RETHINKING PERSISTENT HOMOLOGY
FOR VISUAL RECOGNITION

Ekaterina Khramtsova[1]**, Guido Zuccon**[1]**, Xi Wang**[2]**, Mahsa Baktashmotlagh**[1]
[1]University of Queensland , Brisbane, Australia,
[2]Neusoft, China
{e.khramtsova,g.zuccon,m.baktashmotlagh}@uq.edu.au
wxi@neusoft.com

ABSTRACT

Persistent topological properties of an image serve as an additional descriptor providing an insight that might not be discovered by traditional neural networks. The existing research in this area focuses primarily on efficiently integrating topological properties of the data in the learning process in order to enhance the performance. However, there is no existing study to demonstrate all possible scenarios where introducing topological properties can boost or harm the performance. This paper performs a detailed analysis of the effectiveness of topological properties for image classification in various training scenarios, defined by: the number of training samples, the complexity of the training data and the complexity of the backbone network. We identify the scenarios that benefit the most from topological features, e.g., training simple networks on small datasets. Additionally, we discuss the problem of topological consistency of the datasets which is one of the major bottlenecks for using topological features for classification. We further demonstrate how the topological inconsistency can harm the performance for certain scenarios.

1 INTRODUCTION

Topological Data Analysis (TDA) is a mathematical framework that aims to extract robust geometrical and topological information from high-dimensional data. Recently TDA has experienced an increasing interest from the machine learning community and has proven valuable in a broad range of applications, including shape recognition Hofer et al. (2017), graph Carriere et al. (2020) and image classification Dey et al. (2017), adversarial attacks and model interpretability Moor et al. (2020).

One of the most common methods of calculating robust and stable topological features is persistent homology Edelsbrunner & Harer (2010), Zomorodian & Carlsson (2005). In a nutshell, persistent homology monitors the evolution of the topology of the space throughout a filtration. The output of persistent homology represents a multiset, that summarizes the life and death of topological invariants of different dimensions (i.e., connected components, holes).

A naive way of integrating topological properties into the machine learning pipeline is to apply persistent homology on each input sample and approximate the resulting multiset to form a fixed-sized vector Qaiser et al. (2019); Chung et al. (2018). The drawback of this approach is that the vectorization of the multi-set is non-differentiable, which makes the topological representations task-agnostic. To overcome this limitation, efficient ways of utilizing topology are proposed which work by either imposing certain topological priors through the loss function Clough et al. (2020); Byrne et al. (2021), or by integrating topology as a layer in neural network Hofer et al. (2017); Carriere et al. (2020): the methods of Clough et al. (2020); Byrne et al. (2021) enforce segmentation masks to have a certain number of connected components and holes Clough et al. (2020); Byrne et al. (2021); the method of Gabrielsson et al. (2020) imposes generated images to have a predefined topological structure; and proposed recently, the state-of-the-art topological network layer Hofer et al. (2017); Carriere et al. (2020); Kim et al. (2020) integrates the topology as a fully differentiable layer which can be placed in any part of the main network.

The methods that rely on imposing the topological priors are limited in the sense that the ground truth for the topology of the dataset is not always available in practice, except for some segmentation masks Clough et al. (2020), or for simplistic datasets such as MNIST. Moreover, all the existing methods rely on the assumption that the datasets are topologically consistent. In the context of image classification, this can be translated to a topological inter-class heterogeneity and intra-class homogeneity. In other words, the images that belong to the same class are assumed to be topologically similar, while being topologically distinct from the other classes. Topological consistency depends on the correct choice of the complex and a filtration, which requires a solid understanding of the topological structure of the samples.

In light of the above discussion, in this paper, we analyse the implications of having a topological invariance and discuss how to overcome this issue. We extensively study the effect of utilizing topological features for the task of image classification with respect to the dataset size and complexity, backbone network complexity, and integration method.

In doing so, we make the following contributions:

- We show that the influence of incorporating the topological layer in simple neural network architectures is bigger compared to the deeper networks, resulting in a noticeable performance improvement for the task of image classification.
- We analyse the importance of persistent topological features with respect to the dataset size. Demonstrated by our empirical results, when only a limited number of training samples is available, persistent topological features can significantly improve the overall performance of the classifier. As more data becomes available, the influence of the topological features on the main classifier decreases.
- We show that the classification of the images with more topologically diverse structure (e.g. medical images) can substantially benefit from incorporating topological features, when only a few training samples are available. This finding is particularly important for classification of rare diseases that have limited number of samples. In contrast, adding the layer to the main network might become harmful once there is enough data.

