Learning Visual Representation with Synthetic Images and Topologically-defined Labels

Shizuo Kaji[1] and Yohsuke Watanabe[2]

[1]Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan
[2]ZOZO inc., Fukuoka, Japan

Abstract

We propose a new scheme for convolutional neural networks to learn visual representation with synthetic images and mathematically-defined labels that capture topological information. Our learning is based on an advanced mathematical tool called homology, which is extensively used in the study of manifold theory. We show that the acquired visual representation supplements the one obtained by the usual supervised learning with manually-defined labels by confirming an improved convergence in training for image classification. Our method provides a simple way to encourage the model to learn global features through a specifically designed task based on topology. It requires no real images nor manual labels and can be utilised at a minimal extra cost.

1 Introduction

Convolutional neural networks (CNNs) have been very successful in acquiring visual representation in a data-driven manner and replaced human eyes in various tasks. However, there are differences between how CNNs and humans perceive images. For example, convolutional neural networks (CNNs) are famously known to be "short-sighted" by being biased to textural information [15]. This is because the convolution operation is local unless a huge kernel is used. There are different approaches to capturing global characteristics of images by neural networks. One way is to introduce a new model architecture. The attention mechanism has enabled to learn long-term dependencies in image as demonstrated by Vision Transformer [12]. Another way, which we concern in this paper, is to device a new training scheme, which works with virtually any model architecture with little modification. Topology is a study of shapes whose ultimate goal is to classify shapes by their global topological types. Topologists have invented various *topological invariants* that can discern different shapes. *Homology* is a particularly powerful topological invariant to classify manifolds that are locally the same Euclidean space but globally different. Moreover, homology resembles a certain function of human recognition, which identifies "components" and "holes" in the shape; homology turns this intuitive human perception to a computable quantity. Our idea is to design a task of computing (a generalisation of) homology from an input image by a CNN so that the model is encouraged to learn visual representation that is relevant to the topology of the image. In this way, we teach the mathematical idea to a CNN that learns in a data-driven manner.

Persistent homology (PH), one of the main tools of the emerging field of *Topological Data Analysis* (TDA), provides efficient machinery for computing global topological features of data [2]. Theoretically, PH provides a strictly stronger feature for graphs than any feature obtained by message-passing graph neural networks when applied to the graph isomorphism problem [19]. PH has also been proved to be practically useful for image processing such as classification [13] and segmentation [34]. In recent years, applications of persistent homology have expanded to many areas in science and led to new discoveries [16], but in most cases, it is used just as yet another feature extractor. In view of TDA, we proceed a step further and ask *if we can teach topology to a neural network* so that the model learns low-level image features together with the mechanism for computing high-level topological features by aggregating them. Our scheme is to train the model with regression of the vectorised persistent homology for synthesised images. The model is forced to learn relevant visual representation required to approximate persistent homology through this task, which necessarily involves the global structure of the image since PH cannot be computed locally. We demonstrate the validity of our scheme by experiments showing that a CNN trained by the proposed method can be further trained for image classification tasks to show improved convergence compared with one trained from scratch. The improved convergence means that the visual representation acquired during the training for PH adds complementary expressive capacity to the one obtained by the usual supervised learning with the class labels.

From a broader perspective, our scheme offers a new approach to letting neural networks learn mathematical

structures in data. If we want to teach a certain mathematical structure, e.g., topology to a neural network, we choose a mathematical invariant, e.g., homology, which is based on the structure. Given input data samples, we compute the chosen invariant by its mathematical definition, and train the model to learn the input-output relation for the data sample and its computed invariant. During the training, the model is asked to search for the relevant structures in the data to establish the input-output relation. This learning scheme provides a simple and general way to transfer mathematicians' knowledge to the model. One of the motivations of this paper is to give a proof-of-concept for this idea.

It is also worth pointing out that our scheme does not rely on real images nor labels, but uses mathematically generated images annotated with mathematically defined features. In this way, it is is free from human bias which lies not only in the manual annotation but also the image themselves; photos reflect the present world and the view of the photographer. In fact, the models trained with ImageNet by SSL, even without human annotated labels, are known to be subject to bias [33].

2 Related Work

The proposed method is built on three key ingredients, which we discuss in this section. We introduce some novel ideas to each of the three and combine them to develop our scheme.

