

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

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

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. \square

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:

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

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)). \square

A.3 NECESSARY 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 provide 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 $A : E \rightarrow E$. Then A is a homeomorphism if and only if $\det A \neq 0$.*

Proof. If A is a homeomorphism, then A is bijective and hence invertible ($\det A \neq 0$). Conversely, if $\det A \neq 0$, 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. \square

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 $X = \det A$ 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, one 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 are 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 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 $S_1^{y=1} = \{(v_1, 1), (v_2, 1), \dots, (v_{1000}, 1)\}$, and $S_2^{y=1} = \{(v_1, t_1), (v_2, t_2), \dots, (v_{1000}, t_{1000})\}$, 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, \dots, 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: $k^* = \operatorname{argmax}_k S_k$. Let denote n_{k^*} the corresponding unit vector.
4. Use the Gram-Schmidt orthogonalization algorithm over vectors n_{k^*} and e_1, \dots, 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, \dots, q_{d+1} . Then our transformation matrix will be

$$R = [q_2^T, q_3^T, \dots, q_{d+1}^T, n_{k^*}^{*T}].$$

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 $S_2^{y=0} = \{(a_1, b_1), (a_2, b_2), \dots, (a_{1000}, b_{1000})\}$.

Finally, we repeat the same procedure to generate $P(v_i, t_i = 0)$. Having both $P(v_i, t_i = 0)$ and $P(v_i, t_i = 1)$ 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 |

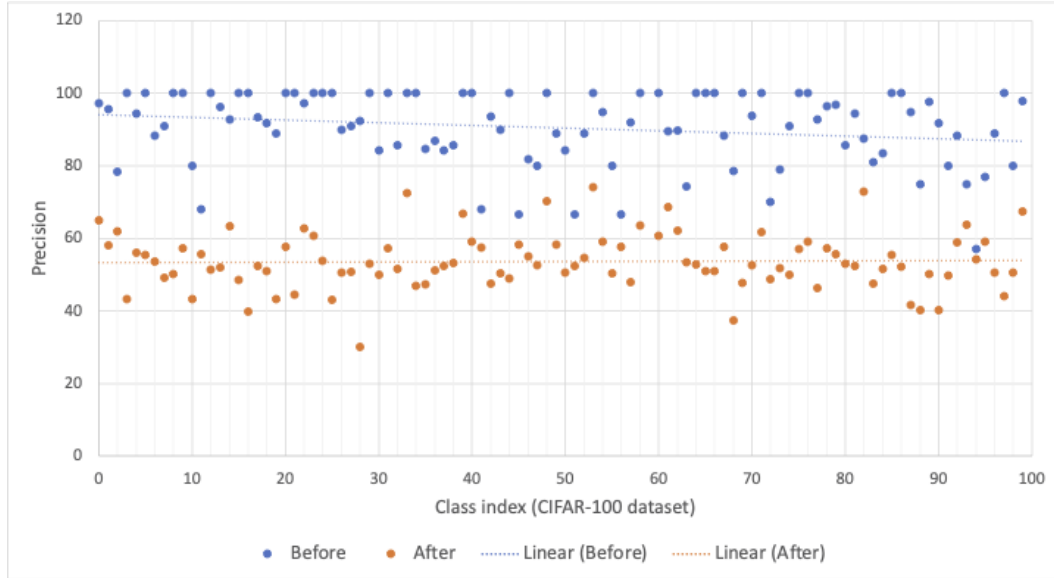


Figure 4: Precision results on CIFAR-100 dataset, trained with the Resnet50 model. It shows the precision of the network before and after using our linear transformation module.

Given a topological space \mathbb{X} , and an integer k , we denote the k th singular homology group of \mathbb{X} by $H_k(\mathbb{X})$, and the k th Betti number by $\beta_k(\mathbb{X}) = \dim(H_k)$. Any continuous function $f : \mathbb{X} \rightarrow \mathbb{Y}$ induces linear maps $f_k : H_k(\mathbb{X}) \rightarrow H_k(\mathbb{Y})$ 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

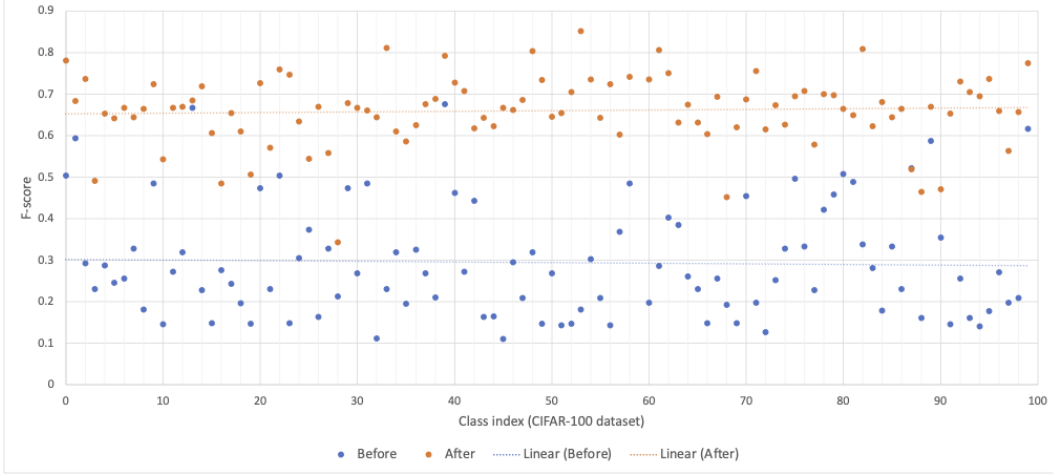


Figure 5: F-score values on CIFAR-100 dataset, trained with the Resnet50 model. It shows the precision of the network before and after using our linear transformation module.

