Convolutional Persistence as a Remedy to Neural Model Analysis

Ekaterina Khramtsova
University of Queensland

Guido Zuccon
University of Queensland

Xi Wang
Neusoft

Mahsa Baktashmotlagh
University of Queensland

Abstract

While deep neural networks are proven to be effective learning systems, their analysis is complex due to the high-dimensionality of their weight space. Persistent topological properties can be used as an additional descriptor, providing insights on how the network weights evolve during training. In this paper, we focus on convolutional neural networks, and define the topology of the space, populated by convolutional filters (i.e., kernels). We perform an extensive analysis of the topological properties of the convolutional filters. Specifically, we define a metric based on persistent homology, namely, Convolutional Topology Representation, to determine an important factor in neural networks training: the generalizability of the model to the test set. We further analyse how various training methods affect the topology of convolutional layers.

1 Introduction

The learning process of neural models evolves around finding an optimal combination of weights that, given an input data, produces a certain output, e.g., a class. The distribution of the weights may define various properties of the model such as model generalizability (Atiya and Ji, 1997). Extracting these properties requires analysing a complex multi-dimensional weight space, which is not a trivial task.

One way to approximate the weight space in neural networks is through the means of topology and persistent homology. Recent studies have explored the potential benefit of analysing topological features for network weights (Rieck et al., 2019; Carlsson and Gabrielsson, 2020; Love et al., 2020; Birdal et al., 2021) and network activations (Corneanu et al., 2019, 2020; Lacombe et al., 2021). For example, Rieck et al. (2019) define a complexity measure of a fully connected layer by employing its graph structure and further use it as an early stopping criterion.

In this work, we focus on convolutional layers. To the best of our knowledge, (Carlsson and Gabrielsson, 2020; Love et al., 2020) are the only existing studies of the geometry of convolutional layers. Carlsson and Gabrielsson (2020) build a complex on 100 trainings of the same model and show the existence of recurring topological patterns in the space of trained convolutional filters, particularly in the first layer. Such construction is computationally cumbersome as it requires to train the same network 100 times. Continuing this line of work, Love et al. (2020) attempt to improve the generalisability of CNNs by restricting the weights of the first convolutional layers to lie on a topological manifold. While demonstrating improvement in generalisability of the network to unseen datasets, the authors report only 30% accuracy when generalizing from MNIST to SVHN.
The work of Carlsson and Gabrielsson (2020) may be closest in spirit to our work. However, our method of extracting and summarizing topological features allows for a per-model comparison and gives rise for a more detailed analysis of the topological space of convolutional filters.

In this work, we explore the manifolds populated by the learnable filters, that form convolutional layers. The filters are projected to a low-dimensional manifold in Euclidean space, from which we construct Vietorios-Rips complexes (Zomorodian (2010)) and extract 1D topological features by applying persistent homology, summarized in a persistence diagram. By aggregating the topological features, we introduce a new compact metric, namely, Convolutional Topology Representation (ConvTopRep), that summarises the topology of a convolutional layer. ConvTopRep is calculated as a 1-Wasserstein distance of the points in persistence diagram to its diagonal. We further demonstrate how it can be used to explain the generalizability of the model.

In particular, we address two challenges:

- Model Selection. In many real-world applications, the train and test data come from different distributions (Gulrajani and Lopez-Paz (2021)). A wide range of methods and models have been introduced to tackle this domain shift problem (Wang et al. (2021b)). However, selecting the best model, which can perform

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

equally well on the training and unseen test data, remains a highly challenging task. We argue that our proposed ConvTopRep is indicative of the generalisability of the model and can therefore be used as a competitive selection criteria. We further show that ConvTopRep outperforms the current state-of-the-art model selection methods in domain generalisation.

- Model Behaviour Analysis. Due to the complexity and multi-dimensionality of neural networks, interpreting or visualising their weight space becomes a challenging task. In this paper, we propose to use ConvTopRep to analyse the impact of various training methods on the network weight space. In addition, we show that the evolution of topology of convolutional layers throughout training can shed light on the behaviour of neural networks and can help to make a more incisive model selection.

Our contributions are three-fold: Firstly, we study the topology of the space of convolutional filters and introduce Convolutional Topology Representation - a novel measure - as a meaningful and compact representation of the topology of convolutional layers. Secondly, we show that the proposed measure is indicative of model generalizability and can be used as a competitive model selection criterion in the context of domain generalization. Lastly, we demonstrate that the described topological properties provide valuable insights into the behavior of the model.

2 Related Work

Various efforts have been taken to analyse the behaviour of neural networks through the lens of topology and persistent homology. The first work in this direction employs the graph structure of the fully connected layer by considering neurons as vertices and the their connection as edges (Rieck et al., 2019). The authors construct a Clique complex of the undirected multi-partite graph, where the filtration is induced by the values of the weight matrix. The constructed Clique complex reflects the structural complexity of the layer, with large (absolute) weights indicating that certain neurons exert a larger influence over the final activation of a layer. Finally, a persistence diagram for 0 dimensional topological features is constructed. The resulting measure, called *neural persistance*, is calculated as the Euclidean distance of the points of the persistence diagram to its diagonal. Neural persistance demonstrates the advantages of dropout and batch normalization, and can be used as an early stopping criterion.However, the authors also acknowledge that the described edge-focused filtration scheme is not efficient for convolutioal layers.

Lacombe et al. (2021) extend this line of work by diverting attention from a weight graph to an activation graph. Their proposed topological quantity, called *topological uncertainty*, estimates the difference between the activation of the network by the unseen test data versus the original train data. The proposed measure gives promising results in detecting out-of-distribution samples and selecting trained networks for unlabeled data. However, calculating persistent homology for each sample is a time-consuming process, which makes the method computationally inefficient and infeasible for large scale datasets.

In the proposed approach by (Corneanu et al., 2019, 2020), a Pearson correlation matrix is computed based on the network activations by the train set. Then, the resulting correlation values are used as a distance matrix for Vietoris–Rips or Clique complexes. Various topological approximators, such as betti numbers and average persistence are used for test error estimation and adversarial attack detection. The main drawback of this approach is that the constructed correlation matrix does not encode the structural information of the network, and thus, is unable to capture conditional dependencies of the neurons.

An entirely different approach to a network representation was taken by Birdal et al. (2021), where instead of focusing on the topology of each individual model or layer, the authors analyse the weight trajectories of the network after convergence and introduce a new measure of intrinsic dimension, that correlates with generalization error (note that their experiments are focusing on the generalisation error for the datasets belonging to the same domain). However, their described complex construction is computationally prohobitive for large networks, as it requires building several VR complexes with simplices, each simplex representing the entire set of network parameters.

Unlike Rieck et al. (2019) and Lacombe et al. (2021), that primarily focus on the topology of fully connected layers, Carlsson and Gabrielsson (2020) analyse convolutional layers and discusses the simple global structures that are encoded in the weights of convolutional layers. The authors employ a method of partial clustering of convolutional weights to extract simplicial complexes, called Mapper (Singh et al., 2007), and create a filtration based on condensity. The described construction of the simplicial complex with Mapper requires performing 100 trainings of the same model, which is computationally cumbersome. Based on the findings of Carlsson and Gabrielsson (2020), the proposed approach by Love et al. (2020) further attempts to improve the generalisability of CNNs by restricting the weights of the first convolutional layers to lie on a topological manifold, such as Primary Circle or Klein Bottle. While demonstrating improvement in generalizability of the network to unseen datasets, the authors report only 30% accuracy when generalizing from MNIST to SVHN. (Carlsson and Gabrielsson, 2020) may be closest in spirit to our work. However, our method of extracting and summarizing topological features allows for a per-model comparison and gives rise for a more detailed analysis of the uniformity of the topological space of convolutional filters.

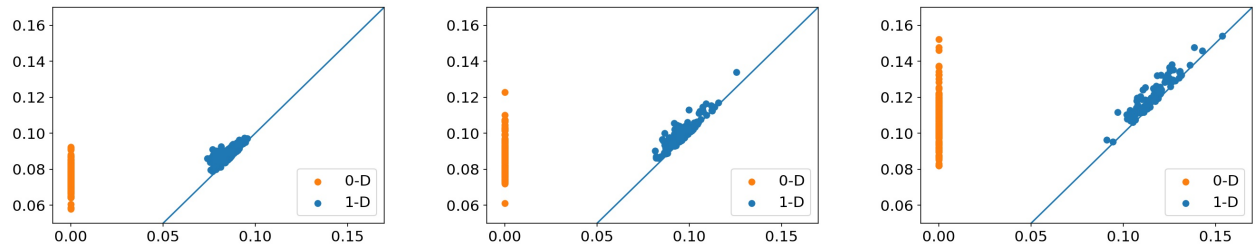


Figure 1: Persistence diagrams with 0D and 1D topological features for epoch 0 (left), 50 (middle), 150 (right).