2.1 Self-supervised learning on images

In self-supervised learning (SSL), image features are learned through specifically designed tasks, which we call pretext tasks, without relying on manually-defined labels. SSL has seen a great success practically by dispensing the labour of manual annotation as well as scientifically by showing the similarity to humans in the learning process. There are three major types of pretext tasks. The first task is to tell if given two images come from the same image or not, where variations of images are generated by applying transformations in spacial and colour domains (see [20] for a survey). The second task is to undone degradation, such as adding noise and masking, and reconstruct the original image. The third is similar to the second one but to perform a pair of (approximately) invertible processes, such as compression-expansion, as represented by the celebrated autoencoder [17]. All of the three tasks demand the model to acquire high-level representation of the images. The main objective of these methods lies in, more or less, capturing the *distribution* of the training data; to be good at in-painting or compression, one has to find a low-dimensional manifold which models the training data well. In contrast, we propose another type of task that put more emphasis on the *computation* process rather than the distribution of the data. When the computation is based on a certain mathematical structure of the data, a model will be incentivised to focus on the structure through learning the task. Our pretext task is to approximate the computation of persistent homology of the image. Persistent homology computed mathematically from the data is used as the label for a regression task. It is also notable that our pretext task does not rely on semantics or human perception but solely on mathematics. This allows us to use synthetic images whose distribution is very different from that of natural images. The procedure is completely free of real data.

2.2 Learning with synthetic images

Even though SSL saves the annotation costs, the preparation of training data is still a vexing problem. Publicly available datasets can be of low-quality, subject to bias, or violating usage rights and privacy. ImageNet, one of the most popular large-scale datasets, suffers from fairness issues [36], and there have been a growing interest in the fairness of machine learning [27]. [33] points out that even models trained on ImageNet with SSL without using labels learn racial, gender, and intersectional biases from the way people are stereotypically portrayed on the web. No matter how much care and attentions are paid for data collection, it is impossible to be free from these issues as long as real images are used. Using generative adversarial networks (GANs) to generate image datasets for training is a popular and successful strategy to mitigate the situation [5], but GANs are also trained with natural images and cannot avoid above-mentioned problems. A promising approach is to use algorithmically synthesised images. Formula-driven Supervised Learning introduced in [24] considers pretraining with synthetic images generated by a mathematical formula. The labels are assigned according to the parameters used for the image generation. Several different formuli are tested and an iterated function system, which generates fractal images, is found to be effective. A wider variety of image generation methods have been tested since then [4, 23]. To see how synthetic images are helpful in acquiring image features is interesting also in terms of cognitive science. In this paper, we also use synthetic images and try to learn topological features from them. Unlike fractal images, our synthetic images are not meant to capture some of the characteristics appearing in natural images, but they are not very meaningful for human eyes, Another difference between the Formula-driven Supervised Learning and ours lies in how the labels are generated. In the former, labels are associated with the parameters used for the image generation and tied to the image generation model. In the latter, we generate labels by a mathematical formula computed directly from the images, which has advantage of being applicable to any image generation model not restricted to the fractal model. The generated labels encode topological features and the model is encouraged to learn image features that are relevant for

approximating the topological features.

2.3 Persistent homology for image analysis

Since persistent homology has an unusual input and output, we briefly explain its definition and how it is used as an image feature extractor. PH takes a series of nested topological spaces and outputs a multiset of intervals of the real numbers. We view an image as a function $f : X \to \mathbb{R}$ defined over a rectangular domain X, and we obtain a series of nested spaces

$$\emptyset \subset X_{t_1} \subset X_{t_2} \subset \cdots \subset X_{t_m} = X,$$
$$X_{t_i} = \{(x, y) \in X \mid f(x, y) \leq t_i\},$$

where $t_m = \max(f)$. Applying the homology functor $H_d(-)$ with the coefficients in the field $\mathbf{F}_2$ with two elements, we obtain the corresponding sequence of $\mathbf{F}_2$ vector spaces, and this sequence is by definition the persistent homology $PH_d(X, f)$ of the pair (X, f). $PH_d(X, f)$ can be written as the direct sum of so-called the interval module having the form

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \cdots & \longrightarrow & 0 & \longrightarrow & \mathbf{F}_2 & \longrightarrow & \cdots \\ \| & & & & \cap & & \cap & & \\ H_d(\emptyset) & \longrightarrow & \cdots & \longrightarrow & H_d(X_{t_{i-1}}) & \to & H_d(X_{t_i}) & \longrightarrow & \cdots \end{array}$$

$$\begin{array}{ccccccccc} \longrightarrow & \mathbf{F}_2 & \longrightarrow & 0 & \longrightarrow & \cdots & \longrightarrow & 0 & \\ & \cap & & \cap & & & & \cap & \\ \longrightarrow & H_d(X_{t_{j-1}}) & \to & H_d(X_{t_j}) & \longrightarrow & \cdots & \longrightarrow & H_d(X). & \end{array}$$

We denote the summand by the interval $[t_i, t_j)$. When $d = 0$, it may happen that $j - 1 = m$, in which case we represent the summand by $[t_i, \infty)$. Figure 1 shows an example of PH of an image. The alien output as multiset can be transformed into a fixed-length vector using vectorisation techniques [1, 6, 9] so that it fits in the standard machine learning pipeline.