2 RELATED WORK

Topological statistics: The first efforts to introduce topological properties of the input samples to neural networks were in calculating various statistics on the persistence diagram and use them to train a stand-alone machine learning model. For example, Chung et al. (2018) train a SVM on persistence curves and persistence statistics for a skin disease analysis. Qaiser et al. (2019) train a Regression Forest on Betti curves. They further design an ensemble strategy to combine the predictions from the regression forest and a convolutional neural network and apply on patch classification and tumour segmentation.

The main drawback of topological statistics is that the transformations for their calculation are non-differentiable, and therefore the network cannot be trained end to end.

Loss function: A more efficient way to benefit from topological information is to enforce some ground truth knowledge on the receptive space. For example, in image segmentation with one object Clough et al. (2020) correct the masks generated by the autoencoder by combining MSE loss with topological loss, that enforce the mask to have one connected component and one hole. Byrne et al. (2021) similarly use the topological loss to correct segmentation masks for multi-class cardiac MRI.

A more generalised approach for segmentation correction was proposed by Hu et al. (2019), where the authors reduce the difference between the topology of the generated mask and those of the ground truth mask by minimizing a modified Wasserstein distance between their persistence diagrams.

Segmentation correction is not the only task, that can benefit from topological loss. For example, Moor et al. (2020) propose to minimize the difference between the topology of the input and latent spaces of the autoencoders. They argue that preserving the topology of the latent space helps improving interpretability of the autoencoder.

Finally, topological loss can be used in the context of noise reduction and data regularisation. Gabrielsson et al. (2020) demonstrate that imposing some topological priors helps to improve the reconstruction loss.

The biggest drawback of the topological loss function is that it requires a ground-truth knowledge of the topology of the receptive space, that is realistically only available for very simplistic datasets Gabrielsson et al. (2020) or for segmentation masks.

Topological layer: The first topological layer is described in detail by Hofer et al. (2019). The authors focus on a variant of Gaussian transformation on points from persistence diagrams and compare different projection types. The layer is used for graph classification and 2D and 3D shape recognition. Differently from other works, that construct Cubical complex for visual input, the authors define a heat-kernel signature of the object's surface mesh and apply a height function to create a filtration.

Zhao & Wang (2019) build a weighted kernel, constructed on persistence images representation and demonstrate its efficiency on a graph classification task. Kim et al. (2020) propose to use topological layer to improve the classification of noisy images. They construct a Cubical complex from input images and use persistence landscapes as a vectorization method. The main contribution of their work is the use of DTM (Distance to measure) function for creating a filtration.
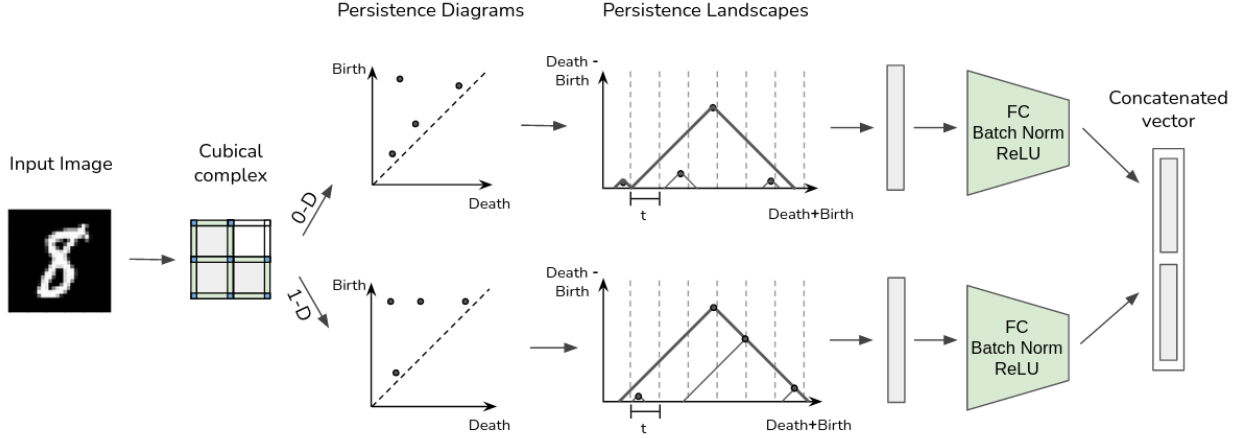
Figure 1: The architecture of the Landscape layer. A filtered cubical complex is constructed from the input image. Persistent homology is further applied to extract topological features, resulting in 0-D and 1-D persistence diagrams. Then, persistence landscapes are calculated to vectorize the multiset of points from persistence diagrams. The resulting fixed-sized vectors are separately fed into two Linear layers, each followed by Batch Normalisation and ReLu activation function. Finally, the output of the layers is concatenated to form the final vector.

Finally, Carriere et al. (2020) extend previous works by generalizing various vectorization methods for persistence diagrams, that they call PersLay. The authors construct a heat Kernel Signatures (HKS) of a graph and use Perslay for graph classification.

