice (Nguyen et al., 2020), we ask the question whether there is efficient vanilla transfer learning method that is both supported by theoretical justification and practical efficiency.

Recently, a number of empirical works suggest topological similarity (we formalise this notion in Section 2) as a means for measuring transferability between source and target tasks (Ramamurthy et al., 2019). More specifically, models with decisions boundaries that are topologically similar to that of the target task often outperform models with decision boundaries which are topological dissimilar. Unfortunately, to date there have not been any theoretical results which explains this empirical phenomena.

1.1 Our Contributions

1. Topological distance vs. statistical distance. Motivated by this line of observations, we aim to provide a theoretical investigation on whether topological similarity will lead to efficient feature representation transfer. To do so, we first investigate how the topological and statistical distances between input distributions are related. In short, the bottleneck stability theorem tells us that the topological distance between any two functions is bounded above by their functional distance (Cohen-Steiner et al., 2006). As a result, our first contribution is to show that small topological distance between distributions forms a necessary condition for their statistical similarity. This explains why the topological similarity of decision boundaries can be so effectively used for vanilla transfer learning.

2. Computationally efficient linear homeomorphism. Note that the necessary condition above does not guarantee that efficient vanilla transfer learning is achievable when there is a topological similarity. Therefore, to further improve the performance of vanilla transfer learning, we ask whether it is possible to reduce the statistical distance while keeping the topological similarity intact (as the latter might contain useful information that can be utilised in explainability or further data processing (Naisat, 2020)). To address this question, we design of an algorithm for learning a feature transformation of input examples in the target domain, with the goal of reducing the statistical divergence between source and target input distributions. More precisely, our algorithm aims to learn a linear homeomorphism which reduces the statistical distance of the target input distribution to the input distribution of the source domain. We choose to learn a homeomorphism so that topological similarities between both distributions are preserved in their entirety, and restrict ourselves to linear transformations with computational tractability in mind. To achieve this, we first approximate the problem of finding optimal linear homeomorphism with a minmax discrete optimisation model. While we conjecture that the solving latter is likely to be difficult (we prove that a restricted version of this problem is APX-hard), we provide an algorithm that finds a near optimal solution with high probability.

3. Application to vanilla transfer learning. In our third contribution, we apply our linear homeomorphism to a number of vanilla transfer learning cases. In particular, we test how our proposed homeomorphism performs on transfer learning tasks from ImageNet to CIFAR-10, from ImageNet to Fashion MNIST, and from ImageNet to the ISIC'19 skin cancer dataset, each with EfficientNetB3, ResNet50, and Inception V3 architectures. The pipeline of integrating our method into a pretrained model is shown in Figure 1. With extensive numerical evaluations, we show empirically that applying our feature transformation leads to a massive improvement in target domain performance. From ImageNet to CIFAR-10/100 we have an increase in performance of 269.4%, to MNIST an increase in performance of 114.1%, and to ISIC'19 an increase in performance of 98.3%. For further details on our pipeline and numerical results see Section 5.

2 RELATED WORK

Topological Data Analysis. Persistence diagrams provide a planar representation of the topology of an underlying dataset with strong theoretical guarantees. As such, there has been a research effort to integrate this additional topological information into machine learning. Persistence diagrams can be embedded into feature vectors or functional summaries for input into arbitrary machine learning models (Adams et al., 2017; Bubenik, 2015; Rieck et al., 2019). There are also positive-definite kernels defined on the space of persistence diagrams, allowing the use of persistence diagrams in kernel methods (Reininghaus et al., 2015; Carrière et al., 2017). Machine learning has demonstrated that learning application specific embeddings is generally far better than fixed vectorisation methods, and indeed such techniques exist to learn embeddings of persistence diagrams as part of a layer in deep learning (Hofer et al., 2017; 2019b; Carrière et al., 2020). Persistence diagrams and their embeddings, collectively referred to as persistence-based summaries, can also be used as part of a topological term in a loss function, either as a topological loss or for topological regularisation (Chen et al., 2018; Gabrielsson et al., 2020; Clough et al., 2020). These topological loss terms have also been used to topologically restrict the latent space in autoencoders (Hofer et al., 2019a; Moor et al., 2020).

Topology has also been used to link the performance of pretrained models on new datasets. Guss & Salakhutdinov (2018) empirically demonstrate that the topological complexity of decision boundaries is linked to the generalisation capability of the neural network. Ramamurthy et al. (2019) show that if you choose pretrained classifiers with similar topological complexity to that of a dataset then the pretrained classifier will perform better on the dataset than if you choose a classifier with dissimilar topological complexity. Davies et al. (2020) cluster the persistence diagrams of the decision boundaries of pretrained models and datasets and demonstrate that models perform better on datasets that are associated to the same cluster centre. Note that these works choose a pretrained model that has a similar topology. We only have access to a single pretrained model, and we explicitly do not change the topology of its decision boundary. In fact, we learn a homeomorphism (a topology-preserving map) on its decision boundary. Meanwhile,

Transfer Learning. The approach we describe in this work falls into the class of feature representation transfer methods. Feature representation transfer methods attempt to learn feature representations of both the source and target domain which reduce the distributional divergence between domains. For example Long et al. (2014) proposed an algorithm for feature learning which aims to jointly adapt the marginal and conditional distributions of both domains via a dimensionality reduction procedure. Ding et al. (2018) learn robust features across domains in a reproducing kernel Hilbert space via maximum mean discrepancy. Numerous works (Courty et al., 2014; Yan et al., 2018) leverage optimal transport techniques to ensure the input distribution of the target domain is close to the input distribution of the source domain in terms of Wasserstein distance. In contrast to these approaches, our methodology is predicated on the assumption that the topological distance between both input distributions is low. Our method focuses on transforming the input space of the target domain via a simple and tractable linear hoemomorphism as a result.

Note that all approaches for developing good feature representations are predicated on theoretical bounds for domain adaption. That is, generalisation bounds specifying how similar the risk of a hypothesis in the source domain is to its risk in the target domain. Nearly all such bounds rely on some measure of statistical divergence between distributions. Ben-David et al. (2007) use the $\mathcal{A}$-distance, a restricted version of total variation distance, and show that efficient domain adaptation is impossible when the $\mathcal{A}$-distance is large. Building upon these results, Mansour et al. (2009) derive generalisation bounds for more general settings using a related distance known as the discrepancy distance. Shen et al. (2018) devise methods for minimising the empirical Wasserstein distance between input distributions, whilst Wu & Zhuang (2020) develop for minimising the distance between the characteristic functions of both distributions. In our work, we focus on minising the total variation distance between distributions, due to its straightforward and interpretation and its connection to common statistical divergence measures for domain adaptation used in the literature.

3 BACKGROUND

3.1 TOPOLOGICAL DATA ANALYSIS

In this section, we give a brief description of the elements of topological data analysis leveraged throughout this paper. Our intent is not to be rigorous, but provide a conceptual overview of relevant elements of the theory. We direct readers to Cohen-Steiner et al. (2006) for a more rigorous exposition (we also provide some further definitions in Appendix D).

More precisely, we describe persistence diagrams, a key tool in topological data analysis for capturing topological information regarding a point cloud or function. In particular, we will consider persistence diagrams constructed from the sublevel sets of tame functions. In short, a persistence diagram of a function $\mathcal{D}(f) \subseteq \bar{\mathbb{R}}$, consists of pairs of real numbers indicating the birth and death points of topological features possessed by sublevel sets of the function, in union with the diagonal. As a result, persistence diagrams, provide a succinct summary of the topological features of a function.