PH provides a shape feature that is complementary to conventional ones. [26] shows that combining PH with other descriptors results in an increased performance in object recognition. To utilise the expressive capacity of PH to complement neural networks, there are studies to assimilate PH into deep learning. A parametric representation of persistence homology with learnable parameters is introduced in [18] so that a task-optimal vectorisation is obtained in a data-driven manner. Dedicated architectures of CNNs are designed in [32] for computing vectorised PH of time-series and point clouds. The topological autoencoder [29] learns a latent space of a point cloud that preserves the topological structure in terms of persistent homology. [19] shows that PH is a strictly more expressive than message-passing graph neural networks in classifying graph isomorphism classes. In this paper, we rely on the fact that PH requires global information

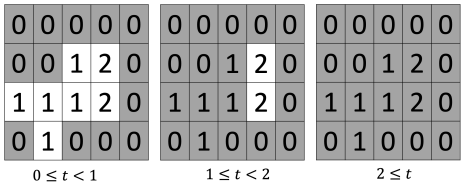

Figure 1: The figure shows the sublevel sets of an image for different ranges of t. For an image, persistent homology consists of two components $PH_0(X, f)$ and $PH_1(X, f)$, which respectively records the transition of islands and holes in the sublevel sets under different threshold. In this example, $PH_0(X, f) = \{[0, 1), [0, \infty)\}$, where the interval $[0, 1)$ corresponds to the connected component consisting of the single pixel at the bottom-left corner in the left-most figure, which is merged to the other connected component in the central figure. Since this feature exists for $t \in [0, 1)$, it is represented by the interval $[0, 1)$, and said to have the birth time 0, death time 1, and life time $1 (= 1 - 0)$. The other component that exists for $t \geq 0$ is represented by $[0, \infty)$, which has the infinite life time. Similarly, $PH_1(X, f) = \{[1, 2)\}$ whose element represents the hole surrounding the two pixels with the value 2 in the central figure. This hole disappears in the right-most figure, and so, it is represented by $[1, 2)$. Persistent homology of an image records the topological features, such as islands and holes, with the threshold in which the features emerge and disappear. The distinctive idea of *persistence* is to trace the emergence and disappearance rather than computing features at each threshold separately.

to compute so the model has to acquire a certain high-level image representation that encodes global topological structures in order to approximate PH, and this makes a good pretext task.

3 Method

Our proposed scheme can be viewed as a type of SSL in which desired image features are learned by solving a pretext task. The pretext task is so designed that makes the model focus on a particular type of image features. Our pretext task is *approximating the mathematical computation* of persistent homology of synthesised images. PH is defined from the binarisation of the input image. Hence, the model should pay more attention on global image characteristic than textural cues to successfully approximate PH. This makes the contrast to the natural tendency of CNNs to be biased to local information when trained for image classification with labels. Our pretext task does not require any natural images, but synthesised images (Sec. 3.1) are used. The model learns the input-output relation (regression) for the image and the corresponding vector computed through persistent homol-

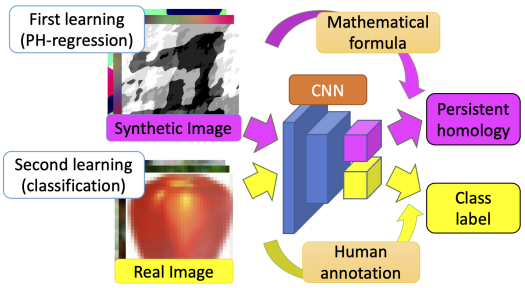

Figure 2: The overall structure of the proposed method.

ogy (Sec. 3.2). The model is expected to learn relevant image features which are required for approximating PH through this PH-regression task. Intuitively, the procedure is described as follows. The teacher knows topology and gives problems to the student with the answer computed mathematically. The student tries to guess how the teacher's answer is computed and search for clues in the image. So it is not guaranteed that the student learns how to compute PH in the way it is defined mathematically, but the student collects image features that are helpful to approximate PH, and those features are necessarily global and topological, as so is PH. This metaphor is also useful for explaining why our scheme complements the usual supervised learning for image classification. In the usual training with class labels, the student tries to find an "efficient way" or a shortcut just enough to solve the problem without paying much attention to all aspects of the problem. For example, to distinguish a cat from an elephant, it would be easier for a CNN to compare the skin texture than to identify the shape of the nose. One kind of homework is not enough for gaining a broad view on the subject, and our scheme adds a new exercise course for learning visual representation.