3   BACKGROUND

In this section, we will go through some concepts and definitions of topological data analysis (TDA).

Cubical Complex: Images consist of pixels, that can also be seen as 2D cubes and can thus be represented with a Cubical complex. It can be defined as follows:

An elementary interval is an interval of the form $I = [k, k+1]$ or the degenerate interval $I = [k, k]$ for some $k \in \mathbb{Z}$. *An elementary cube* is a Cartesian product of elementary intervals: $Q = I_1 \times ... \times I_d$. A set K is a *cubical complex* if it can be written as a finite union of elementary cubes. Naturally, for 2D images each pixel $X[i, j]$ is filled in with the elementary cube $Q_{i,j} = [i, i+1] \times [j, j+1], Q_{i,j} \in K_2$.

Persistent Homology: Let K be a simplicial complex. A sequence of simplicial complexes, such that $\emptyset = K_0 \subset ... \subset K_n = K$ is called a *filtration* of K. For a cubical complex the filtration is defined by increasing the threshold $\alpha \in [\min(X), \max(X)]$ and constructing cubical complexes for each value of α as follows: $K_\alpha = \{Q_{i,j} : X[i, j] > (1 - \alpha)\}$

Persistent homology is a method that keeps track of topological structures that persist throughout a filtration. Our input is 2D, therefore we restrict the analysis to 0D and 1D homological features, aka connected components and holes.

Persistence Diagrams: The resulting homology of the filtration is summarised in a persistence diagram $\mathcal{D} = \{\mathcal{D}_0, \mathcal{D}_1\}$, where each point (b_i, d_i) indicates that a topological feature appeared when the threshold $\alpha = 1 - b_i$ and disappeared when $\alpha = 1 - d_i$. In other words, the feature was *born* at scale b_i and *died* at scale d_i.

The points from persistence diagrams represent a multi-set and therefore are not directly suitable as an input to a Neural Network. For this reason they need to be vectorized first.

Diagram Approximation: There exist various diagram approximation methods Adams et al. (2017); Chazal et al. (2014)). One approach is to discretize the domain into fixed-sized bins and calculate the so called persistence landscape Bubenik (2015).

First, we define the binning interval T$= [t_0, ..., t_q] \in \mathbb{R}$ of equidistant points: $t_{n+1} - t_n = t \; \forall n \in \mathbb{N}$, where $t_0 = \min(\alpha), t_q = \max(\alpha)$. Then each point $p = (b_i, d_i) \in \mathbb{D}$ of the persistence diagram is mapped via the triangle point transformation as follows: $p \to [\Lambda_p(t_0), ..., \Lambda_p(t_q)]; \Lambda_p : t_i \to \max(0, y - |t_i - x|)$. Once all the points are mapped to $\mathbb{R}^q$, the values for each interval are sorted and only the k largest values for each binning point are kept. It is denoted as top-k persistence landscape.
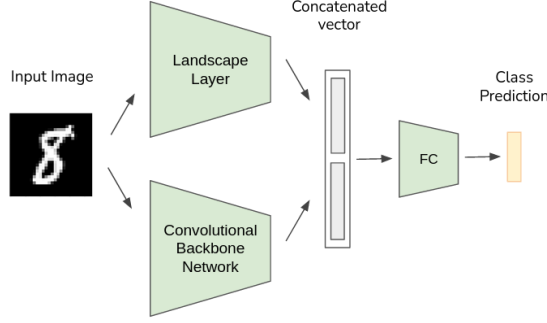
Figure 2: Topological Neural Network architecture. The Landscape Layer and the Convolutional Network are combined into one architecture and trained jointly by backpropagating through both branches, as the Landscape Layer is fully differentiable.
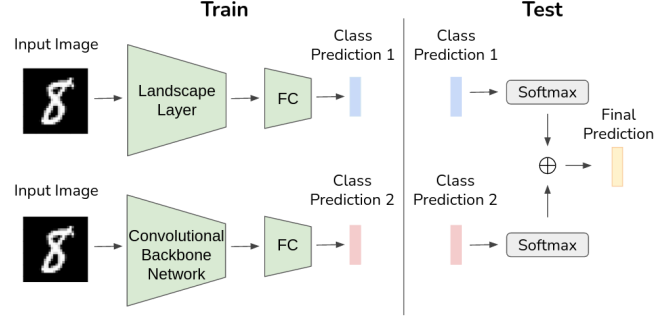
Figure 3: Ensemble strategy. The Landscape Network and the Convolutional Network are trained independently. In inference time, the predictions from two networks are combined to form a final prediction.