Two persistence diagrams can be compared through the bottleneck distance.

Definition 3.1. *The bottleneck distance between two multisets X and Y is $d_B(X,Y) = \inf_\gamma \sup_x \|x - \gamma(x)\|_\infty$ where $x \in X$ and γ ranges over all bijections from X to Y.*

Of particular interest is the bottleneck stability theorem, which implies that the bottleneck distance between the persistence diagrams of two functions is upper bounded by their functional distance.

Proposition 1 (Main Theorem from Cohen-Steiner et al. (2006))**.** *Let $\mathbb{X}$ be a triangulable space with continuous tame functions $f, g : \mathbb{X} \to \mathbb{R}$. Then the persistence diagrams satisfy $d_B(D(f), D(g)) \leq \|f - g\|_\infty$.*

Note that the bottleneck stability theorem provides an explicit link between the geometries of functions and their topologies. More specifically, when the functions f and g play the role of probability densities, the bottleneck stability theorem links statistical divergence to topology.

3.2 TRANSFER LEARNING

In this subsection, we will formalise the notion of transfer learning used throughout this paper, largely following the notation of Ben-David et al. (2010). We define a domain as a pair, consisting of a distribution $\mathcal{D}$ on an input space $\mathcal{X}$ and a labelling function $f : \mathcal{X} \to [0, 1]$ indicating the expected labelling for each member of the input space. For a given domain adaption problem, we specify two domains, a source domain and a target domain. We denote by $\langle \mathcal{D}_S, f_S \rangle$ and $\langle \mathcal{D}_T, f_T \rangle$ the source and target domains respectively.

A hypothesis is a function, $h : \mathcal{X} \to \{0, 1\}$, from the input domain to the unit interval. We define the source error, $\epsilon_S(h)$, of a given hypothesis as follows:

$$\epsilon_S(h) = \mathbb{E}_{x \sim \mathcal{D}_S} \big[\, |h(x) - f_S(x)| \,\big]. \tag{1}$$

That is, the source error of a given hypothesis is the probability it disagrees with the given labelling function. We define the target risk ϵ_T, in a similar manner. We adopt the general definition of transfer learning in Pan & Yang (2009):

Definition 3.2. *(Transfer Learning) Given source domain $\langle \mathcal{D}_S, f_S \rangle$ and a target domain $\langle \mathcal{D}_T, f_T \rangle$, transfer learning aims to improve the learning of the predictive function f_T using knowledge of the source domain.*

Additionally, we will assume that we have access to a labelled training sample from both domains. In what follows, our goal will be to learn a good classifier for the target domain via *feature representation transfer* from source domain. That is, we will attempt to use the source domain to learn a good feature representation for learning the labelling function for the target domain, f_T.

4 A TOPOLOGICALLY MOTIVATED SCHEME FOR TRANSFER LEARNING

Intuitively, the usefulness of source domain in determining a good labelling function for a target domain depends on two factors: (i) how similar the input distributions for both domains are and (ii)

the similarity between the labelling functions in both domains. This intuition is typically reflected in statistical results bounding the performance of a hypothesis in the target domain relative to its performance in the source domain. For example, consider the following prototypical statistical transfer learning bound proposed by Ben-David et al. (2007):

$$\epsilon_T(h) \leq \epsilon_S(h) + d_{TV}(\mathcal{D}_S, \mathcal{D}_T) + \min\{\mathbb{E}_{\mathcal{D}_S}\left[|f_S(x) - f_T(x)|\right], \mathbb{E}_{\mathcal{D}_T}\left[|f_S(x) - f_T(x)|\right]\}$$

where d_{TV} denotes the total variational distance. Of course, theoretical results of this form motivate the use of source domains with input distributions that have small divergence to the input distribution of the target domain. Instead, we propose the selection of source domains based on *topological similarity*. In other words, we aim to find a source domain whose bottleneck distance to the target domain is small.

We do so for two reasons. Firstly the bottleneck stability theorem tells us that the total variational distance between distributions is lower bounded by the bottleneck distance between their corresponding persistence diagrams. As a result, low bottleneck distance is a necessary condition for low total variational distance between distributions. Secondly, we conjecture that learning the correct topology for the target domain forms the main difficulty when attempting to generalise well.

Such insights present a simple scheme for learning the predictive function f_T. First, we identify feature mappings for which the input distributions of both the source and target domains are close in terms of bottleneck distance. Given this feature mapping, we then apply homeomprhisms to try to reduce the distributional divergence further, whilst preserving the topological similarities between the distributions.

In what follows, we will outline a computationally cheap method for transfer learning such a homeomorphism. As previously mentioned, we assume that we have access to labelled training examples from both the source and target domains and denote by D_S the training sample for the source domain and D_T the training sample from the target domain.

Additionally, let $\phi : X \to \mathbb{R}^n$ denote a feature map learned from D_S. For example, ϕ could be the outputs from an intermediate layer of a deep neural network trained to classify examples in the source domain. We assume that the distributions of feature mappings Φ_S and Φ_T are similar topologically. That is, Φ_S and Φ_T are close in bottleneck distance. Our goal is then to reduce the total variational distance between Φ_S and Φ_T whilst keeping the topological distance small. As a result, from now on, we restrict our attention to homeomorphisms on the feature space. More specifically, our goal is to learn a homeomorphism which reduces the total variational distance between Φ_S and Φ_T. Of course, we cannot hope to optimise over all possible homeomorphisms, as this is tantamount to optimising over all continuous bijective open mappings. Instead, we choose to focus on a restricted family of homeomorphisms. More precisely, we consider the set of linear homeomorphisms, i.e. the set of full rank matrix transformations on the space $\mathbb{R}^d$.

4.1 Learning a Linear Homeomorphism

In order to learn a reasonable linear homeomorphism we employ a randomised algorithm as follows. First of all, using both the training sets we estimate the probability densities p_S and p_T of both Φ_S and Φ_T. In our experiments, we employ kernel density estimation for this purpose. Let $\hat{p}_S$ and $\hat{p}_T$ denote the kernel density estimators of p_S and p_T respectively. Moreover, let $\phi(D_T) = \{\phi(x_i)\}_{i=1}^m$ denote the set of feature vectors generated by the domain training set D_T.

Since computing the total variation distance between two continuous densities requires integrating over the entire feature space, which is computationally intractable for problems of high dimension, we instead consider the same problem for a pair of related discrete distributions with support $\phi(D_T)$. More specifically we compute the discrete distribution $\tilde{p}_S$ by considering only the values of the density $\hat{p}_S$ and renormalising:

$$\tilde{p}_S(\phi_i) = \frac{\hat{p}_S(\phi_i)}{\sum_{j=1}^m \hat{p}_S(\phi_j)}$$

Of course, $\tilde{p}_T$ is defined in a similar fashion. Note that computing the total variation distance between $\tilde{p}_S$ and $\tilde{p}_T$ is simple:

$$d_{TV}(\tilde{p}_S, \tilde{p}_T) = \max\left\{ \sum_{\phi_i \,:\, \tilde{p}_S(\phi_i) \geq \tilde{p}_T(\phi_i)} \left(\tilde{p}_S(\phi_i) - \tilde{p}_T(\phi_i)\right), \sum_{\phi_i \,:\, \tilde{p}_S(\phi_i) < \tilde{p}_T(\phi_i)} \left(\tilde{p}_T(\phi_i) - \tilde{p}_S(\phi_i)\right) \right\}$$