We note that any algorithmically computable vector associated with the image can be used in this scheme. However, we particularly choose PH since (1) it is proved to capture global topological characteristics of the image and is known to have discriminative power [1], (2) it has guaranteed stability (including robustness against the change in the pixel values and the invariance against isometric geometric transformations [10, 31]), and (3) it is defined for any image and is efficiently computed [22]. We elaborate on these points below.
(1) As the computation of PH is based on the global structure of the image, its approximation cannot be achieved by learning only local image features. Thus, the image features obtained by PH learning are topological and global, whereas local and textural features are well-captured by the usual supervised learning. We expect that learning topological features and local features separately in two steps will have complementary effects.
(2) Learning invariant features is one of the main principles in visual representation learning [28]. PH gives the same labelling for images that are rotated and reflected, and similar labelling for images with certain types of pixel value alteration, which is guaranteed by the stability theorems. The regression of PH encourages the model to learn those invariances. Furthermore, when a certain transformation of the image which changes its PH is applied, the model is asked to learn the *change in the label*; persistent homology is *functorial*. That is, the transformation in the input is systematically reflected by the transformation in the output, and the model is asked to learn this higher relation as well when trained with data augmentation.
(3) For randomly generated images, we cannot rely on semantics of the images that is meaningful for human eyes to design the pretext task. We do not have this problem with PH, which is mathematically defined. Moreover, computation of PH is relatively cheap (Appendix A.1) so that it does not increase overall training time substantially.

We evaluate our scheme empirically using a dataset consisting of natural images and manually annotated class labels. We measure the performance of a CNN model for image classification with the dataset in terms of its validation accuracy. Basically, we compare the following two models, both of which are trained in a supervised manner by the training split of the dataset but with different initialisations: (a) all weights are initialised randomly at the beginning (b) the weights of the convolutional layers are initialised with those of the model trained by our scheme for the PH-regression task (Figure 2 depicts a schema). If we see any increase in the performance of (b) against (a), it indicates that the model has learned image features through our scheme that are complementary to the ones naturally learned by the classification task. Further experiments which corroborate and explain the result of this evaluation are given in Appendix A.3 and Appendix A.4.

The experiments are conducted with our codes publicly available under the MIT license at `https://github.com/shizuo-kaji/PretrainCNNwithNoData`.

3.1 Image generation

We use the following frequency-based random image generation method to create a synthetic dataset. We choose this image generation model mainly because it is computationally inexpensive and produces images with various frequency profiles (see also Appendix A.5). In particular, the resulting images (see Figure 3) contain patterns at different scales, and hence, have rich topological information.

1. Create an array h of dimension 256×256, where the values are drawn independently from the uniform distribution on $[0, 1]$.

2. For a frequency parameter β drawn from the uniform

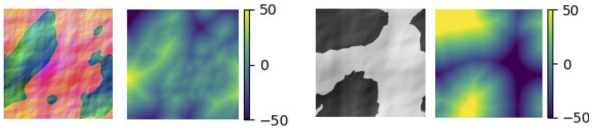

Figure 3: Two examples of synthesised images and their signed distance functions. Persistent homology is computed for the sequence of the sublevel sets for the signed distance function.

distribution on $[1, 2]$, set

$$g(x, y) = \mathrm{Re}\left(\mathrm{iFFT}\left(\frac{\mathrm{FFT}(h)(x, y)}{((x + 1)^2 + (y + 1)^2)^\beta}\right)\right), \tag{1}$$

where FFT is the 2D discrete Fourier transform and iFFT is its inverse, and Re denotes the real part of a complex number.

3. With a probability of p, which we set to 0.5, binarise g by Otsu's thresholding [30].

4. Repeat the process three times independently to create a colour image with RGB channels.

5. Convert the image into greyscale with a probability $q = 0.5$.

The random parameter β controls how fast the high-frequency components decay. The effect of the choices for the hyper-parameters p, q and the range of β does not seem to be large and we haven't done a comprehensive search. We have only checked that setting $p = 0.5$ and $q = 0.5$ is slightly better than $p = 0, 1$ or $q = 0, 1$.