4 METHODOLOGY

Let X be the input image. The computation of topological features involves three main steps, as follows:

- Constructing a cubical complex from the input image: $X \rightarrow K$;

- Applying persistent homology by filtering by grayscale values to extract topological features: $K \rightarrow \{\mathcal{D}_0, \mathcal{D}_1\}$;

- Performing a vectorization of persistence diagrams: $\{\mathcal{D}_0, \mathcal{D}_1\} \rightarrow \{V_0^{kq}, V_1^{kq}\}$.

As a result, two fixed-length vectors of size $[kq]$ are obtained (one for each feature dimension), with q being the number of binning points and k being the the number of landscapes to be kept. q and k are the hyper parameters of the persistence landscapes that need to be defined in advance.

Landscape Layer. As a result of the aforementioned procedure we obtain two finite vectors that summarize the topology of the image. They can be further fed to a Neural Network. In contrast to previous approaches Hofer et al. (2019); Carriere et al. (2020); Kim et al. (2020), that concatenate 0D and 1D persistence landscapes and use the resulting vector as an input to a following layer, we construct separate linear layers for each dimension of the persistent features. This architecture decision is motivated by the difference in the distribution of 0D and 1D features and is particularly beneficial for images with a complex topological structure (e.g., medical patches). The architecture of the Landscape Layer is shown in Fig.1.

The Landscape Layer is fully differentiable and thus can be trained end-to-end using any gradient-based optimizer, e.g., Adam or SGD. However, the topological information alone is usually insufficient to make a competitive classifier. Instead, the topological layer is often used as an addition to the main backbone network.

Integration methods. One way to integrate the layer into a ML pipeline is shown in Fig.2. Here, the input batch is being passed through the Landscape Layer and a Convolutional Backbone Network, cut after the last convolutional layer. The output of the two is concatenated and is being further passed to a fully connected layer (FC), that in turn returns a final prediction. The described topological network, called TopoNet, is trained with a cross-entropy loss, that backpropogates through both branches.

The other way to use the layer within a ML pipeline is to train the backbone network and the landscape classifier separately, as shown in Fig.3. At inference time, the predictions from both networks p_1, p_2 are combined into a final prediction p as follows: $p = \text{argmax}(\text{softmax}(p_1) + \text{softmax}(p_2))$. We call this the ensemble strategy.

The influence of the layer is bigger when it is trained jointly with the main network, rather than separately. However, in some setups the influence of adding the layer to the main architecture might be rather harmful. In these cases, the ensemble technique is preferred, as TopoNet only affects the predictions it is certain about.

The purpose of this paper is to quantitatively analyse the performance of the Landscape Layer in various conditions. In particular, we want to evaluate its efficiency with respect to the following factors: **(1)** Training size; **(2)** Complexity of the backbone network; **(3)** Integration method. These aspects are investigated next.
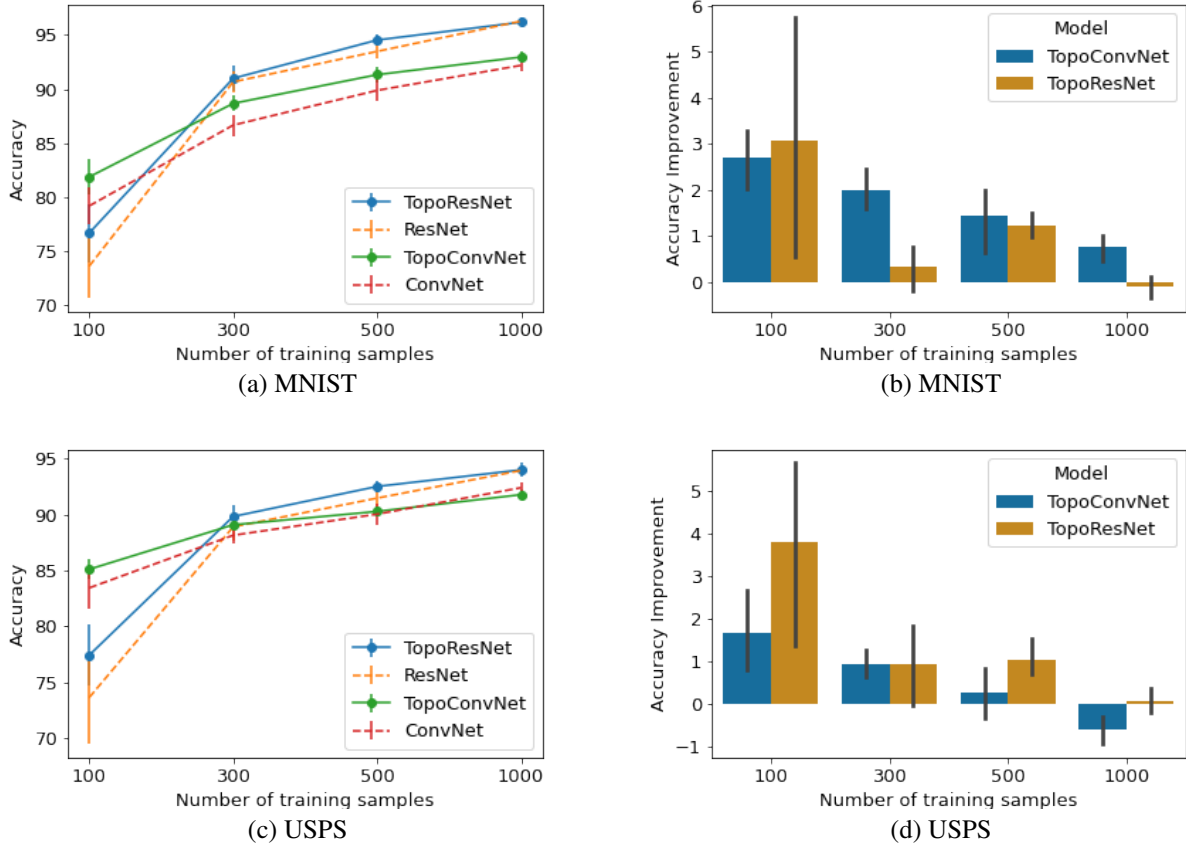
(a) MNIST