3 Methodology

In this work, our goal is to define a measure that can give an insight into the behaviour of a convolutional model, and indicate its ability to generalize on unseen datasets, without requiring access to the source and target datasets. To this end, we restrict our analysis on the weight space of convolutional layers - a primary component of CNNs.

Specifically, we start with a simple straightforward measure - a weight norm, that is commonly used as a baseline criteria for selecting filters in network pruning, where the filters with the small norm are considered to be uninformative and can thus be dropped (Li et al., 2017b). In Section 4.3, we experimentally show that having the largest weight norm cannot give any insight on model generalisability in the context of domain generalisation, making it an ineffective criteria for model selection. We further introduce an alternative weight based measure that represents a topological footprint of the space of convolutional weights. Finally, we show that our method outperforms weight-norm criteria and other state of the art model selection methods.

In the next section, we provide an intuition behind using topological descriptor for convolutional filter space, followed by its formal definition and derivation in Section 3.2. We include the basic definitions of homology in the supplementary material.

3.1 Motivation and Interpretation

In this work, we define a measure that summarises the distribution of the weights of convolutional filters. To this end, we employ persistent homology (PH) - a well-known method from topological data analysis, that provides an insight to the global "shape" of the weight space. An advantage of using PH is in its robustness against weight perturbations. Moreover, by design, PH is calculated based on the pairwise distance between filters; and therefore summarizes a relative distribution of filters with respect to each other. By definition, PH is order invariant and therefore, can be used for comparing the networks trained on different data and with different weight initialisations. The output of PH can be given in a form of a persistence diagram, where each point represents a topological feature.

0-D topological features (i.e., connected components), summarise the density of the filter space. As the network trains, the average pairwise distance between the filters increases. In terms of topology, this translates to longer lifetime of connected components (see Fig.1) - the orange points representing 0D features move upwards along y axis of the persistence diagram. However, our experiments showed that the evolution of the connected components' lifetime is not consistent across layers and training methods, which makes the representation unreliable.

1-D topological features (i.e., holes) In Fig. 1, each blue point in the persistence diagram represents a hole, and the further the point is from the diagonal in the persistence diagram, the larger is the size of the hole in the original space. As the network trains, the points in the persistence diagram move upwards along the diagonal. It reflects the increase of the filters variability: the distance between filters becomes larger, resulting in holes appearing at a larger scale. However, no hole persists for a long time indicating that the filters are relatively uniformly distributed. To aggregate the information about all the holes, we calculate total persistence as the sum of distances of the points to the diagonal, which represents the sum of the lifespans of the holes. Our main finding is that the behaviour of the model and its generalisability is correlated with the total lifetime of 1d topological features.

3.2 Convolutional Topology Representation

In this section, we introduce and formally define the Convolutional Topology Representation (ConvTopRep). By exploiting the topology of the feature space of convolutional layers, ConvTopRep provides a compact, yet representative summary of a filter distribution.

The calculation of ConvTopRep consists of three main steps, summarized in Figure 2.

Step 1. Construct a metric space by projecting convolutional filters to the Euclidean Manifold.

A convolutional layer can be seen as a finite set of points, each representing a filter. Characterizing the topology of this point set requires having a metric space. Formally, we define it as follows:

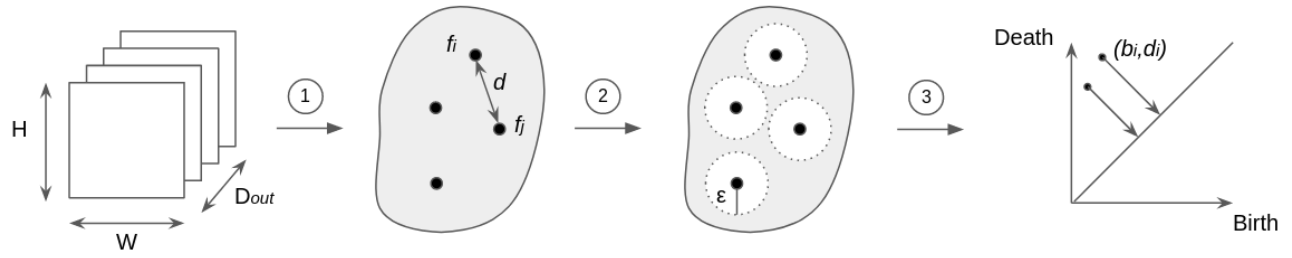


Figure 2: The pipeline of calculating ConvTopoRep for one channel. (1) The set of filters is projected to a Euclidean Manifold, where the distance between filters is defined as Eucledean distance. (2) Vietorios-Rips (VR) complex is constructed on the set of filters. (3) Persistent homology is applied by gradually increasing the threshold ϵ and monitoring the creation and destruction of 1D topological features. The resulting persistence diagram is summarised as 1-Wasserstein distance of the points to the diagonal.

In a convolutional layer, the set of learnable filters transforms the input x^{in} to the output x^{out}. Lets assume that the layer l has a set filters, each of which is of the size H x W: $F \in \mathbb{R}^{H \times W \times D_{in} \times D_{out}}$. We define $F_c = \{f_0, f_1, ..., f_{D_{out}-1}\} \in \mathbb{R}^{HW \times D_{out}}$ as a finite set of flattened filters for a channel c. The distance between filters which belong to one channel can be derived as the Euclidean distance of $\forall f_i, f_j \in F_c, d(f_i, f_j) = \sqrt{\sum_{k=1}^{HW} (f_{ik} - f_{jk})^2}$, where ($F_c$, d) represents a complete metric space.

Step 2. Construct Vietorios-Rips complex.

Definition 1 ***Vietoris–Rips Complex*** *$VR_\epsilon(F_c, d)$ is an abstract simplicial complex, where each filter f_i represents a point. $VR_\epsilon(F_c, d)$ has k-simplex if the distance between every pair of k+1 points is at most ϵ: $d(f_i, f_j) < \epsilon \forall i, j \in k+1, k < D_{out}$.*

Given the metric space, we can construct a filtration as follows: A filtration K of VR complex is populated by considering the distance ϵ as a free parameter. By gradually increasing ϵ, we obtain the family of nested complexes.

Step 3. Calculate Persistent homology and approximate the resulting persistence diagram into a fixed-sized metric.

We further apply persistence homology on a constructed VR complex and obtain a persistence diagram.
Intuitively, zero-dimensional topological features, or connected components, summarise the density of the space. On the other hand, the existence of the first-dimensional topological features indicates that the filters are unevenly distributed in space. In this work, we only look at 1-D topological invariants, or holes.

Persistence diagram contains a multiset of points $D = D_1 \in \mathbb{R}^2$, where each point $x_i = (b_i, d_i)$ indicates that a hole appeared when the distance $\epsilon = b_i$ and disappeared when $\epsilon = d_i$.

Assume that persistence diagram D has j+1 points, $D = \{x_0, ..., x_j\}$. Then the degree-1 total persistence is defined as follows: $Pers(D) = \sum_{i=0}^{j} d_i - b_i$.

It is equivalent to a 1-Wasserstein distance to the diagonal of the persistence diagram.

Finally, in order to summarise the persistence of the filter space for a convolutional layer l, we average the total persistence of VR complexes for all channels. Formally,

$$ConvTopRep_l = \frac{\sum_{i=1}^{c} Pers(D_i)}{D_{in}}$$

Remark 1 *Given the layer l with $F \in \mathbb{R}^{H \times W \times D_{in} \times D_{out}}$ filters, the upper bound of $ConvTopRep_l$ can be derived as*

$$ConvTopRep_l < \frac{D_{out} \times d_{max}}{D_{in}}$$

with d_{max} being a largest possible Euclidean distance between filters: $d_{max} = sup(d(f_i, f_j)) \forall f_i, f_j$.