3.2 PH label computation

The images labels (vectors) for the PH-regression task are computed using persistent homology. There are various ways to utilise PH to encode topological information of images (see [35, 14] for a survey). Our choice aims at extracting shapes contained in the image by their contours and capturing its topological features such as connected components and holes, together with their scale. To this end, a nested sequence of spaces, which is the input for PH, is built from the image by the following procedure: (1) Convert the image into greyscale (2) Binarise it using Otsu's thresholding [30] (3) Compute the signed distance function (see Figure 3)

$$\phi(x, y) = \begin{cases} -\min_{(x',y'):\text{background}} |(x, y) - (x', y')| \\ \qquad \text{if } (x, y) \text{ is foreground} \\ \min_{(x',y'):\text{foreground}} |(x, y) - (x', y')| \\ \qquad \text{if } (x, y) \text{ is background.} \end{cases} \tag{2}$$

(4) Construct the sequence of the sublevel sets $X_t = \{(x, y) \in X \mid \phi(x, y) \leq t\}$, where X is the domain of the original image; that is, the rectangular grid. This procedure gives a bounded nested sequence satisfying $X_s \subset X_t$ when $s < t$ and $X_{-D} = \emptyset, X_D = X$, where D is the length of the diagonal of the image. The PH (with the coefficients in the field $\mathbf{F}_2$) of this sequence is computed by the software called Cubical Ripser [22]. An important remark is that instead of considering the sublevel sets with respect to the original (greyscaled) pixel values, we consider the sublevel sets of the signed distance function. In this way, the resulting PH captures the metric-aware structure of the original image; the scale of topological features are encoded as life time in persistent homology. We also emphasise the fact that the PH label is computed from binarised and signed distance transformed images from which textural information is stripped off. The PH label encodes topological features with their scale, and solving the regression for the label requires recognition of the global structure of the image.

The degree 0 and 1 parts of persistence homology are vectorised separately and concatenated into a single vector, which serves as the label for the image. We test four popular vectorisation techniques; the persistence image [1], the persistence landscape [6], the Betti number curve [9], and the birth-life histogram. The last one is simply the histograms of the birth time (the left end of the interval) and the life time (the length of the interval) of the persistence homology for each degree. The Betti number curve is nothing but a sequence of Betti numbers (the dimension of homology) for each sublevel set X_t. Hence, it does not carry any information on the relation between X_t and X_s for different thresholds s and t; the Betti number curve can be obtained by computing the usual homology (not persistent homology). We will see that not only homology but also its persistence plays a role by comparing the result the Betti number curve against the other three that consider the change of sublevel sets with respect to different thresholds in terms of life time in persistent homology.

All the vectorisation techniques have hyper-parameters for the "resolution" of the output, which are determined from the specified output dimension in our experiments.[1] In most of the experiments, the output dimension is fixed to 200. We allocate the same dimension for the degree 0 and 1 persistence homology so that we obtain a pair of 100-dimensional vectors that are concatenated to form a 200-dimensional output vector. To reduce the dynamic range and suppress overflow, the square root is taken for each coordinate of the vector. Although we may elaborate on the hyper-parameter tuning for the PH vectorisation, Tab. 1 shows that even the choice of the vectorisation techniques does not have a large impact. Therefore, we limit ourselves to an essential set of experiments to find

[1] The only non-canonical choices among the standard set of hyper-parameters of the vectorisation techniques are the number of *landscape functions* for the persistence landscape and the sigma of the Gaussian kernel used in the persistence image, which we set to 2 and 1.0 respectively.

out the nature of the scheme as detailed in Sec. 3.3.

3.3 Evaluation result

We compare the performance of a CNN model trained for a classification task with different initialisation methods. In addition to random initialisation (learning from scratch), we compare four types that rely on different pretext tasks and data: the PH-regression with synthetic images (ours), contrastive learning with synthetic images (popular in SSL), classification with synthetic images (FractalDB [24]), classification with natural images (the usual supervised learning for image classification). We call the training for the pretext task *the first learning*, and the training for classification with the target dataset *the second learning*. At the beginning of the second learning, the fully-connected layer of the model is replaced with a randomly initialised one. and the model is trained in a supervised manner with the target dataset with its class labels. The evaluation is based on the behaviour of the validation accuracy of the second learning. The convergence of the validation accuracy reflects the amount of visual representation obtained during the first learning in addition to the one obtained during the second learning. We set FractalDB as the main comparison target since both our method and FractalDB rely on no real images.

For the target dataset used in second learning, along with the popular ImageNet-1k (IMN-1k) [11] and CIFAR100 (C100) [25] consisting of natural images, we choose the animal dataset (ANM) [3], which is of relatively small size and consists of 2,000 binary images of animal contours of various size that are labelled with 20 classes. Since no texture information is present in the animal dataset, it is used in [18] for evaluation of their method of incorporating topological techniques into deep learning. We split the animal dataset into two sets with 1,600 training images (80 images per class) and 400 validation images.