Next we apply the following heuristic method for finding a good transformation (i.e., full rank matrix transformation). Let $u_i \in \mathbb{R}^{d+1}$ denote the vector $(\phi_i, \hat{p}_S(\phi_i) - \hat{p}_T(\phi_i))$ for each feature vector $\phi_i \in D_T$. Clearly,

$$d_{TV}(\tilde{p}_S, \tilde{p}_T) = \max \left\{ \sum_{i\,:\,u_{i,d+1} \geq 0}^{m} u_{i,d+1}, \sum_{i\,:\,u_{i,d+1} < 0}^{m} -u_{i,d+1} \right\}$$

Now, consider any matrix transform $A \in \mathbb{R}^{d+1 \times d+1}$ operating on the vectors u_i. We consider the following optimisation problem:

$$\begin{aligned} \min_{A} \quad & \max \left\{ \sum_{i\,:\,A_{d+1}u_i \geq 0} A_{d+1}u_i, \sum_{i\,:\,A_{d+1}u_i \leq 0} |A_{d+1}u_i| \right\} \\ \text{s.t.} \quad & A^T A = I \end{aligned} \tag{2}$$

This problem has the following interpretation. The most common idea to "bring" Φ_S closer to Φ_T is to fix the latter while apply a homeomorphism to the former, as this can be done by e.g., retraining the source model. However, the resulting homeomorphism will not be linear, and the cost of retraining is not cheap either, which is what we want to avoid. Instead, we rely on the following idea. Notice that if we consider the point cloud $\{u_i\}$ as a manifold, then the hyperplane spanned by $\{\phi_i\}$ divides that manifold in to "positive" and "negative" half spaces. As discussed above, the total variation distance between Φ_S and Φ_T is either the sum of the points on the positive side or the sum of the other points on the negative side. This implies that if a transformation of the manifold/point cloud $\{u_i\}$ can balance these two sides and make both of the sums small, then hopefully it will be a good heuristic for a good homeomorphism. This idea is formalised in the problem described in Eq. equation 2. We refer to this problem as MINMAXSUM.

Unfortunately MINMAXSUM is computationally hard. While the exact complexity class of MINMAXSUM is not known, we prove that solving a slightly more restrictive version is indeed APX-Hard. In particular, we define MINMAXSUM-K as follows:

$$\begin{aligned} \min_{A} \quad & \max_{S \in 2^{\{u_i\}}, |S| \leq K} \sum_{i \in S} A_{d+1}u_i \\ \text{s.t.} \quad & A^T A = I \end{aligned} \tag{3}$$

where $S \in 2^{\{u_i\}}$ is a set of u_i with maximum cardinality of K. It is clear that our MINMAXSUM problem is equivalent to the case of $K = \infty$ (or unbounded). We state the following:

Theorem 1. *MINMAXSUM-K for $K < \infty$ is APX-Hard.*

As MINMAXSUM-K is difficult to solve, we conjecture that MINMAXSUM is also a computationally hard problem. In light of this, we turn to heuristics with provable guarantees. In particular, we provide a probably approximately correct learning (PAC) guarantee for a randomised algorithm, which proceeds as follows.

To begin, notice that the main task is to identify the optimal A^*_{d+1}. The remainder of the matrix A will be generated by simply generating a orthanormal basis from starting with the chosen candidate vector A^*_{d+1}, by e.g., using the Gram-Schmidt orthonormalization process (Trefethen & Bau III, 1997). Given this, our randomised algorithm, called RMMS (for randomised minmax sum) first generates random unit vectors $n_1, \ldots, n_K$. The vectors n_k represent candidate choices for the variables A_{d+1}. In what follows, we will show that once the number of candidate vectors K is sufficiently large, there will be a candidate close in performance to the optimal vector A^*_{d+1}. Let $S_k = \sum_{i\,:\,n_k^\top u_i > 0} n_k^\top u_i$ denote the sum of inner products of n_k with all u_i such that $n_k^\top u_i > 0$. Let k^* denote the index with smallest of such sums, i.e. $k^* = \arg\min_k S_k$. The following theorem guarantees that. with high probability, n_{k^*} will perform similarly to A^*_{d+1}.

Theorem 2. *With parameters θ and K, RRMS returns a solution n^*_K which guarantees approximation error of*

$$\epsilon = \max_{S \in 2^{\{u_i\}}} \sum_{i \in S} n^*_K u_i - \min_{A_{d+1}} \max_{S \in 2^{\{u_i\}}} \sum_{i \in S} A_{d+1}u_i \leq \sqrt{2}m \sin(\theta/2)$$

with a probability of at least $\delta = \exp\left(-\frac{\sin^{k-1}(\theta)}{3\sqrt{d-1}}K\right)$.

Note that, in order for a linear operator A to be a homeomorphism, it is necessary and sufficient to ensure that A is invertible. Due to the randomised nature of RMMS, the an invertible matrix is returned almost surely. We discuss this observation more in Appendix A.3.

5 NUMERICAL RESULTS

We wish to demonstrate that transforming our data using the linear transformation can improve the performance of vanilla transfer learning. That is, we can significantly improve the performance of a pretrained model on a new task without any additional training. To do so, we apply the proposed linear homeomorphism which is learnt using RMMS to a pretrained source model as described in Figure 1. In particular, we take the the chosen pretrained model (e.g., ResNet50), and apply on both source and target datasets. We then take the feature map of each dataset, generated by the last representation layer. We estimate the probability density functions for the positive and negative cases using these feature maps, and use RMMS to compute a linear homeomorphism on the probability distribution function, approximated by Gaussian kernel density estimation (KDE). Finally we use a generative classifier to predict the labels.

5.1 EXPERIMENTAL SETUP

We demonstrate the efficacy of this approach on EfficientNetB3 (Tan & Le, 2019), ResNet50 (He et al., 2016), and InceptionV3 (Szegedy et al., 2016), all pretrained on Imagenet (Deng et al., 2009).[1] We transfer learn onto CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), MNIST (Deng, 2012), and the ISIC 2019 skin cancer datasets (Sreena & Lijiya, 2019).[2] Specifically, our scheme for setting up transfer learning tasks is as follows. We download a model that has been pretrained on ImageNet. Given a new dataset to transfer learn onto, we choose a specific class from that dataset and use it to create a binary dataset: the task is to identify whether an image is from the specified class. For each dataset, we do this for 10 different classes. We then compute the precision (proportion of correctly classified images) before and after applying our homeomorphism, and see a significant improvement in performance by applying our approach, as shown in Table 4.

5.2 NUMERICAL EVALUATION

We initially tested our vanilla transfer learning pipeline from Imagenet to CIFAR-10 and CIFAR-100. We found that we could improve accuracy (measured by F-score) by 3.69-times without training (Tables 1 and 4). This is mainly due to the fact that our method significantly improves the sensitivity of the classification process (the improvement is typically a 4 to 5-fold), while precision is not decreased much (33% decrease on average). A more detailed visualisation of the improvement in sensitivity can be seen in Figure 2 (due to space limitations we defer the visualisation of changes in precision to the appendix). This results imply that before applying our linear transformation, the pre-trained network would reject most of the data with positive label. After the applying the linear transformation, however, most of the positive labels will be correctly detected, while the number of false positive cases is still being kept moderately low.

However, ImageNet and CIFAR-10 have significant overlap in the classes; for example, airliner corresponds to airplane. In this case, it seems that our transformation helps map the CIFAR-10 feature vectors to a form more recognisable as ImageNet images. To challenge our approach more we next tried to transfer learn from ImageNet to MNIST. This is a much more challenging scenario, as no categories in ImageNet are relevant to identifying handwritten digits. Despite this, we find our approach still improves performance on average by by 73% (EfficientNetB3), 114% (Resnet50), and 91% (InceptionV3), respectively (see Tables 2 and 4 for more details).