Proof 1 *The recent study by Lim et.al. Lim et al. (2020) defines an upper bound on a death time d_i for VR complex on a metric space (F, d) to be smaller or equal than the radius of F. Given that $b_i < d_i, \forall i \Rightarrow d_i - b_i \leq radius(F)$.*

If a persistence diagram D has j points, the total persistence is bounded by: $Pers(D) < j * radius(F)$. Since VR complex is build on a pairwise Euclidean distance matrix between the filters in our setup, $radius(F)$ represents the largest possible distance between the filters: $d_{max} = sup(d(f_i, f_j)) \forall f_i, f_j$. Finally, by construction, the number of 1-D topological features can not exceed the number of the elements in a distance matrix: $j < D_{out}$. Therefore, $ConvTopRep_l$ is bounded by:

$$ConvTopRep_l < \frac{j * radius(F)}{D_{in}} \leq \frac{D_{out} \times d_{max}}{D_{in}}$$

Finally, the ultimate measure $ConvTopRep$, that describes a topological footprint of a model, is defined as the average $ConvTopRep_l$ across all its convolutional layers. A model selection criteria is then to choose the model with the smallest $ConvTopRep$.

	Digits			CIFAR-10			
	SVHN	USPS	SYNTH	Weather	Blur	Noise	Digits
ERM	34.89 ± 1.75	79.05 ± 1.14	44.43 ± 1.1	75.56 ± 1.88	77.51 ± 1.67	52.15 ± 3.75	78.35 ± 1.26
ME-ADA	38.73 ± 2.43	79.13 ± 1.01	48.11 ± 1.61	80.3 ± 0.44	83.22 ± 0.36	68.57 ± 2.17	83.13 ± 0.24
RandConv	64.72 ± 1.2	85.75 ± 1.18	68.09 ± 1.67	71.1 ± 1.34	72.74 ± 1.53	58.46 ± 2.81	75.07 ± 1.63

Table 1: Average test accuracies of LeNet (left) and WideResNet (right), trained with ERM, ME-ADA and RandConv.

4 Experiments

We designed experiments to investigate the scenarios in which ConvTopRep can be used as a competitive measure for model selection and the effect that different training methods have on the filter space of the network. A diverse range of experiments has been conducted to show a practical use of the proposed measure for both shallow and deep convolutional networks. We start with a description of the experimental setup and then proceed to comparing the models using the topology of their convolutional layers. We finally conclude with the analysis of the evolution of topology throughout training.

4.1 Experimental setup

Datasets We consider three collection of datasets:
Digits, CIFAR-10 and ImageNet.
Digits consists of four datasets (or domains) which represent a collection of handwritten digits with different styles: MNIST (LeCun et al., 2010), USPS (Denker et al., 1989), SVHN (Netzer et al., 2011), and SYNTH (Ganin and Lempitsky, 2015). Following single domain generalization setup by Zhao et al. (2020), we use 10K samples from MNIST for training, and evaluate the performance on all other datasets.
CIFAR-10 (Krizhevsky and Hinton, 2009) contains natural images from 10 different classes, and consists of 50K training samples and 10K test samples. We use CIFAR-10 for training, and CIFAR10-corrupted (Hendrycks and Dietterich, 2019) for evaluation. CIFAR10-corrupted is constructed by exposing CIFAR-10 to 15 types of corruption.
ImageNet classification set (Russakovsky et al., 2015) is a large collection of more than 1M images, distributed across 1000 classes. ImageNet-Sketch (Wang et al., 2019) consists of 50K sketch-like images, that match the ImageNet in categories and scale.

Networks We consider three neural network structures: LeNet (Lecun et al., 1989) as a relatively shallow network for the Digits dataset, WideResNet (16-4) (Zagoruyko and Komodakis, 2016) as a more complex network for CIFAR10, and a deep network AlexNet (Krizhevsky et al., 2012) for ImageNet. We refer to the supplementary material for a summary of the training hyperparameters.

Algorithms We selected three state-of-the-art methods from domain generalization: ERM (Vapnik, 1998), ME-ADA (Zhao et al., 2020), and Random Convolutions (RandConv)(Xu et al., 2021). ME-ADA generates hard adversarial samples, that maximizes the entropy of the latent representation of the network. The entropy is calculated on the softmax output of the model and is designed to enlarge its predictive uncertainty. RandConv is a data augmentation technique, where the dataset is extended by applying convolutional filters with random parameters, incorporating various textures while preserving global shapes.

The algorithms of choice are the ideal candidates for our topological analysis, as none of them is extending or modifying the structure of the network. Therefore, they share the same number of learnable parameters, including convolutional filters, and only differ by the value of their weights. Additionally, we share an insight that some generative methods display unstable behaviour, that can be detected with ConvTopRep. We refer to the supplementary material for more discussion.

4.2 ConvTopRep gives an insight on how the choice of objective affects the weight space of the network

MNIST, LeNet: We start by analysing the filter space of a small model, LeNet, trained on the MNIST dataset accross the different domain generalisation methods selected (ERM, ME-ADA, RandConv). For each training, the model from the last epoch is selected. The values of ConvTopRep are calculated for each convolutional layer, shown in Fig.3, left.

We first observe that the variance of ConvTopRep for the first layer of LeNet is significantly larger than for the second layer regardless of the domain generalisation method examined (e.g. 0.13 against 0.012 for ERM). This indicates that, regardless of the method, the weight space of the first convolutional layer of a relatively small network like LeNet is sensitive towards parameter initialisation.

Next, we measure the correlation between the test accuracies and the values of ConvTopRep. The test accuracies for each method are shown in Table.1. The resulting plots from Fig.3 reveal that on MNIST dataset, the training using RandConv leads to the most uniformly distributed filter space of both first and second convolutional layers, reflected in the smallest values of ConvTopRep, and the highest accuracies on all three test datasets. Similarly, the highest values of ConvTopRep, corresponding to the ERM method, match the lowest test accuracies.

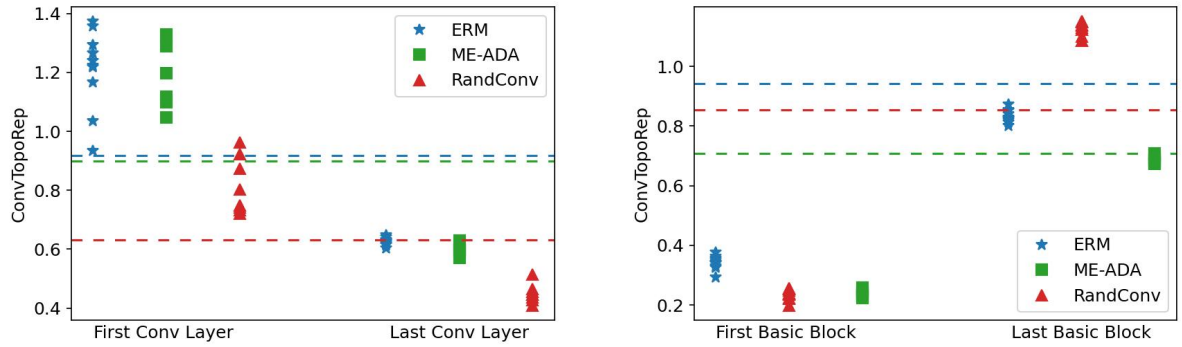


Figure 3: Distribution of ConvTopRep for LeNet convolutional layers (left) and WideResNet blocks (right). Dashed lines represent the average ConvTopRep across all convolutional layers.

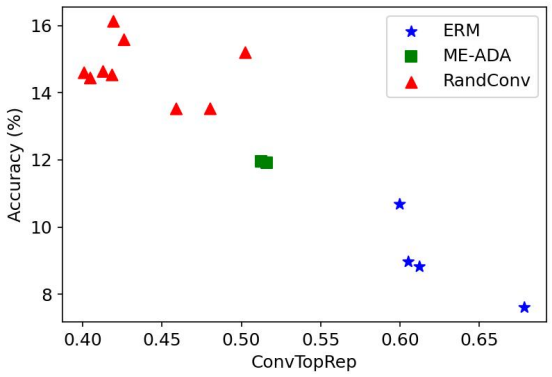


Figure 4: Correlation between the accuracy of ImageNet-Sketch and the ConvTopRep for AlexNet. Lower values of ConvTopRep correspond to the higher test accuracies.