(b) MNIST

(c) USPS

(d) USPS

Figure 4: A comparison of ConvNet vs TopoConvNet on digits. X axis depicts the number of samples in the training set, Y axis depicts the test accuracy in percentages.

5 EXPERIMENTS

In this section, we extensively evaluate the impact of the topological properties on the classification task for three image datasets: MNIST LeCun et al. (2010), USPS Hull (1994), and a medical dataset Zhang et al. (2019). For a more detailed description of the datasets, networks and training hyperparameters, please refer to the extended version of our paper [1].

We start our experiments by comparing the influence of adding the topological layer with respect to the backbone network complexity (Section 5.1). We first study the effect of adding the Landscape Layer to a simple Convolutional Backbone Network (ConvNet), namely TopoConvNet. We further proceed by taking a more complex backbone network of ResNet, and observe if it can similarly benefit from the topological features.

The next set of experiments focuses on analysing the impact of using topology with respect to the dataset size (Section 5.2). We vary the number of samples available during training, and observe if adding the Landscape Layer to the main network can impact accuracy.

After performing an extensive study on digits datasets with images of a relatively simple topological structure, we further proceed to analyze the influence of the Landscape Layer on a more complex medical dataset (Section 5.3). The complexity of the medical dataset is reflected in a high variability of the topological structures of the patches, which results in significantly bigger influence of the landscape layer.

Finally, we compare the influence of incorporating topology for two integration methods (Section 5.4). We validate our hypothesis that the ensemble approach is a more reliable, but less efficient method than TopoNet.

---

[1]https://arxiv.org/abs/2207.04220

Dataset	ConvNet	TopoConvNet	ResNet	TopoResNet
MNIST	86.6±1.0	88.6±0.7	90.6±0.9	91.0±1.2
USPS	88.1±0.7	89.0±0.5	88.9±1.3	89.8±1.1

Table 1: Performance comparison with regards to the network complexity, and training size of 300 samples. The values represent the mean and a std of the classification accuracy over 10 folds.

5.1 NETWORK COMPLEXITY

To study the effectiveness of incorporating topology with respect to the complexity of the backbone architecture, we show the results of adding a Landscape Layer to a simple baseline of ConvNet. For this set of experiments, we set the training size to the lowest possible number of training samples $n = 300$, for which we get reasonably stable results in different runs of the algorithm.

The resulting test accuracy can be found in Table 1. We report the mean and the standard deviation over 10 runs of the experiment. The results show that for a small dataset, adding the Landscape Layer to ConvNet has a beneficial effect: it improves the accuracy by 2% on MNIST and by 1% on USPS.

Experiments, described above, demonstrated that the topological layer can significantly enhance the performance of a simple neural network. However, there is no existing analysis on the influence of using topology in a more complex deep neural network, such as ResNet, for a topologically simple dataset. Therefore, we perform a set of experiments to compare the performance of ResNet with its topological counterpart. The results from Table 1 reveal that the impact of the Landscape Layer is marginal on ResNet in comparison to the ConvNet. This confirms that incorporating topological information in the training stage helps improving the recognition accuracy, however, such information can become redundant for more complex backbone networks.

5.2 DATASET SIZE

In this section, we further investigate the influence of incorporating topological information of the dataset in the training stage with respect to different dataset sizes.

We vary the number of samples $n \in \{100, 300, 500, 1000\}$ available during training, and observe that the training size can significantly affect the performance. For small n, the performance is sensitive not only to the choice of the samples, but also to the initialization of the network. Therefore, for a fair comparison between the backbone network and its topological counterpart, we fix the initialisation of their common layers, as well as the data splits. Moreover, to ensure the stability of the reported results, the samples are drawn randomly 10 times for each training size n. The test set is always the same throughout these experiments.