The hyper-parameters for learning are fixed in a standard manner as follows. ResNet50 network architecture is used. Input images are resized to 256×256 and then cropped randomly down to 224×224. Random horizontal flipping is applied. We vary the number of epochs according to the dataset; both in first and second learning, 90 epochs for the CIFAR100 dataset, 90 epochs for synthesised dataset, 300 epochs for the animal dataset, and 45 epochs for ImageNet. The stochastic gradient descent with a momentum of 0.9 and weight decay of 10^{-4} is used as the optimiser. The initial learning rate is set to 0.1 for the first learning and 0.01 for the second learning, and multiplied by 0.1 twice at the 1/3 and the 2/3 of the total training epochs. The batch size is set to 128.

Table 1 shows the validation accuracy for the second learning initialised with different first learning schemes: **Scratch** is without first learning and the weights are initialised randomly. **Label** is trained in a supervised manner with the training dataset using its labels.[2] That is, the model is trained twice (in first and second learning) with the same training dataset but with different learning rates. This is to make sure that the model is trained for enough number of iterations. The four PH-based ones are trained for the PH-regression with the synthetic dataset with 400,000 images described in Sec. 3.2 with different PH vectorisation targets (**PH-PI** with the persistence image, **PH-LS** with the persistence landscape, **PH-BC** with the Betti number curve, and **PH-HS** with the birth-life histogram). **MoCo-v2** is trained by a popular SSL scheme MoCo-v2 [8] with the same synthetic dataset with 400,000 images as the PH-regression. For the FractalDB training, we use the publicly available weight file.[3] **FDB-1k** is trained for classification with the FractalDB dataset consisting of 1,000,000 images with 1,000 classes and **FDB-10k** is trained for classification with the larger FractalDB dataset consisting of 10,000,000 images with 10,000 classes. The values in Tab. 1 differ slightly from the ones reported in [24], possibly due to minor differences in the experimental settings. For **IMN**, we use the publicly available weights of the ImageNet-1k trained model provided as a part of PyTorch's torchvision library (version 0.10.1). Table 1 also includes the timing for the first learning. Although publicly available weight files are used for **FDB-1k** and **IMN**, the hours for **FDB-1k** and **IMN** are measured on our system. For **FDB-10k**, the hours of **FDB-1k** is extrapolated by simply multiplied by ten.

Since there are only 400 validation images for the animal dataset, the accuracy values fluctuate. The values listed in Tabs. 1 and 2 are the mean for the last epochs from 290 to 300. The 90 percentile falls within ± 0.6 of the mean. Moreover, to see the stability of the results, we have performed ten trials of first and second learning with different random seeds for the entry of **PH-PI** for C100 in Tab. 1, and have observed that the 90 percentile falls within ± 0.3 of the mean.

The models that underwent the PH-based first learning show better convergence than **Scratch**, **Label**, **MoCo-v2**, and **FDB-1k**, and comparable convergence to **FDB-10k**. We note that the PH-regression is performed with a much smaller number of 400,000 images and about 70 times less computation time than **FDB-10k**. All four PH-based models show more or less similar convergence regardless of the used PH vectorisation techniques. Among them, **PH-BC** performs the worst, and this can be attributed to the fact that only **PH-BC** does not care about the "persistence"; as explained in Sec. 3.2, **PH-BC** can be computed independently for each X_t, while the other three, **PH-PI**, **PH-LS**, and **PH-HS**, capture the

[2] The deteriorated accuracy of Label for Animal compared to Scratch is likely due to over-fitting. We observe the validation loss converges quickly although to a higher value.

[3] `https://github.com/hirokatsukataoka16/FractalDB-Pretrained-ResNet-PyTorch`

relation among different X_t's.[4] This suggests that not only homology but also its persistence structure helps learn visual representation.

Figure 4 shows the transition of the accuracy in the second learning of the models selected from Table 1 for the CIFAR100 dataset. A similar figure for the ImageNet-1k dataset is given in Appendix in Figure 5. **IMN** surpasses all other models in both training and validation. This is reasonable since only **IMN** relies on the external dataset among the models in Table 1. The convergence behaviour of **PH-PI** and **FDB-10k** are quite similar, sitting between **IMN** and **MoCo-v2**.

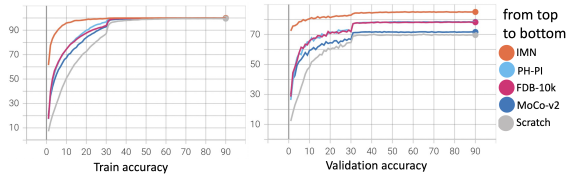

Figure 4: Transition of the training and validation accuracy of the CIFAR100 dataset.

To see the nature of our PH-based learning, we conduct a few more experiments by changing parameters for the PH-regression in the first learning. We fix the vectorisation method to the persistence image. Unless otherwise stated, the output dimension is fixed to 200 and the size of the synthetic image dataset is fixed to 200,000.