CIFAR10, WideResNet We proceed with the model selection analysis of a more complex model, WideResNet, trained on CIFAR-10. ConvTopRep for the first and the last basic blocks of WideResNet are showed in Fig.3, right, and the associated test accuracies on CIFAR10-corrupted are summarised in Table.1. Similarly to LeNet, the ConvTopRep of the last layer of WideResNet is correlated with average accuracies of all methods. The best performing model, ME-ADA, has the smallest value of ConvTopRep, and RandConv, representing the worst average accuracy among all methods, has the largest ConvTopRep.

Additionally, the topology of convolutional layers gives an insight into the nature of the methods. For example, the ConvTopRep values of RandConv of the first basic block are smaller than ERM and even smaller than ME-ADA for some trainings. It shows that RandConv enhances the topology of the first basic block. Indeed, the strength of RandConv is in making the model robust against style perturbations, which primarily affects the earlier layers in the network. It is also reflected in the test performance: while being on average less accurate than other methods, RandConv outperforms ERM on the "noise" corruption.

ImageNet, AlexNet We conclude our set of model selection experiments with AlexNet trained on ImageNet. We train one ERM model, one ME-ADA model, and three different configurations of RandConv models. In order to increase the number of models to select from, we additionally add early checkpoints for each training, namely, after epoch 30, epoch 60, and the last epoch 90. The resulting collection contains 15 different models. For more details about the training hyperparameters, please refer to the supplementary material.

According to the values of ConvTopRep, visualised in Figure 4, RandConv outperforms both ME-ADA and ERM, which is confirmed by the values of the test accuracy of ImageNet-Sketch. However, it can be seen that the value of ConvTopRep, corresponding to the average ConvTopRep_l across convolutional layers are not very accurate when selecting models within one training method (e.g.,, RandConv models). As discussed in the previous sections, RandConv mostly affects the first layers of the network by making them more robust against style perturbations.

Correlation Analysis We evaluated the relationship between the model accuracy and the average ConvTopRep value by computing Pearson correlation coefficients, which can range from -1 to 1, with values indicating the direction and strength of the relationship. A coefficient of 0 indicates no correlation, while 1 and -1 indicate perfect positive and negative correlation, respectively.

For each experimental setup, we calculated the average accuracy across target datasets and computed its correlation with ConvTopRep. Our findings show strong negative correlations between the accuracy of the models studied and ConvTopRep value, with Pearson Correlation coefficients of -0.8767 for Digits, -0.841 for CIFAR10, and -0.934 for ImageNet.

	Digits			
	SVHN	USPS	SYNTH	Avg
Source Risk	35.69 ± 1.45	79.80 ± 1.00	45.45 ± 0.65	53.65 ± 1.03
Entropy	46.46 ± 11.6	76.17 ± 7.95	54.77 ± 9.20	59.13 ± 9.59
Source SND	38.47 ± 1.02	70.72 ± 2.36	48.86 ± 1.55	52.68 ± 1.64
Target SND	33.07 ± 0.96	79.94 ± 8.51	67.47 ± 1.40	60.16 ± 3.62
Weight Norm	63.29 ± 2.41	86.48 ± 0.55	67.47 ± 1.39	72.41 ± 1.45
NP	63.21 ± 0.99	87.00 ± 0.99	68.32 ± 0.99	72.84 ± 1.73
ConvTopRep	65.47 ± 0.11	86.38 ± 1.72	69.50 ± 0.63	73.78 ± 0.82

Table 2: Model Selection Comparison for LeNet, trained on MNIST dataset. The values represent the average accuracy and the std of top-3 best models, selected from 30 candidates. Proposed ConvTopRep outperforms other baselines.

	CIFAR-10				
	Weather	Blur	Noise	Digits	Avg
Source Risk	67.73 ± 0.94	78.90 ± 0.70	53.29 ± 2.16	72.32 ± 0.83	68.06 ± 1.16
Entropy	67.63 ± 1.26	78.86 ± 0.78	50.12 ± 1.29	71.97 ± 1.06	67.15 ± 1.09
Source SND	67.46 ± 1.23	78.81 ± 0.35	53.07 ± 2.32	72.02 ± 0.69	67.84 ± 1.15
Target SND	56.79 ± 3.11	73.01 ± 1.49	26.87 ± 0.93	60.71 ± 1.74	54.34 ± 1.82
Weight Norm	63.15 ± 1.78	71.92 ± 1.68	30.06 ± 4.98	64.88 ± 1.85	57.50 ± 2.57
NP	59.07 ± 2.97	68.52 ± 1.73	42.89 ± 1.30	62.25 ± 3.02	58.43 ± 2.26
ConvTopRep	68.55 ± 0.91	78.91 ± 0.72	53.31 ± 2.34	72.35 ± 0.67	68.28 ± 1.16

Table 3: Model Selection Comparison for WideResNet, trained on CIFAR-10. The values represent the average accuracy and the std of top-3 best-selected models, selected from 30 candidates. Proposed ConvTopRep outperforms other baselines.

4.3 The network with the smallest ConvTopRep value generalises better on unseen target domains

One of the challenging tasks in domain generalization is to estimate the performance of the model without having access to the test data. In this section, we demonstrate how ConvTopRep can be used to assess the generalisability of the model and help to select an optimal method for model selection. In particular, we propose the following model selection procedure:
We first calculate $ConvTopRep_l$ for all the convolutional layers l of the network. We further average the values of $ConvTopRep_l$ accross the layers to obtain $ConvTopRep$ - a measure, that summarises the average total persistence of the filter space of the network. The model selection criteria is thus to select the network with the smallest value of $ConvTopRep$.

Model Selection Baselines We compare the performance of our method with a variety of domain adaptive model selection approaches, including Weight Norm (Li et al., 2017b), Source Risk and Source Class-Entropy (Ganin and Lempitsky, 2015) and a current state-of-the-art unsupervised criterion called Soft Neighborhood Density (SND) (Saito et al., 2021). Source Risk, commonly used as an early stopping criterion in general ML applications, and Source Entropy, that can be viewed as a confidence measure for the model predictions (Wang et al., 2021a), are sensitive to the source data distribution and are thus ineffective when a large domain gap is present. In contrast, SND employs a target feature space by hypothesising that a good source classifier would embed the target samples into dense neighborhoods. Unlike SND, our method does not require the access to the data in target domains and therefore can also be used in domain generalisation setup. For the completeness of the experimental setup, we calculate SND on both source validation data and target data.
Finally, we include in the analysis a topology-based model selection method, namely Neural Persistence (NP) (Rieck et al., 2019), that defines a measure based on the topology of the fully connected layers. We show that, while efficient in fully connected networks, NP is less representative of the behaviour of large convolutional networks.

Experiments We build a pool of 30 trained models (10 for each algorithm - ERM, ME-ADA and RandConv). We further choose the top 3 best models according to various model selection strategies and report their average target accuracies. The results for LeNet, trained on MNIST, are summarised in Table 2; for WideResNet, trained on CIFAR-10 - in Table 3, and for AlexNet, trained on ImageNet - in Table 4. The results reveal the superiority of ConvTopRep over the other methods across all the datasets and network structures. Note that the other topology-based method NP performs well on MNIST dataset, but fails on WideResNet, as it has only one fully connected layer, which proves to be insufficient to represent the generalisability of the whole deep network. Similarly, SourceSND works well on WideResNet and AlexNet, but results in an

ImageNet						
SourceRisk	Entropy	SourceSND	TargetSND	NP	WeightNorm	ConvTopRep
11.52 ± 0.6	11.52 ± 0.6	14.39 ±0.68	8.46 ±0.61	14.23 ±0.5	9.03 ±1.26	14.56 ±0.08

Table 4: Model Selection Comparison for AlexNet, trained on ImageNet. The values represent the average accuracy and the standard deviation of top-3 best models, selected from 15 candidates. Proposed ConvTopRep outperforms other baselines.

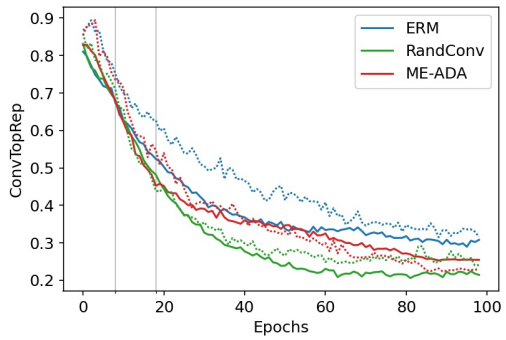
inefficient model selection for LeNet. Unlike other methods, that succeed in some setups, while failing in others, our approach provides competitive results across all the investigated datasets and networks.