To clarify our findings, in Fig.4(b,d) we further show the performance improvement with respect to training size as a pairwise difference between the accuracies of ConvNet vs. TopoConvNet, and ResNet vs. TopoResNet.

The resulting plots for MNIST can be found in Fig.4(a,b). The biggest improvement for ConvNet is achieved when n is small: over 2.5% on average for $n = 100$ and around 2% for $n = 300$. With increasing training size, the accuracy improvement of TopoConvNet decreases and falls below 1% for $n = 1,000$.

Analogous performance trends are observed while training on USPS as shown in Fig.4(c,d): the biggest accuracy improvement of 1.6% appears while training ConvNet on small subsets of data ($n = 100$). Similarly, augmenting the number of training samples leads to a decline in the influence of the Landscape layer. When n is big enough, the Landscape Layer does not help the performance of the main classifier, and in certain cases, e.g., when $n = 1,000$, it can even be harmful.

Unsurprisingly, the performance of ResNet and TopoResNet on the smallest n is very poor and even inferior to ConvNet: it quickly overfits on the training data and does not generalise well on the test data. As the training size increases, ResNet starts to significantly outperform ConvNet.

Note that, we are more interested in settings where ResNet is better than ConvNet: $n >= 300$, for MNIST and USPS. For both datasets, we observe the trend of having a positive effect of the Landscape Layer when n is small. Moreover, for USPS, when n is big enough, e.g., $n = 1,000$, the Landscape Layer does not harm the performance of the ResNet as much as the ConvNet. This may come from the fact that ResNet by itself produces more confident predictions compared to the layer. Note that, the influence of the Landscape Layer on ResNet does not have the same linear correlation with the training size as observed for ConvNet. For example, the average improvement on MNIST for $n = 500$ is bigger than for $n = 300$; similarly, the variance of the improvement on USPS for $n = 500$ is higher than for $n = 300$.
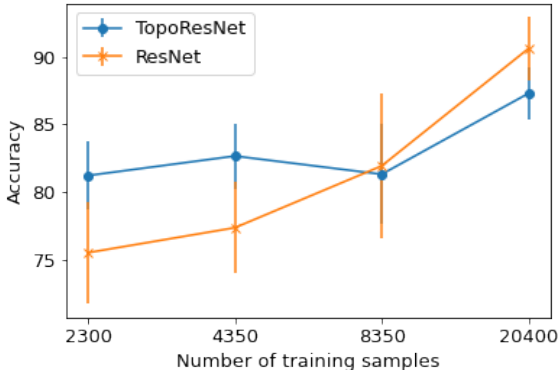
Figure 5: A comparison of ResNet vs TopoResNet, trained on Medical data. X axis depicts the number of samples in the training set, Y axis depicts the test accuracy in percentages.
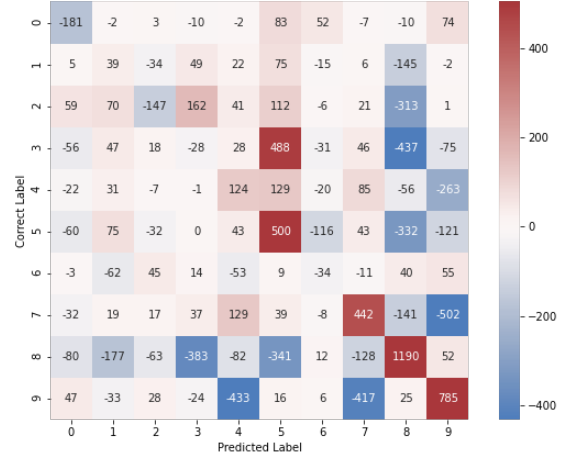

Figure 6: Confusion matrix improvement for MNIST with 100 training samples, calculated as conf_t - conf_c, with conf_t being the confusion matrix of ConvNet, and conf_c being the the confusion matrix of TopoConvNet.

Dataset	ResNet	TopoResNet
MNIST	97.85±0.16	97.87±0.33
USPS	95.78±0.37	95.79±0.38
Medical	75.57 ±3.75	81.25 ±2.47

Table 2: Performance comparison with respect to data complexity, and training size of $\approx 2K$ samples. The values represent the mean and a std of the classification accuracy over 10 folds.

5.3 DATA COMPLEXITY

In the previous sections, the experiments have been performed on the digits datasets, that consist of small images with simple topological structure. In this section, we report our results on the more topologically diverse medical dataset.

For the medical data, the training size n is defined by the number of Whole Slide Images (WSI) with $n \in \{5, 10, 20, 50\}$. Each WSI has annotated regions of different sizes, meaning that the actual number of patches in a training set vary depending on the training split. For example, 5 WSIs can indicate from 838 to 3,434 patches in the training set.