The impact of the choice of the vectorisation dimension of PH is assessed by varying the value among $\{100, 200, 400, 800\}$. As we observe in Tab. 2 (left), the choice affects the performance and the optimal value, in this case, is about 200. However, we guess the optimal value depends on the complexity and the dimension of the synthesised images as the vectorisation dimension controls the resolution of the discretisation of PH.

The impact of the size of the dataset is assessed by varying the number of synthesised images among $\{50,000, 200,000, 400,000, 800,000\}$. As is observed in Tab. 2 (middle), the performance increases as the size gets larger. It is interesting to note that **PH-PI** performs better than **Label** in Tab. 1 even at the size of 50k, which is the same as that of the CIFAR100 training dataset. In this case, **PH-PI** and **Label** are trained with the same number of iterations with the same learning rates. The model gains more by solving different kinds of problem (PH-regression and classification) than by solving a single problem (classification) over and over again in the same duration.

The PH-regression task also works with real images in place of synthetic ones. In the next experiment, we use the training split of the CIFAR100 dataset (and the animal dataset, respectively) for the PH-regression task in the first learning, and then use the class label in the second learning. The result is shown in Tab. 2 (right). In this example, the setting of **Label** and **PH-C** (**PH-A**, respectively) for the CIFAR100 dataset (the animal dataset, respectively) is the same except for the label used in the first learning; **Label** uses the class label while **PH-C** (**PH-A**, respectively) uses the label computed with the persistence image. We see the improvement in the performance in the latter, which indicates the benefit of learning not only from the class labels but also from topology-based labels. Comparing the result of **PH-C** with the entry for 50k in Tab. 2 (middle), we see our synthetic dataset offers slightly better quality for learning than the CIFAR100 dataset, which consists of the same number of 50k natural images. This could be attributed to the design of the image generation model that produces patterns at various scales. When the animal dataset is used for the PH-regression (**PH-A**), the performance gain is much smaller. This is explained by the small number of training images (1,600). The topological variety in the training dataset is too limited. This observation agrees with the result in Tab. 2 (middle) that shows the impact of the size of the synthetic dataset.

4 Limitations

Our scheme is based on the assumption that CNNs learn different image features from different tasks. In our scheme, a model is trained sequentially with the regression task of persistent homology and the usual image classification task with human-annotated labels. This strategy is analogous to the heuristics in multi-objective optimisation in which multiple cost functions are optimised sequentially. However, this leaves us some questions; we may stack more learning steps than two, and the order of learning may affect. Furthermore, there are many SSL pretext tasks which target at different image features. To devise a more sophisticated way to combine different learning schemes than learning sequentially is an interesting research direction.

Another fundamental question, which is related to the explainability just as in most deep-learning schemes, is to understand what features are really learned from the PH-regression task and how different they are from the ones learned in other supervised or self-supervised manners. To fully answer this question requires a substantial advance in explainability of deep learning in general, and is beyond the scope of this work. Instead, we provide a few empirical results in Appendix A.3. From a theoretical perspective, it is also interesting to investigate to what extent PH of an image is approximatable by CNNs. This is challenging since universal-approximation-theorem-type results on CNNs is quite limited.

[4] Mathematically, persistent homology is (a sequence of) the ordinary homology equipped with the action of $\mathbb{R}$. We can interpret this result as indicating that not only the vector space structure but also the module structure is relevant to visual learning. In general, mathematical invariants together with the rich structure on them should be utilised in machine learning.

Table 1: Comparison of various first learning in terms of the validation accuracy of second learning with three datasets. The hours taken for the first learning are measured on a PC with a single NVIDIA RTX 3090 and an Intel Core i9-10850K.

	Scratch	Label	PH-PI	PH-LS	PH-BC	PH-HS	MoCo-v2	FDB-1k	FDB-10k	IMN
C100	69.6	70.3	78.4	78.1	76.6	77.9	71.6	75.3	78.1	85.0
ANM	80.7	80.1	91.0	90.1	89.1	90.6	85.0	84.6	85.2	93.3
IMN-1k	70.8	NA	72.9	72.0	71.4	72.8	71.0	71.8	72.4	75.3
time	0h	3h	22h	22h	22h	21h	38h	144h	1442h	74h

Table 2: Validation accuracy of second learning with C100 and ANM for the models trained with different parameters of the PH-regression in first learning: (Left) varying the dimension of the PH vectorisation. (Middle) varying the size of the synthesised dataset. The values for 400k are reproduced from Tab. 1. (Right) using the CIFAR100 (PH-C) and the animal (PH-A) datasets respectively in place of the synthetic dataset. The values for Scratch are reproduced from Tab. 1.