4.4 The evolution of ConvTopRep value sheds light on the model's behaviour throughout training

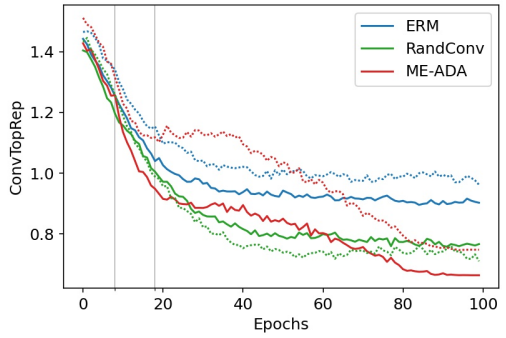
In order to get an insight into the importance of each convolutional layer to make an informed and incisive model selection, it is important to understand the correlation between the topology of different layers and its evolution throughout training. In this section, we investigate how the topology of convolutional layers evolve during the training and how it is affected by different methods. As an example, we take 3 trainings of WideResNet and analyse the evolution of topology separately for each method (ERM, RandConv and ME-ADA). In summary, we show that the method that improves the ConvTopRep of the earlier blocks may incite a degradation of the topology of the later blocks, and vice versa (see Fig.5).

ERM: We start our analysis with ERM. As the model trains, ConvTopRep of most blocks continuously decreases, with the exception of the first convolution layer of the last Network block (see dashed blue line in Fig.5c, where ConvTopRep only decreases for the first 26 epochs). The inconsistency in the evolution of ConvTopRep reflects the instability of the test accuracies, with one corruption type (noise) achieving the best accuracies in the early epochs, while others continue to increase until later epochs.

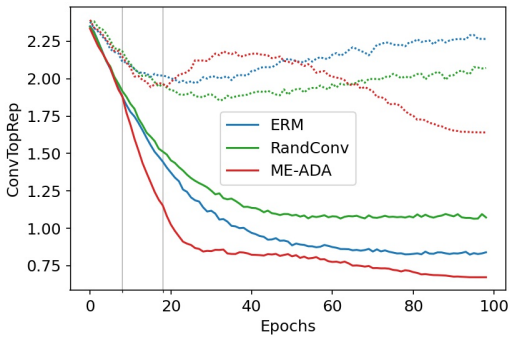
ME-ADA: The analysis of ME-ADA reveals more fascinating patterns. As expected, during the first 8 epochs, ME-ADA exhibits the same behaviour as ERM on all blocks. After epoch 8, the first round of adversarial data generation is performed, which produces significant drops of ConvTopRep, particularly for the last convolution layer (see the solid red line in Fig.5c). It is consistent with our knowledge of ME-ADA, as it produces adversarial samples based on the output of the last fully connected layer, which is best reflected in the last layers of the network. The second round of adversarial data generation happens after epoch 18. While the ConvTopRep of the last basic block keeps decreasing, the earlier basic blocks become negatively affected by the insertion of adversarial samples, with the values of ConvTopRep decreasing considerably slower than earlier epochs and even increasing for the first layer of the last block (dashed red line in Fig.5b). The ConvTopRep measure stabilizes towards the end of the training, with the lowest values for all the blocks obtained in the last couple of epochs. It is correlated with the model performance on the test dataset, that achieves the highest accuracies on all corruption types in the last epochs of the training.



(a) Network Block 1



(b) Network Block 2



(c) Network Block 3

Figure 5: Evolution of ConvTopRep during training for different WideResNet blocks. Dashed lines represent the 1-st basic block, solid lines represent the 2-d basic block. The vertical grey lines indicate the min-max step of ME-ADA.

RandConv: Finally, we conclude our study with the analysis of the topology of WideResNet, trained with RandConv. In contrast to ME-ADA, RandConv aims to finetune the first layers by introducing texture and color variance to the training samples. It is reflected in a continuous decrease of ConvTopRep for the first two Network blocks. However, once the ConvTopRep of the first basic block of RandConv starts outperforming ERM in epoch 15 (see the blue and red solid lines in Fig.5a), the ConvTopRep of the last block starts converging to a higher value than ERM (see the blue and red solid lines in Fig.5c).

5 Conclusion

In this work, we studied the space of convolutional filters through the lens of persistent homology. We introduced a measure, called ConvTopRep, that summarises 1-D topological features of a convolutional layer, and showed how it can be used to understand the effect of various objectives on the weight space, to perform an informed model selection and to analyse the model's behaviour.

Acknowledgements
This research is supported by the National Key Research and Development Program of China No. 2020AAA0109400 and the Shenyang Science and Technology Plan Fund (No. 21-102-0-09).

References

Atiya, A. and Ji, C. (1997). How initial conditions affect generalization performance in large networks. *IEEE Transactions on Neural Networks*.

Birdal, T., Lou, A., Guibas, L. J., and cSimcsekli, U. (2021). Intrinsic dimension, persistent homology and generalization in neural networks. In *NeurIPS*.

Carlsson, G. and Gabrielsson, R. B. (2020). Topological approaches to deep learning. In Baas, N. A., Carlsson, G. E., Quick, G., Szymik, M., and Thaule, M., editors, *Topological Data Analysis*.

Corneanu, C. A., Madadi, M., Escalera, S., and Martinez, A. M. (2019). What does it mean to learn in deep networks? and, how does one detect adversarial attacks? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Corneanu, C. A., Madadi, M., Escalera, S., and Martinez, A. M. (2020). Computing the testing error without a testing set. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Denker, J., Gardner, W., Graf, H., Henderson, D., Howard, R., Hubbard, W., Jackel, L. D., Baird, H., and Guyon, I. (1989). Neural network recognizer for handwritten zip code digits. In Touretzky, D., editor, *Advances in Neural Information Processing Systems*, volume 1.

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*.

Gulrajani, I. and Lopez-Paz, D. (2021). In search of lost domain generalization. In *International Conference on Learning Representations*.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*.

Kendall, M. and Gibbons, J. D. (1955). *Rank Correlation Methods*. Hafner Publishing Co, 5 edition.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, page 1097–1105. Curran Associates Inc.

Lacombe, T., Ike, Y., and Umeda, Y. (2021). Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs. In *IJCAI*.

Lecun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Li, D., Yang, Y., Song, Y., and Hospedales, T. M. (2017a). Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. (2017b). Pruning filters for efficient convnets. *Proceedings of the International Conference on Learning Representations*.

Lim, S., Mémoli, F., and Okutan, O. B. (2020). Vietoris-rips persistent homology, injective metric spaces, and the filling radius. *ArXiv*, abs/2001.07588.

Love, E., Filippenko, B., Maroulas, V., and Carlsson, G. E. (2020). Topological convolutional neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*.

Maria, C., Boissonnat, J.-D., Glisse, M., and Yvinec, M. (2014). The gudhi library: Simplicial complexes and persistent homology. In Hong, Hoonand Yap, C., editor, *Mathematical Software – ICMS 2014*, pages 167–174. Springer Berlin Heidelberg.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

Pérez, J. B., Hauke, S., Lupo, U., Caorsi, M., and Dassatti, A. (2021). giotto-ph: A python library for high-performance computation of persistent homology of vietoris–rips filtrations.

Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., and Borgwardt, K. (2019). Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Saito, K., Kim, D., Teterwak, P., Sclaroff, S., Darrell, T., and Saenko, K. (2021). Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9164–9173. IEEE Computer Society.

Singh, G., Memoli, F., and Carlsson, G. (2007). Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In *Eurographics Symposium on Point-Based Graphics*.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2021a). Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. (2019). Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518.

Wang, J., Lan, C., Liu, C., Ouyang, Y., and Qin, T. (2021b). Generalizing to unseen domains: A survey on domain generalization. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Xu, Z., Liu, D., Yang, J., Raffel, C., and Niethammer, M. (2021). Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. In *BMVC*.

Zhao, L., Liu, T., Peng, X., and Metaxas, D. (2020). Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zomorodian, A. (2010). Fast construction of the vietoris-rips complex. *Comput. Graph.*

Supplementary Material of: Convolutional Persistence as a Remedy to Neural Model Selection

This Supplementary Material is organised as follows: We introduce some basic definitions from Persistent Homology in Section A, followed by a detailed description of our experimental setup in Section B. We further perform additional experiments on model selection in Section C, where we show that the value of ConvTopRep of individual layers can also be used for selecting the best model. Additionally, we perform similar experiments on another dataset PACS, described in Sec.C.3. Finally, we proceed with the behaviour analysis in Section D. Specifically, we show that ConvTopRep reflects model instability during training and can even be used as an early stopping criteria in some setups.