Since ConvNet is not deep enough to capture the complex structure of the medical images, we perform our experiments using the ResNet backbone. We compare the performance of ResNet and TopoResNet on the smallest $n = 5$ WSIs for the medical dataset. To analyse the effect of incorporating Landscape Layer with respect to data complexity, we further add the results of ResNet and TopoResNet on the same training size of $n = 2,300$ samples for MNIST and USPS datasets. The results shown in Table 2 demonstrate that for a small training size, the medical dataset significantly benefits from the additional information provided by the Landscape Layer. Using TopoResNet, we observe a large performance improvement (5.5%) for medical data in comparison to no improvement for the topologically simple digits datasets. This finding has a substantial practical application, as some rare diseases have only a few available WSIs.

On medical data, the influence of the Landscape layer, both positive and negative, is significantly larger (Fig.5). The complexity of the medical data is reflected in a high variability of the topological structures of the patches and thus results in more confident predictions. The biggest improvement is obtained when the training size is small: 5.7% on average for $n = 5$; and 5.3% on average for $n = 10$. However, when the training size is big enough, $n = 50$, the accuracy of TopoResNet drops on average by 3.3% compared to ResNet.

5.4 INTEGRATION METHOD

In this section, we analyze the effect of incorporating topological information in the training stage with respect to the integration method. When the training size is small, the topological layer trained as a part of the main network, aka TopoNet, has proven to significantly enhance the main classifier. However, it might become harmful once enough training samples become available. Such trend is justified by the fact that some of the images belonging to different

7

(a) Normal

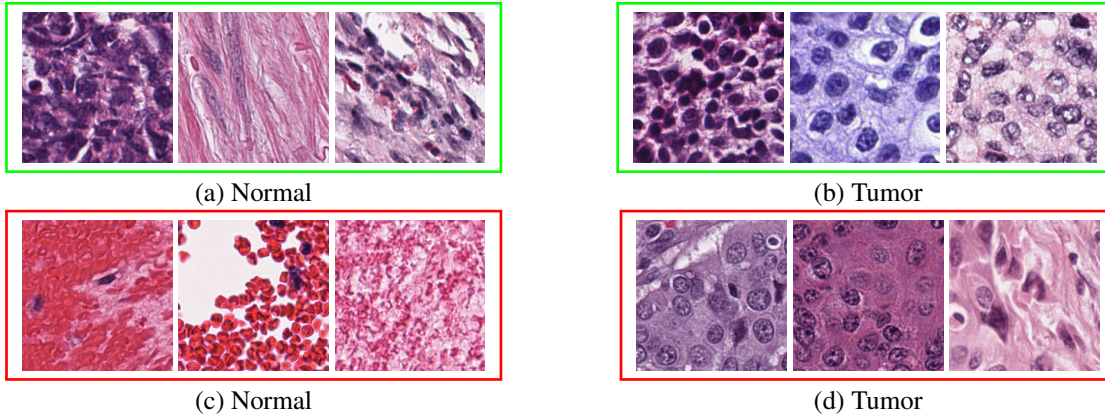
(b) Tumor

(c) Normal

(d) Tumor

Figure 7: Example of medical patches where adding topology helped (top) and harmed (bottom) the performance. Top: samples correctly classified by TopoResNet and misclassified by ResNet. Bottom: samples correctly classified by ResNet, but misclassified by TopoResNet.

Dataset	Backbone type	Net	TopoNet	Lanscape Net	Ensemble
USPS	ConvNet	92.4±0.49	91.78±0.45	49.73±0.77	92.7±0.58
Medical	ResNet	90.59 ±2.3	87.29±1.94	83.26±1.65	91.28±1.6

Table 3: Performance comparison with regards to the integration method. The values represent the mean and a std of the classification accuracy over 10 folds.

classes are topologically equivalent. Therefore, for a relatively efficient classifier, it might be preferable to only correct the predictions for topologically distinct images. This can be achieved via the Ensemble technique.

We select the setups where TopoNet showed inferior results to Net (i.e., the network without topological layer). For these setups, we construct the Landscape network, that consists of the Landscape Layer, followed by a fully connected layer. The Landscape Network is further trained using SGD. In inference time, the predictions from the backbone network are combined with predictions from the Landscape network, as illustrated in Section 4. The performance of the resulting Ensemble method is compared with the TopoNet in Table 3. The results demonstrate that unlike TopoNet, the Ensemble never harms the performance of the main classifier, and achieves slight improvements of around 0.7% for the medical dataset.