Finally, we try transfer learning from ImageNet to the ISIC 2019 skin cancer dataset. Similarly to the MNIST case, this is also a challenging task. This is reflected when we inspect how ImageNet interprets our dataset: each image is simply classified as a tick (Figure 3). Despite this, our approach is still able to make an average additional improvement of 70.4% (Tables 3 and 4). For the latter 2 datasets, the trend is the same as in CIFAR-10 and CIFAR-100. That is, our method significantly improves the sensitivity of the classication task, while keeping the precision at an acceptable level.

[1] Weights downloaded from `https://keras.io/api/applications/`

[2] ISIC'19 is available under a CC-BY-NC licencse and ImageNet is available under an Apache 2 license, and the rest of the models/pretrained networks are available under an MIT license.

Table 1: Accuracy (F-score values) of EfficientNetB3, ResNet50, and InceptionV3 trained on ImageNet when used to classify the given category in CIFAR-10, before and after applying our linear homeomorphism module.

F-score		Air	Auto	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
EfficientNetB3	Before	0.290	0.350	0.172	0.257	0.296	0.187	0.322	0.353	0.318	0.661
	After	0.733	0.785	0.737	0.717	0.476	0.764	0.752	0.747	0.698	0.750
ResNet50	Before	0.240	0.457	0.141	0.102	0.115	0.260	0.175	0.142	0.194	0.628
	After	0.726	0.662	0.681	0.502	0.700	0.712	0.668	0.726	0.759	0.726
InceptionV3	Before	0.406	0.496	0.145	0.120	0.179	0.337	0.164	0.451	0.367	0.758
	After	0.745	0.758	0.630	0.668	0.712	0.662	0.745	0.706	0.737	0.757

Table 2: Accuracy (F-score values) of EfficientNetB3, ResNet50, and InceptionV3 trained on ImageNet when used to classify the given category in MNIST, before and after applying our linear homeomorphism module.

F-score		0	1	2	3	4	5	6	7	8	9
EfficientNetB3	Before	0.656	0.588	0.727	0.658	0.487	0.299	0.418	0.470	0.234	0.419
	After	0.808	0.490	0.806	0.770	0.806	0.754	0.730	0.779	0.844	0.791
ResNet50	Before	0.572	0.492	0.320	0.184	0.261	0.789	0.248	0.551	0.247	0.718
	After	0.740	0.891	0.782	0.651	0.730	0.804	0.723	0.771	0.787	0.745
InceptionV3	Before	0.644	0.434	0.347	0.395	0.326	0.272	0.252	0.517	0.197	0.822
	After	0.736	0.773	0.671	0.763	0.538	0.765	0.659	0.427	0.699	0.748

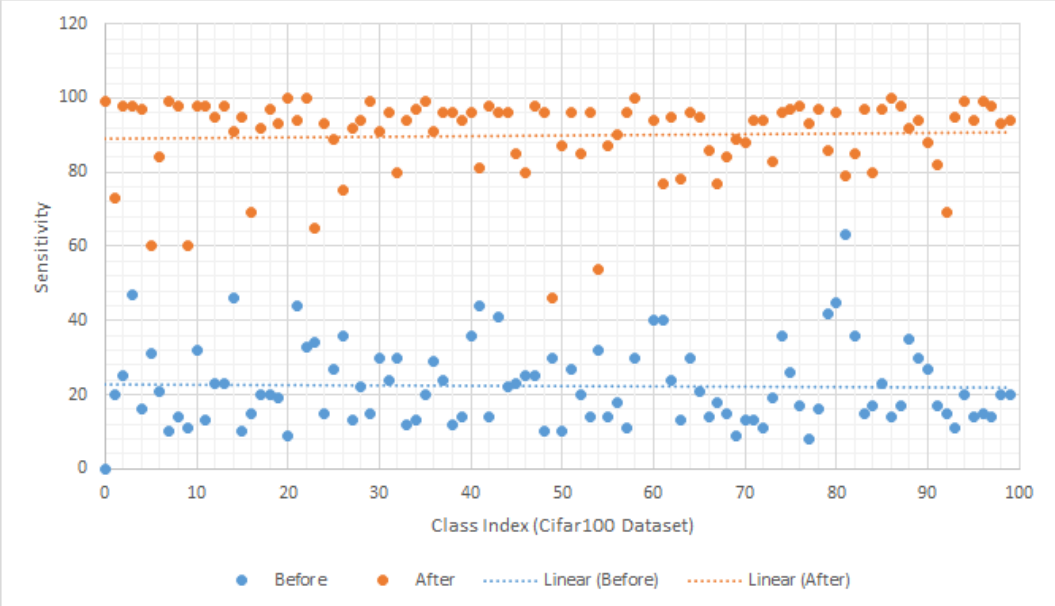
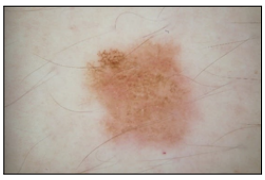


Figure 2: Sensitivity results on CIFAR-100 dataset, trained with the Resnet50 model. It shows the sensitivity of the network before and after using our linear transformation module.

Running time. We also compared the running time of our vanilla transfer learning method with that of a full retraining process. In particular, the time needed for the calculations of our linear transformation requires 68 seconds on average, whereas a complete retraining requires 1296 seconds.

Application to multiclass classification. While we only discuss the binary classification case in this paper, it is not difficult to extend our framework to the L-class classification tasks (where $L > 2$). This can be easily done by running our vanilla transfer learning pipeline depicted in Figure 1 (and

(a) Melanoma (ISIC'19)



(b) Nevus (ISIC'19)

(c) Tick (ImageNet)

Figure 3: We find that our approach works worst on the ISIC'19 skin cancer dataset. Under further examination, most classes in CIFAR-10 and CIFAR-100 coincide with ImageNet classes, whereas in ISIC'19 we find that skin cancer images are mostly associated with ticks which, although visually similar, will not lead to features representative of the presence of skin cancer. This demonstrates that our algorithm allows vanilla transfer learning to best utilise the information that is present in the initial dataset, but it cannot learn new information.

Table 3: Accuracy of EfficientNetB3, ResNet50, and InceptionV3 trained on ImageNet when attempting to classify the given category in the ISIC 2019 skin cancer dataset, before and after applying our learnt linear homeomorphism.

F-score		Melanoma	Nevus
EfficientNetB3	Before	0.491	0.648
	After	0.589	0.768
ResNet50	Before	0.287	0.454
	After	0.704	0.685
InceptionV3	Before	0.247	0.490
	After	0.622	0.662

Table 4: Percentage increase/decrease in F-score when performing vanilla transfer learning from ImageNet to the specified dataset with the specified model when adding in our linear homomeorphism.

Change (%)	CIFAR-10/100	MNIST	ISIC'19
EfficientNetB3	+117.6	+73.9	+19.2
ResNet50	+269.4	+114.1	+98.3
InceptionV3	+182.6	+91.2	+93.6

described in more detail in Appendix B) $k-1$ times, then use the class $\hat{l} = \arg\max_l \hat{P}_l(y = l|x)$ where $\hat{P}_l$ is the output probability distribution of the pipeline designed for label l.