A Definitions

Simplicial Complex: A Simplicial complex is a combination of simplices, where the intersection between two simplices is also a simplex.
In other words, given a set S and a set C, C is a simplicial complex of S if $a \in C, b \subset a \implies b \in C$

Vietoris-Rips(VR) Complex: Given a finite metric space (X, d), VR complex of X at a threshold $r > 0$ is a Simplicial Complex containing all the set of simplices with the diameter smaller than r:
$VR_X(r) := \{\nu \subset X : diam(\nu) < r\}$, where $diam(\nu) = max\{d(x_i, x_j) \,|x_i, x_j \in \nu\}$

Filtration: A sequence of simplicial complexes, such that $\emptyset = C_0 \subset ... \subset C_n = C$ is called a *filtration* of C. For VR complex, a filtration is created by gradually increasing the diameter r.

Persistent Homology: Persistent homology is a method that keeps track of topological structures that persist throughout a filtration. The resulting homology of the filtration is summarised in a persistence diagram $\mathcal{D}$, where each point (b_i^k, d_i^k) indicates that a k-th dimensional topological feature appeared when the threshold $\alpha = 1 - b_i^k$ and disappeared when $\alpha = 1 - d_i^k$. The points from persistence diagrams represent a multi-set.

B Implementation details

In this section, we provide the details of our experimental setup.

LeNet consists of two convolutional layers, followed by three fully connected layers. Both convolutional layers have kernel size = [5 x 5]. ReLU activation function is applied on every layer apart from the last layer, and 2D maxpooling is applied after the convolutional layers. Networks are trained using Adam optimizer with learning rate equal to 0.0001 and batch size equal to 32. ERM models are trained for 30 epochs, ME-ADA models are trained for 60 epochs, RandConv models - for 200 epochs.

WideResNet (16-4) consists of a convolutional layer and 3 Network blocks, followed by a batch normalization layer and a fully connected layer. Each Network block has two basic blocks with two convolutional layers each. The kernel size for all convolutional layers is of size [3 x 3]. WideResNet is optimized by SGD using Nesterov momentum with a learning rate of 0.1, weight decay of 0.0001, and the learning rate decay following a cosine annealing schedule. All the networks are trained for 100 epochs with batch size equal to 128.

AlexNet is trained for 90 epochs using SGD optimizer with momentum of 0.9, weight decay of 0.0001 and batch size equal to 256. We initialize the learning rate to be 0.01 and decrease it according to the Step LR scheduler with gamma of 0.1 and step size of 30 epochs. For each training, we normalise the data and perform random resized crop and random horizontal flip.

	Digits			
	SVHN	USPS	SYNTH	Avg
Source Risk	0.053	0.180	-0.196	0.012
Entropy	0.365	0.223	0.529	0.372
Source SND	0.376	0.228	0.476	0.360
Target SND	0.148	-0.005	0.333	0.159
Weight Norm	0.317	0.122	**0.556**	0.332
NP	0.376	0.117	0.540	0.339
ConvTopRep	**0.513**	**0.377**	0.444	**0.445**

Table 1: Kendall's τ coefficient values for LeNet, trained on MNIST

Due to the computational complexity of the training, we only train the networks under the following setups:

- Trained from scratch with ERM (only basic augmentations)
- Trained from scratch with RandConv:
  $RC_{img1-7}, p = 0.5, \lambda = 10$
- Pretrained on ImageNet
- Pretrained on ImageNet, finetuned with RandConv:
  $RC_{img1-7}, p = 0.5, \lambda = 10$
- Pretrained on ImageNet, finetuned with RandConv:
  $RC_{img1-7}, p = 0.8, \lambda = 10$
- Pretrained on ImageNet,
  finetuned with MeAda:
  $lr\_max = 50, loops\_adv = 50, \beta = 1, \gamma = 10$

Note that for ImageNet, we only train one model with MeAda due to both memory and time inefficiency of this method (it requires to store a copy of modified version of the whole dataset, alongside with the original samples). Finally, we additionally add early checkpoints for each training, namely, after epoch 30, epoch 60, and the last epoch 90. Overall, we obtain 15 different models in our selection pool.

We use giotto-ph library (Pérez et al., 2021) for cpu-paralleled calculation of Vietoris–Rips persistence and gudhi library (Maria et al., 2014) for the other topology-related calculations. In practice, the computation of ConvTopRep for all the convolutional layers of the network takes around 0.4 sec for LeNet, 19.2 sec for WideResNet and 22.7 sec for AlexNet (used 10 Intel(R) Core(TM) i7-9850H CPU @ 2.60GHz).

C Model Selection

C.1 Alternative evaluation via Kendall Rank Correlation

In the main paper, we evaluate the quality of model selection methods by the performance of the top-3 chosen models. For a further quantitative evaluation, we propose to extrapolate model selection methods to perform a ranking task. We further calculate Kendall rank correlation coefficient, or τ coefficient (Kendall and Gibbons, 1955), between the ground-truth ranking, which is the ranking of the models based on their target accuracies, and the ranking, produced by various model selection methods.

Kendall's τ coefficient evaluates the degree of similarity between two sets of ranks, with the values ranging from -1 to 1, with 1 being positive correlation, and -1 being negative correlation. In our case, positive correlation indicates that the models are ranked in a decreasing order of their target accuracies, negative correlation represents the ranking in an increasing order of the accuracies, while $\tau = 0$ represents random ranking. Therefore, large values of τ correspond to a better quality of a ranking method.

The results are summarised in Table 1 for LeNet, trained on digits; in Table 2 for WideResNet, trained on CIFAR10, and in Table 3 for AlexNet, trained on ImageNet. The results for both LeNet and AlexNet show the superiority of ConvTopRep over the other ranking methods, with $\tau = 0.638$ for ConvTopRep-based AlexNet ranker outperforming its closest counterpart WeightNorm by a large margin.

	CIFAR-10				
	Weather	Blur	Noise	Digits	Avg
Entropy	0.166	0.355	0.471	0.208	0.290
Source SND	**0.395**	**0.671**	0.265	**0.399**	**0.443**
Target SND	-0.121	0.426	-0.834	0.029	-0.067
Weight Norm	-0.158	-0.290	-0.659	-0.253	-0.325
NP	-0.325	-0.585	-0.177	-0.384	-0.384
ConvTopRep	0.138	0.219	**0.633**	0.226	0.288

Table 2: Kendall's τ coefficient for WideResNet, trained on CIFAR10

ImageNet						
SourceRisk	Entropy	SourceSND	TargetSND	WeightNorm	NP	ConvTopRep
-0.257	-0.238	0.238	-0.619	0.276	0.029	**0.638**

Table 3: Kendall's τ coefficient for AlexNet, trained on ImageNet

	SVHN	USPS	SYNTH	Avg
ConvTopRep, 1-st Layer	$\underline{65.75 \pm 0.49}$	85.8 ± 1.03	69.13 ± 0.4	73.56 ± 0.28
ConvTopRep, 2-d Layer	64.57 ± 1.22	$\underline{86.41 \pm 0.34}$	68.24 ± 0.99	73.08 ± 0.37
ConvTopRep average	65.47 ± 0.11	86.38 ± 1.72	$\underline{69.49 \pm 0.63}$	$\underline{73.78 \pm 0.67}$
Oracle	65.86 ± 0.41	87.44 ± 0.81	69.64 ± 0.45	74.32 ± 0.56

Table 4: Model Selection Comparison for LeNet, Trained on MNIST

Unlike LeNet and AlexNet, ConvTopRep-based WideResNet ranker is outperformed by activation-based rankers, such as Source SND and Entropy. According to a Kendall's τ coefficient, only the *noise* corruption is ranked better based on ConvTopRep. In particular, ConvTopRep ranker successfully identifies that Me-Ada outperforms both RandConv and ERM, however it has difficulty distinguishing the models within RandConv and ERM categories.