6 DISCUSSION AND FUTURE WORK

MNIST and USPS are composed of digits, distributed among 10 classes. Due to the simplicity of these datasets, we can infer the ground truth of the topology of each class, valid for the majority of the images. For example, all digits have only one persistent connected component; digits 1,3,4,5 and 7 have zero holes, digits 0,6,9 have one hole, and digit 8 has two holes. Note that this ground truth is not available during training; it is only used to analyse the results.

The accuracy improvement described in Section 5 comes from the Landscape Layer. In order to validate the origins of this improvement, we analyze the changes in per-class predictions. Fig.6 depicts the difference between the confusion matrix of TopoConvNet and the one of ConvNet. It shows that the biggest enhancement is achieved for the digit 8, as it is the only class with the unique topological signature (one connected component and two holes). The performance for digits 9, 7 and 5 has also increased. For example, TopoConvNet better recognises digit 9 from digits 4 and 7 as digit 9 has one hole, while digits 4 and 7 have no holes. On the other hand, digits 0 and 2 have experienced a slight deterioration in their classification. These experimental results are consistent with the ground truth, described above.

The improvement from adding topology layer to the network trained on the medical dataset comes primarily from eliminating false positives (Fig.7, a). TopoResNet is more robust to color variations; it successfully detects the connectivity of the cells and recognises tumorous cases with distinct and well-separated formations (Fig.7, b). Concurrently, TopoResNet tends to confuse some normal patches as being tumorous due to their topological similarity (Fig.7, c), and misclassify patches with low color variation (Fig.7, d).

Overall, we identify the biggest discovered issue to be the topological inconsistency of the datasets. To tackle this problem, one might design an optimal dataset-specific transformation to extract consistent topological features.

REFERENCES

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), 2017. URL http://jmlr.org/papers/v18/16-337.html.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(3), 2015. URL http://jmlr.org/papers/v16/bubenik15a.html.

Nick Byrne, James R. Clough, Giovanni Montana, and Andrew P. King. A persistent homology-based topological loss function for multi-class cnn segmentation of cardiac mri. In *Statistical Atlases and Computational Models of the Heart. M&Ms and EMIDEC Challenges*, 2021. ISBN 978-3-030-68107-4.

Mathieu Carriere, Frederic Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020.

Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, and Larry Wasserman. Stochastic convergence of persistence landscapes and silhouettes. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, 2014. ISBN 9781450325943. doi: 10.1145/2582112.2582128. URL https://doi.org/10.1145/2582112.2582128.

Y. Chung, C. Hu, A. Lawson, and C. Smyth. Topological approaches to skin disease image analysis. In *IEEE International Conference on Big Data*, 2018. doi: 10.1109/BigData.2018.8622175. URL https://doi.ieeecomputersociety.org/10.1109/BigData.2018.8622175.

J. Clough, Ilkay Öksüz, Nicholas Byrne, V. Zimmer, J. Schnabel, and A. P. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Tamal Krishna Dey, Sayan Mandal, and William Varcho. Improved Image Classification using Topological Persistence. In *Vision, Modeling and Visualization*, 2017. ISBN 978-3-03868-049-9. doi: 10.2312/vmv.20171272.

Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010. ISBN 978-0-8218-4925-5.

Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, 2020.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, 2017.

Christoph D. Hofer, Roland Kwitt, and Marc Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126), 2019. URL http://jmlr.org/papers/v20/18-358.html.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems*, 2019.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5), 1994. doi: 10.1109/34.291440.

Kwangho Kim, Jisu Kim, Manzil Zaheer, Joon Kim, Frederic Chazal, and Larry Wasserman. Pllay: Efficient topological layer based on persistent landscapes. In *Advances in Neural Information Processing Systems*, 2020.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL `http://proceedings.mlr.press/v119/moor20a.html`.

Talha Qaiser, Yee Tsang, Daiki Taniyama, Naoya Sakamoto, Kazuaki Nakane, David Epstein, and Nasir Rajpoot. Fast and accurate tumor segmentation of histology images using persistent homology and deep convolutional features. *Medical Image Analysis*, 55, 2019. doi: 10.1016/j.media.2019.03.014.

Zizhao Zhang, Pingjun Chen, Mason McGough, Fuyong Xing, Chunbao Wang, Marilyn Bui, Yuanpu Xie, Manish Sapkota, Lei Cui, Jasreman Dhillon, Nazeel Ahmad, Farah Khalil, Shohreh Dickinson, Xiaoshuang Shi, Fujun Liu, Hai Su, Jinzheng Cai, and Lin Yang. Pathologist-level interpretable whole-slide cancer diagnosis with deep learning. *Nature Machine Intelligence*, 1, 2019. doi: 10.1038/s42256-019-0052-1.

Qi Zhao and Yusu Wang. Learning metrics for persistence-based summaries and applications for graph classification. In *Advances in Neural Information Processing Systems*. Curran Associates Inc., 2019.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33: 249–274, 02 2005. doi: 10.1007/s00454-004-1146-y.