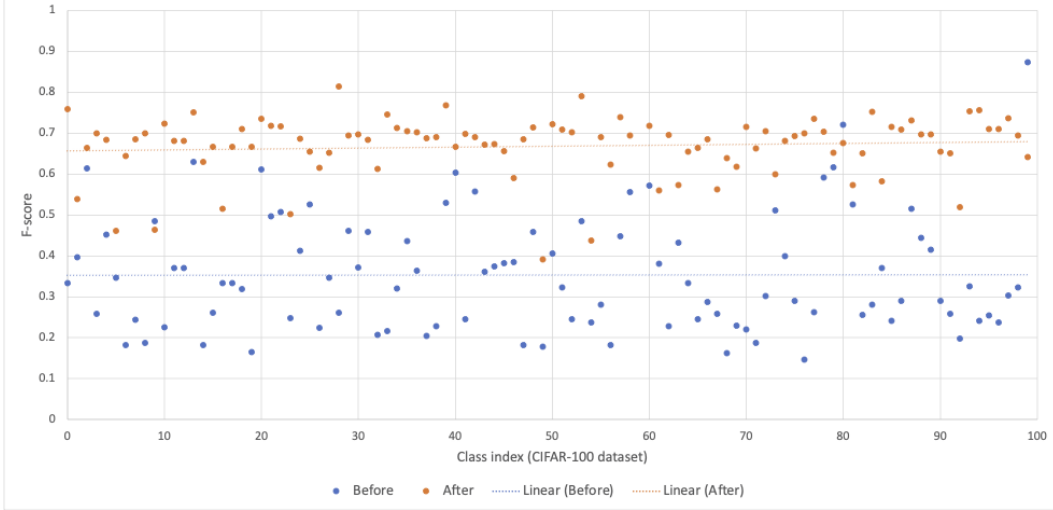


Figure 6: F-score values on CIFAR-100 dataset, trained with the EfficientNetB3 model. It shows the precision of the network before and after using our linear transformation module.

$\epsilon > 0$, the map $H_k(f^{-1}(-\infty, a - \epsilon]) \rightarrow H_k(f^{-1}(-\infty, a + \epsilon])$ 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 $f : \mathbb{X} \rightarrow \mathbb{R}$ is tame if it has a finite number of homological critical values and the homology groups $H_k(f^{-1}(-\infty, a])$ 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 $F_x = H_k(f^{-1}(-\infty, x])$, and for $x < y$, we write $f_x^y : F_x \rightarrow F_y$ 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.

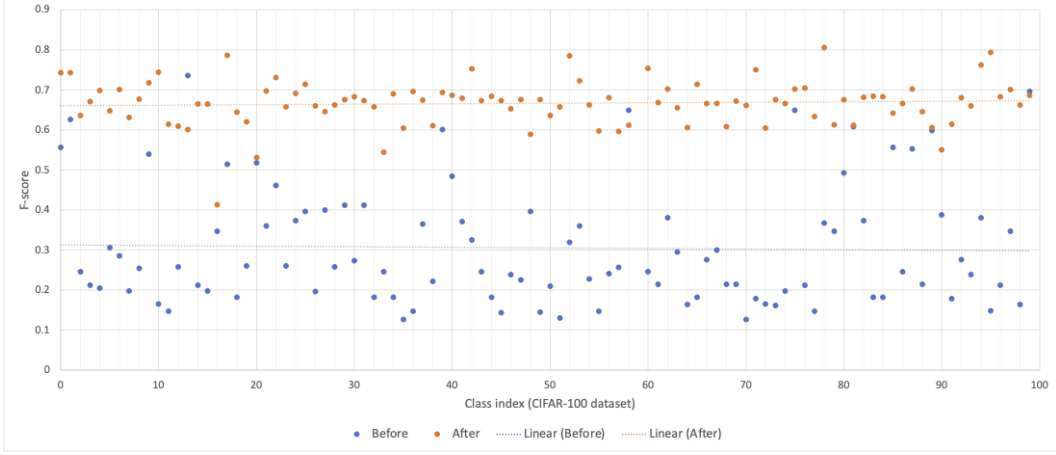


Figure 7: F-score values on CIFAR-100 dataset, trained with the InceptionV3 model. It shows the precision of the network before and after using our linear transformation module.

The persistent homology groups of a tame function can be succinctly represented by a planar drawing known as a persistence diagram. Let $f : \mathbb{X} \rightarrow \mathbb{R}$ 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.*