C.2 Model Selection: Ablation study

In this section, we provide additional analysis of the model selection experiments. We start by examining if each individual layer or block can be used separately as a measure for model generalisability. To this end, we calculate ConvTopRep separately for each convolutional layer/block, and use its smallest value to select top-3 best models. The resulting averaged accuracies are summarised in Table 4 for LeNet, trained on MNIST, Table 5 for WideResNet, trained on CIFAR10 and Table 7 for AlexNet, trained on ImageNet. For comparison, we provide the average performance of the original ConvTopRep by choosing the model with the smallest ConvTopRep, averaged across all convolutional layers. Finally, we compare ConvTopRep with the so-called Oracle by selecting the best models.

The results confirm that the ConvTopRep is a competitive model selection method: ConvTopRep average has only 0.5% drop in comparison with Oracle for LeNet and only 1.3% drop for WideResNet. In addition, for a small network, both first and second layers can be used separately for model selection without a significant performance loss (less than 0.7% on average across target domains). This finding is particularly useful for scenarios with limited computational capacities. Similarly, the ConvTopRep of individual blocks for a larger ResNet model can be used for model selection: aside from the first, less informative block, the other blocks contain topological information, representative of the model generalisability. In fact, selecting ConvTopRep computed from one block instead of the average ConvTopRep can lead to better model selection for some domains (e.g., see Weather domain in Table 3). However, selecting an appropriate layer or block requires some prior knowledge about target domain, that is often unavailable in general domain generalisation setup.

In all the previous experiments, we were choosing the last epoch to select the representative model of the training. In order to be consistent with the experimental setup of (Xu et al., 2021) and (Zhao et al., 2020), we perform an additional experiment, where we select the best model to represent a training. The results, summarised in the Table 6, show that both Oracle and ConvTopRep best are noticeably larger than Oracle and ConvTopRep last, which might be an indication of the models' instability. However, our method remains effective, with less than 1.5% drop from Oracle in both "best" and "last" representation strategies.

	Weather	Blur	Noise	Digits	Avg
ConvTopRep, 1-st ResNet Block	58.01 ± 3.37	67.49 ± 1.31	40.21 ± 1.91	62.38 ± 3.67	57.03± 2.56
ConvTopRep, 2-d ResNet Block	67.60 ± 1.03	78.75 ± 0.53	55.29 ± 5.05	71.95 ± 0.91	68.40 ± 1.88
ConvTopRep, 3-d ResNet Block	68.51 ± 1.08	78.95 ± 0.85	51.05 ± 2.01	72.16 ± 0.87	67.68 ± 1.20
ConvTopRep average	68.55 ± 0.91	78.91 ± 0.72	53.31 ± 2.34	72.35 ± 0.67	68.28 ± 1.16
Oracle	68.82 ± 0.12	79.21 ± 0.15	57.76 ± 1.82	72.73 ± 0.15	69.63 ± 0.56

Table 5: Model Selection Comparison for WideResNet, trained on CIFAR-10

	ConvTopRep last	Oracle Last	ConvTopRep best	Oracle best
Digits	73.78	75.32	74.48	75.7
CIFAR	68.28	69.63	79.67	80.53

Table 6: Model Selection Comparison for different train representation strategies

ConvTopRep, 1-st layer	15.63 ± 0.38
ConvTopRep, 2-st layer	14.19 ± 0.47
ConvTopRep, 3-d layer	14.59 ± 0.04
ConvTopRep, 4-d layer	14.19 ± 0.47
ConvTopRep average	14.56 ± 0.08
Oracle	15.63 ± 0.38

Table 7: Model Selection Comparison for AlexNet, trained on ImageNet dataset.

Lastly, a similar set of experiments was conducted for AlexNet, trained on ImageNet dataset. The results, summarized in Table 7, show that even by using the topology of the middle layers as a selection criterion, the average accuracy of the selected models does not fall lower than 14%. In addition, due to the prevalence of RandConv over the other training methods, choosing the models based on the first layer of AlexNet leads to selecting the actual top 3 models within the selection pool (it matches the oracle accuracy, see Table. 7). Finally, in case where no prior knowledge of the training is available, choosing the average across convolutional layers value of ConvTopRep remains a reliable indicator of a model performance and leads to a competitive model selection.

C.3 Model Selection on PACS

We perform model selection experiments on PACS dataset (Li et al., 2017a), trained on AlexNet. PACS dataset containing 224×224 images from 4 domains -Art Paining, Cartoon, Photo and Sketch - distributed across 7 categories. We follow (Li et al., 2017a) for the recommended train, validation, and test splits. AlexNet is optimized by SGD using Nesterov momentum with a learning rate of 0.001, weight decay of 0.0001, and the learning rate decay following a cosine annealing schedule. All the networks are fine-tuned for 100 epochs with batch size equal to 32. We use mixing variant of RandConv. For ME-ADA experiments, the parameters are: $\beta = 10, K = 1, \eta = 20$.

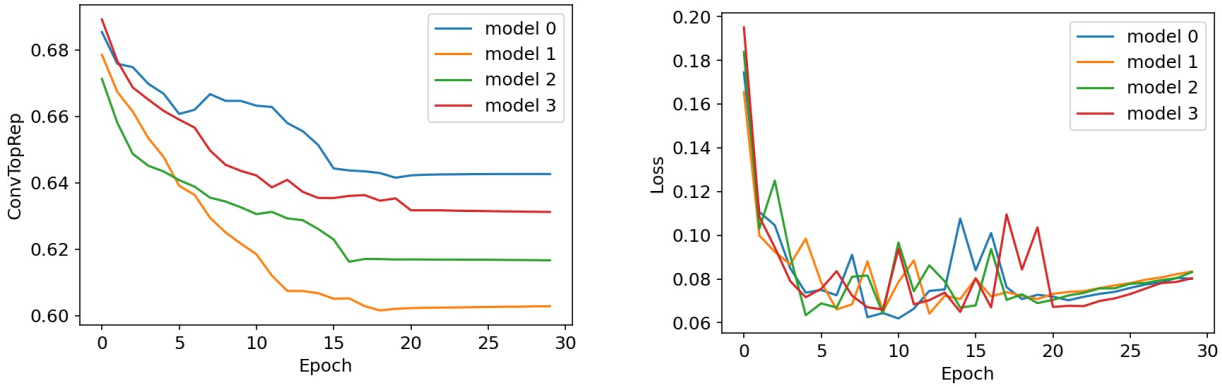
Differently from LeNet and WiseResNet models in our previous experiments, AlexNet is not trained from scratch, but fine-tuned from a pre-trained on ImageNet model. The fine-tuning of ERM, ME-ADA, and RandConv on PACS dataset shows unstable behaviour, which is reflected in the model selection performance of various baselines, shown in Table 8. We observe that a certain model selection method achieves splendid results on some domains, while drastically failing on the others. For example, Source Risk for Art domain is only 0.1% worse than Oracle, while having more than 10% drop on Sketch domain. We observe a similar pattern in our ConvTopRep measure: the model chosen based on the topology of the last layers performs well on Cartoon, Art, and Photo domains, while the model chosen based on the smallest ConvTopRep of the first layers outperforms other approaches on Sketch domain.

D Behaviour Analysis: Model Stability

Another problem we would like to address is the stability of the methods. Given that the test and train data come from different distributions, standard evaluation metrics such as validation loss and validation accuracy might not be representa-

	Cartoon	Art	Photo	Sketch	Avg
Source Risk	65.87	**64.01**	**86.95**	60.27	69.27
Entropy	65.97	61.04	79.58	59.74	66.58
Source SND	65.42	63.18	82.63	59.42	67.66
NP	65.12	61.85	79.6	**71.38**	**69.49**
ConvTopRep$_{best}$	**66.06**	**64.01**	**86.95**	71.24	69.36
Oracle	67.38	64.02	87.46	71.56	72.61

Table 8: Model Selection Comparison for PACS dataset



(a) Evolution of ConvTopRep.

(b) Evolution of the validation loss.

Figure 1: ConvTopRep and validation loss for LeNet trained with ERM, plotted for 4 independent trainings.

tive of the model behaviour on the test set. In this section, we discuss the insight that the topology of convolutional layers provides on the model behaviour, and in particular, on model stability. We chose our first experimental setup, namely LeNet model trained on MNIST dataset, for the described analysis.

ERM: We start our experiments by analyzing the topology of the filter space of the last convolutional layer.

In the beginning of the training, the filters are randomly initialized, which leads to the sparsity of the feature space. As the model trains, the filter space becomes uniformly distributed, which is reflected in the continuous decrease of ConvTopRep (see Fig.1a).