Potential application domains of our approach. As discussed above, our method can significantly improve the sensitivity while keeping precision at an acceptable level in vanilla transfer learning. Furthermore, it can do this with a significantly less computational cost, compared to a full retraining process (20-fold faster on average). Therefore, it can be useful in computationally constrained and changing environments, where there is no time for retraining, and the goal is to correctly detect as many positive labels as possible. Possible applications of this type of system include (but is not limited to): mobile device based early disease/infection detection (e.g., TBC, or other viral infections), UAV based survivor detection in disaster response scenarios, and first line of outlier/malfunctioning detection in operation critical systems (e.g., smart traffic control, and other smart IoT systems).

6 Conclusions and Future Work

In this paper we investigate the information the topological distance between two domains provides about the statistical discrepancy between them. More precisely, we hypothesised that topological similarity forms a necessary condition for good vanilla transfer learning performance. Based on this intuition, we then proposed a computationally cheap linear homeomorphism which significantly improved the performance across a wide range of transfer learning tasks in our empirical experiments. Note that, in this work, only linear homeomorphisms were considered. One direction for future work is to find heuristics for other kinds of homeomorphism which may be able to capture richer geometric relationships. Similarly, the explicit connection between the topological distance and statistical divergence of distributions is not well understood. We believe that more empirical experimentation is required in order to develop a theory which fully explains the relationship between these two quantities. In addition, our investigations indicates that topological summaries may be useful in developing techniques for model selection and source domain selection, however more work is needed in this area.

REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6430–6439, 2019.

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18 (8):1–35, 2017.

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2309–2313. IEEE, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

P. Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 01 2015.

Mathieu Carrière, Marco Cuturi, and S. Oudot. Sliced wasserstein kernel for persistence diagrams. In *ICML*, 2017.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, T. Lacombe, Martin Royer, and Y. Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *AISTATS*, 2020.

Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. Toporeg: A topological regularizer for classifiers. *CoRR*, abs/1806.10714, 2018.

J. Clough, N. Byrne, I. Oksuz, V. A. Zimmer, J. A. Schnabel, and A. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, December 2006. doi: 10.1007/s00454-006-1276-5. URL https://doi.org/10.1007/s00454-006-1276-5.

Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 274–289. Springer, 2014.

Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126, 2006.

Thomas Davies, Jack Aspinall, Bryan Wilder, and Long Tran-Thanh. Fuzzy c-means clustering for persistence diagrams. *arXiv preprint arXiv:2006.02796*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Xiao Ding, Bibo Cai, Ting Liu, and Qiankun Shi. Domain adaptation via tree kernel based maximum mean discrepancy for user consumption intention identification. In *IJCAI*, pp. 4026–4032, 2018.

Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010.

Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. volume 108 of *Proceedings of Machine Learning Research*, pp. 1553–1563. PMLR, 2020.

Edward Gimadi and Ivan Rykov. Efficient randomized algorithm for a vector subset problem. In *International Conference on Discrete Optimization and Operations Research*, pp. 148–158. Springer, 2016.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

William H Guss and Ruslan Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *arXiv preprint arXiv:1802.04443*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems 30*, pp. 1634–1644. Curran Associates, Inc., 2017.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Mandar Dixit. Connectivity-optimized representation learning via persistent homology. volume 97 of *Proceedings of Machine Learning Research*, pp. 2751–2760, Long Beach, California, USA, 2019a. PMLR.

Christoph D. Hofer, Roland Kwitt, and Marc Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019b.

Chris Jones and Matt McPartlon. Spherical discrepancy minimization and algorithmic lower bounds for covering the sphere. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 874–891. SIAM, 2020.

Ker-I Ko and Chih-Long Lin. On the complexity of min-max optimization problems and their approximation. In *Minimax and Applications*, pp. 219–239. Springer, 1995.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Sinno Jialin Pan, and Philip S. Yu. Adaptation regularization: A general framework for transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 26(5):1076–1089, May 2014. ISSN 1041-4347.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *22nd Conference on Learning Theory, COLT 2009*, 2009.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International Conference on Machine Learning*, pp. 7045–7054. PMLR, 2020.

Gregory Naisat. *Tropical Algebra and Algebraic Topology of Deep Neural Networks*. PhD thesis, The University of Chicago, 2020.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pp. 7294–7305. PMLR, 2020.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Lorien Pratt. Reuse of neural networks through transfer. *Connection science (Print)*, 8(2), 1996.

Karthikeyan Natesan Ramamurthy, Kush Varshney, and Krishnan Mody. Topological data analysis of decision boundaries with application to model selection. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5351–5360. PMLR, 2019.

J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. A stable multi-scale kernel for topological machine learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4741–4748, 2015.

Bastian Alexander Rieck, F. Sadlo, and H. Leitte. Topological machine learning with persistence indicator functions. *ArXiv*, abs/1907.13496, 2019.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18) 2-7, 2018*, pp. 4058–4065. AAAI Press, 2018.

S Sreena and A Lijiya. Skin lesion analysis towards melanoma detection. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, volume 1, pp. 32–36, 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1395–1405, 2019.

Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

Fuping Wu and Xiahai Zhuang. Cf distance: A new domain discrepancy metric and application to explicit domain adaptation for cross-modality cardiac image segmentation. *IEEE Transactions on Medical Imaging*, 39(12):4274–4285, 2020.

Yuguang Yan, Wen Li, Hanrui Wu, Huaqing Min, Mingkui Tan, and Qingyao Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI*, volume 7, pp. 2969–2975, 2018.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

A PROOFS

A.1 PROOF OF THEOREM 1

Proof. Consider the following special case of MINMAXSUM-K: Let $K = 1$ and for each i, we have $||u_i||_2 = 1$ where $||\cdot||_2$ is the L_2 norm. In this case, the problem becomes

$$\begin{aligned} \min_{A_{d+1}} \quad & \max_i \sum_{i \in S} A_{d+1} u_i \\ \text{s.t.} \quad & ||u_i||_2 = 1 \\ & ||A_{d+1}||_2 = 1 \end{aligned} \tag{4}$$

which is equivalent to the spherical discrepancy problem, which is known to be APX-hard (Jones & McPartlon, 2020).

Now, for any $K > 1$, we reduce the problem of spherical discrepancy to MINMAXSUM-K as follows. For an arbitrary problem instance of the latter with m vectors, we generate another $(K-1)$ copy of each vector. These, together with the original ones, we have Km vectors. Consider the MINMAXSUm-K problem instance on these Km vectors. It is easy to prove that a solution of this MINMAXSUm-K instance is optimal if and only if it is also the optimal solution of the original spherical discrepancy instance. This concludes the proof. □

Remark. Consider matrix U which has its i^{th} column as u_i. It is easy to show that MINMAXSUM can be rewritten as follows:

$$\begin{aligned} \min_{x} \quad & \max_y x^T U y \\ \text{s.t.} \quad & ||x||_2 = 1 \\ & ||y||_\infty = 1 \end{aligned} \tag{5}$$

where $||\cdot||_\infty$ is the max-norm, and x and y are $(d+1)$-dimensional and m-dimensional vectors, respectively. The spherical discrepancy problem (i.e., MINMAXSUM-1), on the other hand, is a modified version of this where we replace the second constraint with $||y||_1 = 1$. That is, we take the L_1 norm of y instead of the max-norm. We conjecture that if we take the general form of $||y||_p = 1$ constraint with p going from 1 to ∞, the problem becomes more difficult in terms of computational complexity. Thus if with the L_1 norm constraint the problem is already APX-hard, we conjecture that MINMAXSUM is also APX-hard. In addition, based on the argument of Ko & Lin (1995), we further conjecture that MINMAXSUM is Π_2^{P}-hard, where Π_2^{P} denotes the second level of the polynomial-time hierarchy.