Moreover, we observe that the evolution of the convolutional feature space might be used as an insight into the behaviour of the model during training. More precisely, the lack of change in ConvTopRep indicates that the model has converged to some local optima. As a consequence, it is correlated with the validation loss (see Fig.1b). Therefore, it can be used as an alternative criteria for the early stopping in scenarios where the validation data is not available (e.g. small-scale datasets).

To further validate this, 10 independent trainings were conducted. For each training, we perform an early stopping based on the smallest validation loss and an early stopping based on the smallest ConvTopRep. We compare the performance of these methods on SVHN, USPS, and SYNTH datasets and report a mean and standard deviation of the accuracies over 10 trainings. The result, summarised in Table 9, demonstrate that using ConvTopRep as an early stopping criteria improves the performance on all three datasets.

ME-ADA: In the next set of experiments we analyze how ME-ADA affects the topology of the parameter space in a shallow network (e.g., LeNet).

During the first 300 iterations, the algorithm generates samples that resemble the worst-case data shifts and corruptions according to the output of the last fully connected layer. These samples are meant to reduce the gap between the source domain and the unseen target domain. In reality, the introduction of these diverse out-of-distribution samples leads to the model instability, reflected in large inter-training and inter-step accuracy variance on all three test datasets.

With the lack of test data, the instability of the model can be foreseen through the topology of the second convolutional

ERM

	USPS	SVHN	SYNTH
Validation Loss	78.5±1.1	34.3±1.9	44.1±1.1
ConvTopRep	80.2±1.0	35.9±2.1	45.2±1.5

ME-ADA

	USPS	SVHN	SYNTH
Validation Loss	79.6±0.8	39.3±2.8	48.8±1.7
ConvTopRep	77.6±1.5	38.2±1.9	47.5±1.4

RandConv

	USPS	SVHN	SYNTH
Validation Loss	84.3±0.8	61.7±1.1	63.3±1.1
ConvTopRep	85.7±1.1	64.5±1.7	67.9±1.7

Table 9: Early stopping comparison for LeNet. The values represent the mean and the std of the test accuracy over 10 trainings.

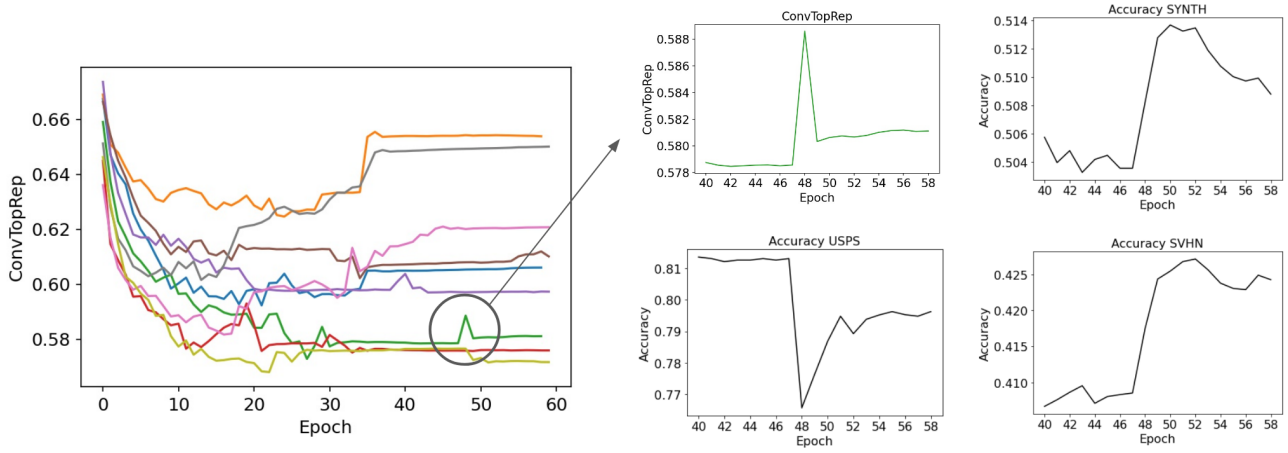


Figure 2: Left: Evolution of ConvTopRep during training of LeNet with ME-ADA. Right: Zoomed ConvTopRep (top left) and the accuracy of SYNTH (top right), USPS (bottom left) and SVHN (bottom right) for corresponding epochs.

layer, summarised by ConvTopRep. It exhibits the following pattern: during the first 9 epochs, the value of ConvTopRep decreases for all 10 trainings. However, after 9 epochs, the evolution of the topology is inconsistent within trainings as we observe significant fluctuations of ConvTopRep.

We observe a correlation between ConvTopRep and test accuracies. That is, the decrease in ConvTopRep does not always lead to the accuracy improvement on all the datasets, but rather indicates the change in test accuracy. For example, in Fig. 2, we showcase that the change in ConvTopRep leads to an increase in accuracy on SYNTH and SVHN datasets, but a decrease in the accuracy on USPS. As a consequence, using the smallest value of ConvTopRep as a criteria for early stopping in such an unstable scenario proves to be ineffective. In fact, it is outperformed by the validation loss-based early stopping on all three test datasets.

Random Convolution: We use mixing variant of Random Convolutions, where the original image is blended in with the output of Random convolutions to retain augmented images from being too far from the source distribution, and to continuously interpolate between the source domain and randomly sampled domain. Described interpolation results in more stable behaviour of the model.

It is worth noting that RandConv is a model-independent strategy. Unlike ME-ADA, that uses the last fully connected layer as a basis for adversarial sampling, the main focus of RandConv is to fine-tune convolutional layers to make them more robust against style perturbations. It is reflected in the evolution of the topology space of the second convolutional layer, shown in Fig. 3.

In particular, for the first 50 epochs the variance of the ConvTopRep between trainings is very small, which means that the

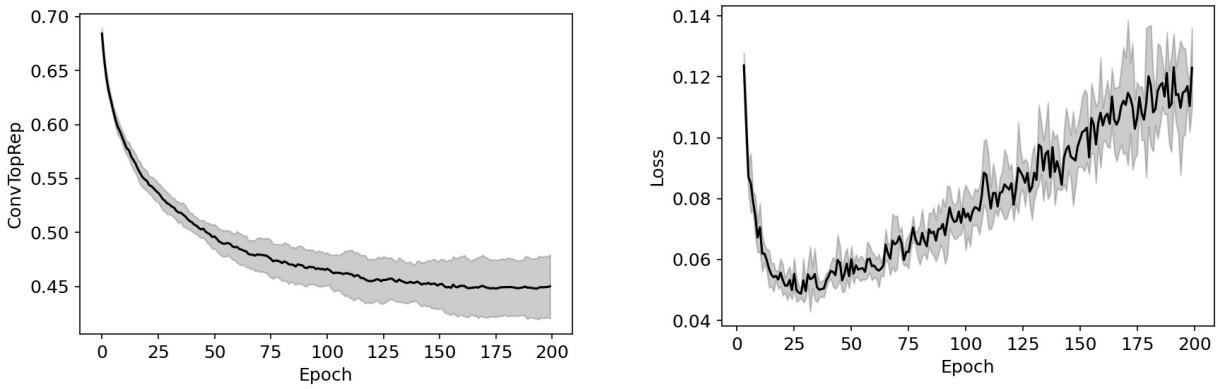


Figure 3: ConvTopRep and Validation loss for LeNet trained with RandConv. The solid line represents the mean, the grey area represents the variance over 10 trainings.

evolution of convolutional layers is consistent and stable. Furthermore, the value of ConvTopRep continuously decreases, indicating that the filter space becomes more structured, with filters becoming more connected. ConvTopRep eventually converges to a certain value after epoch 150.

Note that one of the main concerns of generative methods such as RandConv is their instability during training. This is due to the diversity of generated samples, which makes the network continuously changing. This behaviour is reflected in the values of ConvTopRep, which makes it a reliable measure of convergence for training.

Finally, for RandConv, validation loss indicates to stop earlier than ConvTopRep, when the model is yet under-trained, as shown in see Fig.3. As a result, it can be seen that for RandConv our proposed ConvTopRep significantly outperforms the early stopping using validation loss. The early-stopping comparison results, summarised in Table 9, show that using ConvTopRep leads to an average of 1.4% accuracy improvement on USPS, 3% improvement on SVHN, and 4.6% improvement on SYNTH.