A.2 PROOF OF THEOREM 2

Proof. First assume that one of the candidate vectors is within angle θ of the optimal direction A^*_{d+1}. We denote this candidate vector by n. Note that, for any $u \in \mathbb{R}^d$ we have:

$$n^\top u - A^*_{d+1} u = (n - A^*_{d+1})^\top u \leq \|n - A^*_{d+1}\| \|u\|$$

and note that:

$$(n - A^*_{d+1})^\top (n - A^*_{d+1}) = \|n\|^2 - 2n^\top A^*_{d+1} + \|u\|^2 = 2 - 2\cos(\theta) = 2\sin^2(\theta/2)$$

Putting both observations together we have:

$$n^\top u - n^\top A^*_{d+1} \leq \sqrt{2}\|u\| \sin(\theta/2)$$

It then follows that:

$$S_{k^*} \leq \sum_{i\,:\,n_k^\top u_i > 0} n^\top u_i \leq \sum_{i\,:\,A^*_{d+1} u_i > 0} A^*_{d+1} u_i + \sqrt{2} m \sin(\theta/2)$$

Thus to prove the proposed result, we need only show that such a candidate n will be sampled with probability δ. This result was proved by Gimadi & Rykov (2016), as a result we defer the interested reader to the proof of Theorem 4 in Gimadi & Rykov (2016). □

A.3 NECCESSARY AND SUFFICIENT CONDITION ON OPERATOR A

It is well known that a necessary and sufficient condition on matrix A to be a homeomorphism is that A is invertible. For the sake of completeness we provides the statement and its proof below:

Theorem 3. *Let E be complete metric and finite dimensional linear space. A square matrix A can be seen as a linear map $A : E \to E$. Then A is a homeomorphism if and only if $detA \neq 0$.*

Proof. If A is a homeomorphism, then A is bijective and hence invertible ($detA \neq 0$). Conversely, if $detA \neq 0$, then A is bijective. Since E is finite dimensional, A is continuous. Then A^{-1} is continuous via Banach's isomorphism theorem. Therefore, A is a homeomorphism. □

Working on $\mathbb{R}^n$ a square matrix only has to be invertible to be a homeomorphism. However, in practice, even a random matrix is invertible. In probability's language, since $X = detA$ is a continuous random variable, the probability that $X = 0$ is 0 which means that we should not worry about the invertible condition for the matrix A. Also, ones should note that invertible matrix is full rank.

B ADDITIONAL EXPERIMENTAL DETAILS

Detailed description of experiments. All the pretrained models used in our experiment were sourced from Keras. For each dataset, we constructed binary classification tasks in the following manner. First, we select one of the many classes in the dataset. We call this class, the target class. Every training example belonging to the target class is given a positive labelling, whilst all remaining training examples are given a negative label. In order to have a balanced dataset, we select (uniformly at random) 1000 examples belonging to the target class, and 1000 examples belonging to other classes. Our methodology only differs for the ISIC'19 skin cancer dataset, as there not 2000 available images. We then pass each selected example through the pretrained network in question to compute the precision of the network on this new binary classification task. More specifically, we take the precision of the pretrained network to be the precision of the the most correct ImageNet class, that is, the class with the highest proportion of positive labellings.

For each input x_i from our selected task, let the output of the last layer (the feature vectors) be v_i. Now we generate two data sets $S_1^{y=1} = \{(v_1, 1), (v_2, 1), \ldots, (v_{1000}, 1)\}$, and $S_2^{y=1} = \{(v_1, t_1), (v_2, t_2), \ldots, (v_{1000}, t_{1000})\}$, in which t_i is the predicted value (1 for a prediction of the selected class, or 0 for any other prediction).

Next we estimate the PDF function p_1 for $S_1^{y=1}$ using Gaussian KDE. We estimate the PDF function p_2 for the subset S_2' of $S_2^{y=1}$ consisting of only $(v_i, 1)$. However, the v_i feature vector has very large dimension (around 1500). As a result, the density is so small so that it appears to be zero and therefore is not meaningful. To mitigate this issue, we reduce the dimensionality of the feature vector to 32 using principal component analysis, and perform min-max normalisation.

The TV norm of $(p_1 - p_2)$ is calculated by

$$\|p_1 - p_2\|_{TV} = \sum_{v_i \in J} [p_{1,nor}(v_i, 1) - p_{2,nor}(v_i, t_i)]$$

where $p_{i,nor}$ is a normalised version of p_i (i.e., to discretize a continuous pdf into a probability distribution over finite samples), and $J = \{v_i : p_{1,nor}(v_i, 1) - p_{2,nor}(v_i, t_i) > 0\}$.

RMSS transformation. Suppose the feature vector v_i is d-dimensional. Consider the $(d + 1)$ dimensional point $u_i = (v_i, p_{1,i} - p_{2,i})$ where $p_{1,i} = p_{1,nor}(v_i, 1)$ and $p_{2,i} = p_{2,nor}(v_i, t_i)$ for all feature vectors v_i. The process to compute the transformation is as follows.

1. Randomly and uniformly generate K unit vectors $n_1, n_2, \ldots, n_K$ in $\mathbb{R}^{d+1}$.
2. For each unit vector n_k calculate $n_k \cdot u_i$ for all u_i points, where $\cdot$ is the inner product. Now, let's choose the points u_i for which $n_k \cdot u_i > 0$, and sum up the inner product $n_k \cdot u_i$ over them. Let's S_k^+ be equal to this sum. That is $S_k^+ = \sum_i n_k \cdot u_i$ such that $n_k \cdot u_i > 0$. Similarly we define S_k^- to be the sum of $n_k \cdot u_i$ for $n_k \cdot u_i < 0$. We denote by $S_k = \max\{S_k^+, -S_k^-\}$. It is easy to prove that S_k is the TV distance between p_1 and p_2 after the transformation determined by n_k.

3. Among all the S_k, choose the smallest one: $k^* = \text{argmax}_k S_k$. Let denote n_{k^*} the corresponding unit vector.
4. Use the Gram-Schmidt orthogonalization algorithm over vectors n_k^* and $e_1, \ldots, e_{d+1}$ (where e_i is the i-th unit vector). After the orthogonalisation we obtain $d+1$ vectors, then ignore the vector having the smallest norm. Let the remainder be $q_1, q_2, \ldots, q_{d+1}$. Then our transformation matrix will be

$$R = [q_2^T, q_3^T, ...q_{d+1}^T, n_k^{*T}].$$

   We output a square matrix R with dimension $(d+1)x(d+1)$. In our experiments we reduced to $d = 32$ with PCA, so for us $R \in \mathbb{R}^{33}$.
5. We now obtain the PDF's after applying the matrix transformation. Let $P_1 = (v_i, p_{1,i})$ and $P_2 = (v_i, p_{2,i})$ where $p_{1,i} = p_{1,nor}(v_i, 1)$ and $p_{2,i} = p_{2,nor}(v_i, t_i)$.

   We take the last entry $p_1^R(v_i, 1)$ in each output vector $R \cdot P_1$. The entry $p_1^R(v_i, 1)$ is the image of $p_{1,nor}(v_i, 1)$ under the action of R. We then normalize $p_1^R(v_i, 1)$ for all $z_i = (v_i, 1) \in S_1^{y=1}$. Similarly, we obtain $p_{2,nor}^R(z_i)$ for all $z_i \in S_2^{y=1}$. The value $p_{2,nor}^R(z_i)$ serves as the joint probability $P(x_i, t_i = 1)$ after the action of the matrix R.

We also compute the total variation norm of $p_1 - p_2$ after transformation by the matrix R by

$$\|p_1 - p_2\|_{TV} = \sum_{v_i \in J} [p_{1,nor}^R(v_i, 1) - p_{2,nor}^R(v_i, t_i)]$$

where $J = \{v_i : p_{1,nor}^R(v_i, 1) - p_{2,nor}^R(v_i, t_i) > 0\}$.

Next we analyse the feature vectors for the negative labels. For each input x_i which is classified as negative, let the output of the last layer (the feature vectors) be a_i. Now as before we generate a data set $S_2^{y=0} = \{(a_1, b_1), (a_2, b_2), \ldots, (a_{1000}, b_{1000})\}$.

Finally, we repeat the same procedure to generate $P(v_i, t_i = 0)$. Having both $P(v_i, t_i = 0)$ anf $P(v_i, t_i = 1)$ calculated, we can use them to implement our classifier. In particular, if for a vector v we have $P(v, t = 1) > P(v, t = 0)$, then

$$P(t = 1|v) = \frac{P(v, t = 0)}{P(v)} > P(t = 0|v) = \frac{P(v, t = 0)}{P(v)},$$

and therefore we assign v to class 1, and *vice versa*.

Hardware details. We ran experiments on an internal machine which has the following specification: Core i7-10700K @ 3.8GHz 16 core CPU and NVIDIA GeForce RTX 3090 graphics card.

C Additional Numerical Results

As well as computing the accuracy for each task, we also computed the change in total variation distance before/after applying our transformation. Tables 5-6 display the total variation (TV) distance. Note that the TV distance decreases after RMMS is applied, as expected. Interestingly, although the TV distance decrease is huge in the skin cancer dataset, this does not correspond to a similarly large increase in precision.

We also present the change in the precision and F-score values before and after applying our linear transformation in transfer learning from ImageNet to CIFAR-100, using the ResNet50 network (Figures 4 and 5). We also include the F-score of the experiments run on EfficientNetB3 and InceptionV3 network architectures in Figures 6 and 7 (before and after applying our transformation method).

D Further Definitions in Topological Data Analysis

In this section, we give some more detailed descriptions of the elements of topological data analysis. As mentioned earlier, for a more detailed introduction to topological data analysis, we refer the reader to Edelsbrunner & Harer (2010).

Table 5: Total variation norm, computed as described in Appendix B, of EfficientNetB3, ResNet50, and InceptionV3 trained on ImageNet when attempting to classify the given category in CIFAR-10, before and after applying our learnt linear homeomorphism.

TV norm		Airp'ne	Autom'le	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
EfficientNetB3	Before	0.588	0.639	0.742	0.822	0.502	0.861	0.724	0.193	0.616	0.271
	After	0.032	0.105	0.119	0.194	0.020	0.164	0.003	0.037	0.056	0.005
ResNet50	Before	0.660	0.488	0.844	0.908	0.696	0.862	0.833	0.508	0.657	0.358
	After	0.066	0.053	0.205	0.016	0.043		0.076	0.114	0.103	0.045
InceptionV3	Before	0.484	0.569	0.857	0.945	0.670	0.787	0.851	0.219	0.642	0.235
	After	0.036	0.039	0.0234	0.156	0.028	0.013	0.187	0.028	0.024	0.015

Table 6: Total variation norm, computed as described in Appendix B, of EfficientNetB3, ResNet50, and InceptionV3 trained on ImageNet when attempting to classify the given category in ISIC'19, before and after applying our learnt linear homeomorphism.

TV norm		Melanoma	Nevus
EfficientNetB3	Before	0.741	0.205
	After	0.038	0.015
ResNet50	Before	0.769	0.480
	After	0.010	0.129
InceptionV3	Before	0.723	0.418
	After	0.144	0.081

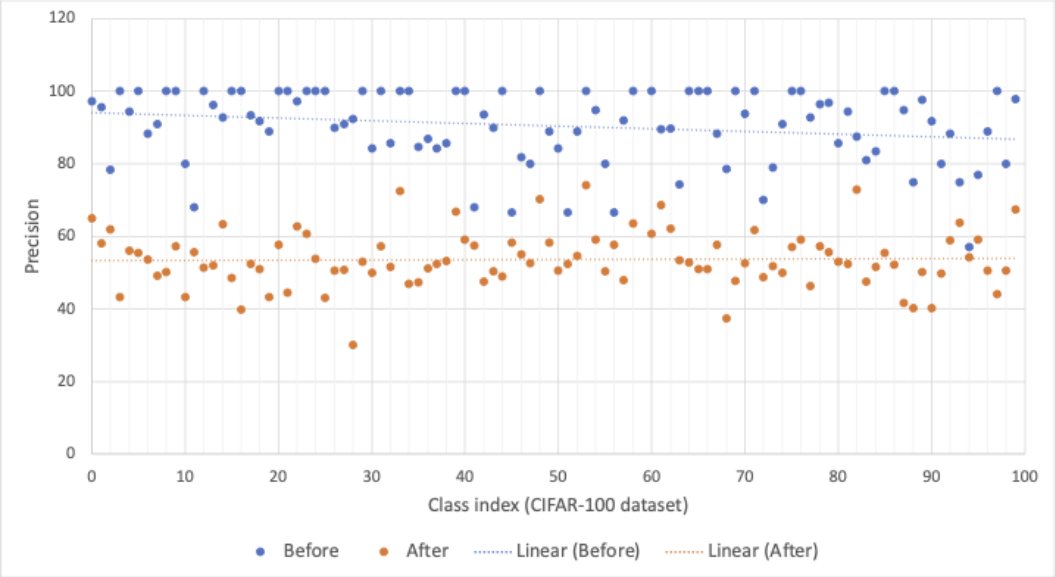


Figure 4: Precision results on CIFAR-100 dataset, trained with the Resnet50 model. It shows the precision of the network before and after using our linear transformation module.

Given a topological space $\mathbb{X}$, and an integer k, we denote the kth singular homology group of $\mathbb{X}$ by $\mathtt{H}_k(\mathbb{X})$, and the kth Betti number by $\beta_k(\mathbb{X}) = \dim(\mathtt{H}_k)$. Any continuous function $f : \mathbb{X} \to \mathbb{Y}$ induces linear maps $f_k : \mathtt{H}_k(\mathbb{X}) \to \mathtt{H}_k(\mathbb{Y})$ between the homology groups. The results which follow apply to the class of tame functions. Before we proceed with a definition of tame functions, we must first define the concept of a homological critical value.

Definition D.1. *Let $\mathbb{X}$ be a topological space and f a real function on $\mathbb{X}$. A homological critical value of f is a real number a for which there exists an integer k, such that for all sufficiently small*

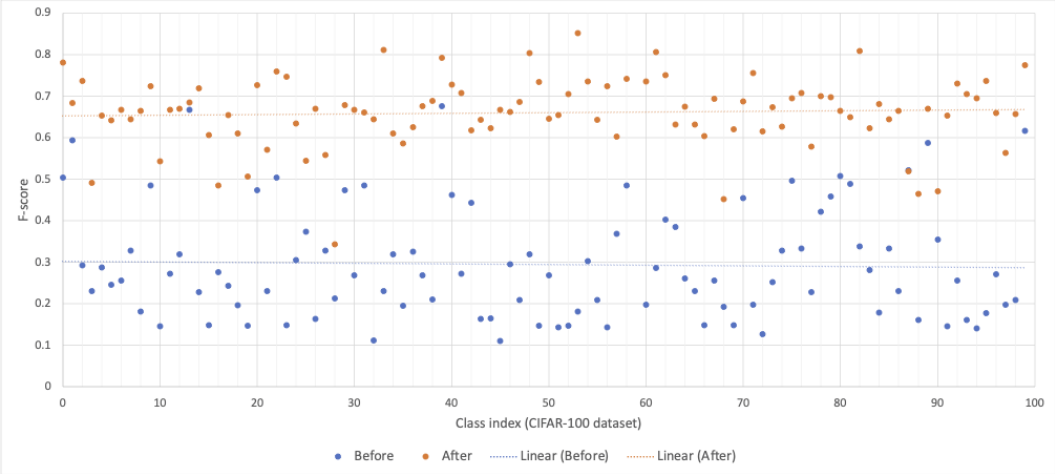


Figure 5: F-score values on CIFAR-100 dataset, trained with the Resnet50 model. It shows the precision of the network before and after using our linear transformation module.

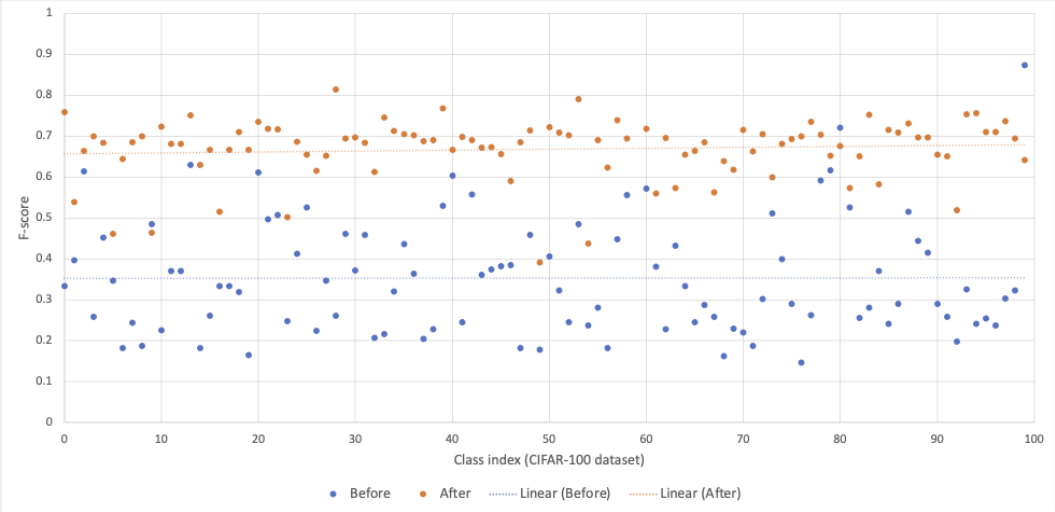


Figure 6: F-score values on CIFAR-100 dataset, trained with the EfficientNetB3 model. It shows the precision of the network before and after using our linear transformation module.

$\epsilon > 0$, *the map* $\mathrm{H}_k(f^{-1}(-\infty, a-\epsilon]) \to \mathrm{H}_k(f^{-1}(-\infty, a+\epsilon])$ *induced by inclusion is not an isomorphism.*

More generally speaking, homological critical values are levels at which the homology of the sublevel sets change. For Morse functions, homological critical values correspond with the standard definition of critical values. In other words, homological critical values of f correspond to the values of f at its critical points. We now proceed with the definition of tame functions.

Definition D.2. *A function* $f : \mathbb{X} \to \mathbb{R}$ *is tame if it has a finite number of homological critical values and the homology groups* $\mathrm{H}_k(f^{-1}(-\infty, a])$ *are finite dimensional for all* $k \in \mathbb{Z}$ *and* $a \in \mathbb{R}$.

Note that all Morse functions defined on compact manifolds are tame. Moreover, we write $F_x = \mathrm{H}_k(f^{-1}(-\infty, x])$, and for $x < y$, we write $f_y^x : F_x \to F_y$ to denote the map induced by the sublevel of set of x in that of y. Furthermore, let $F_x^y = \text{im } f_x^y$ denote the image of F_x in F_y. We refer to the groups F_x^y as the persistence homology groups. The persistence homology groups inform us about the topological relationships between sublevel sets.

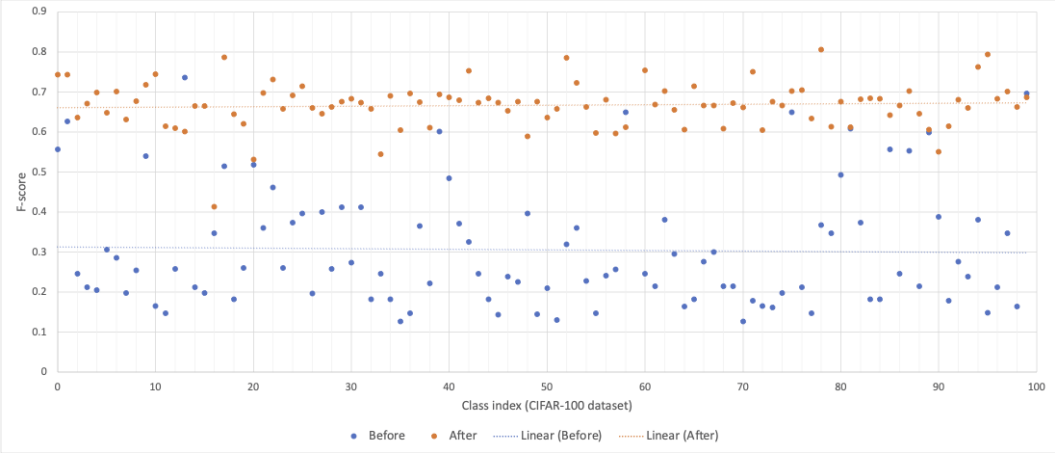


Figure 7: F-score values on CIFAR-100 dataset, trained with the InceptionV3 model. It shows the precision of the network before and after using our linear transformation module.

The persistent homology groups of a tame function can be succinctly represented by a planar drawing known as a persistence diagram. Let $f : \mathbb{X} \to \mathbb{R}$ be a tame function, $(a_i)_{i=1,\dots,n}$ its homological critical values, and $(b_i)_{i=1,\dots,n}$ an interleaved sequence, that is, $b_{i-1} < a_i < b_i$ for all i. We set $b_{-1} = a_0 = \infty$ and $b_{n+1} = a_{n+1} = +\infty$. For two integers $0 \leq i \leq j \leq n+1$ we define the multiplicity of a pair (a_i, a_j) by: $\mu_i^j = \beta_{b_{i-1}}^{b_j} - \beta_{b_i}^{b_j} + \beta_{b_i}^{b_{j-1}} - \beta_{b_{i-1}}^{b_{j-1}}$ where $\beta_x^y = \dim F_x^y$ denote the persistent Betti numbers for $\infty \leq x \leq y \leq \infty$. The multiplicity of each pair (a_i, a_j) is in fact the same for all possible interleavings, and thus is well-defined. We are now ready to formally define persistence diagrams.

Definition D.3. *The persistence diagram $D(f) \subset \bar{\mathbb{R}}^2$ of f is the set of points (a_i, a_j) counted with multiplicity μ_i^j for $0 \leq i < j \leq n+1$, union all points on the diagonal, counted with infinite multiplicity.*