	100	200	400	800	50k	200k	400k	800k	Scratch	PH-C	PH-A
C100	76.8	77.7	77.4	75.0	76.1	77.7	78.4	78.8	69.6	75.3	72.4
ANM	87.4	89.9	90.5	87.5	88.6	89.9	91.0	91.6	80.7	86.5	83.2

5 Conclusion

We proposed a scheme for learning visual representations that were relevant to the topological features of images through a regression task for a mathematical function defined by persistent homology. Our method can be applied to any image dataset and virtually any neural network architecture with a small cost, yet achieves promising results. Our experiment with a synthetic image dataset suggested that even images that were little meaningful for human eyes could be used for learning certain visual representation. Our main contributions are two fold: (I) We proposed a pretext task that was mathematically defined and applicable to synthetic images without labels. (II) We experimentally showed that a convolutional neural network trained with the pretext task acquired visual representation that complemented those obtained by the usual supervised learning.

We would like to conclude the paper with a few possible future directions to investigate. (a) Since global topological features are generally robust compared to local ones, learning them could add resistance against adversarial attacks. (b) Mathematical invariants such as PH can be computed algorithmically. Instead of giving a concrete instruction of the algorithm to the computer, we let a neural network guess how the invariant can be computed from examples consisting of input and output pairs. In this way, the model learns, in a data-driven manner, the *mathematical structure* in data which the invariant quantifies. Regressing other mathematical invariants than PH could be used to train neural networks to equip them with an understanding of the geometric, topological, and algebraic structures of the image. This would provide a generic method to incorporate various knowledge established by mathematicians over past centuries into deep learning research. We believe this is a novel and interesting direction in machine learning. (c) Our experiments raise a question on the theoretical study of the relation between neural network architecture and learning capacity. It is known that universal-approximation-theorem-type statements have some constraints that relate to the topology of the target function to be learned and the width of neural networks [21]. It would be interesting to investigate what kind of topological invariants are computable by neural networks.

References

[1] 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. *J. Mach. Learn. Res.*, 18(1):218–252, January 2017.

[2] Henry Adams and Michael Moy. Topology applied to machine learning: From global to local. *Frontiers in Artificial Intelligence*, 4:54, 2021.

[3] Xiang Bai, Wenyu Liu, and Zhuowen Tu. Integrating contour and skeleton for shape classification. In *Int. Conf. Comput. Vis. Worksh.*, pages 360–367, 2009.

[4] Manel Baradad, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. In *Adv. Neural Inform. Process. Syst.*, 2021.

[5] Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *Int. Conf. Acoustics, Speech, & Sign. Process.*, pages 1–5, 2020.

[6] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, January 2015.

[7] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[9] Yu-Min Chung and Austin Lawson. Persistence curves: A canonical framework for summarizing persistence diagrams, 2021. arXiv:1904.07768v4.

[10] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*, SCG '05, page 263–271, New York, NY, USA, 2005. Association for Computing Machinery.

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

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021.

[13] Olga Dunaeva, Herbert Edelsbrunner, Anton Lukyanov, Michael Machin, Daria Malkova, Roman Kuvaev, and Sergey Kashin. The classification of endoscopy images with persistent homology. *Pattern Recognition Letters*, 83:13–22, 2016.

[14] Adélie Garin and Guillaume Tauzin. A topological "reading" lesson: Classification of MNIST using TDA. In *Proceedings of the IEEE International Conference on Machine Learning And Applications (ICMLA)*, pages 1551–1556. IEEE, 2019.

[15] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *Int. Conf. Learn. Represent.*, 2019.

[16] Barbara Giunti. TDA-Applications: Zotero database. `https://www.zotero.org/groups/2425412/tda-applications`, 2021. [Online; accessed 14-Oct-2021].

[17] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[18] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Adv. Neural Inform. Process. Syst.*, 2017.

[19] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

[20] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1), 2021.

[21] Jesse Johnson. Deep, skinny neural networks are not universal approximators. In *Int. Conf. Learn. Represent.*, 2019.

[22] Shizuo Kaji, Takeki Sudo, and Kazushi Ahara. Cubical ripser: Software for computing persistent homology of image and volume data, 2020. arXiv:2005.12692.

[23] Hirokatsu Kataoka, Ryo Hayamizu, Ryosuke Yamada, Kodai Nakashima, Sora Takashima, Xinyu Zhang, Edgar Josafat Martinez-Noriega, Nakamasa Inoue, and Rio Yokota. Replacing labeled real-image datasets with auto-generated contours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21232–21241, June 2022.

[24] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. In *Asian. Conf. Comput. Vis.*, 2020.

[25] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[26] Chunyuan Li, Maks Ovsjanikov, and Frederic Chazal. Persistence-based structural recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[27] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), July 2021.

[28] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[29] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 7045–7054, 2020.

[30] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[31] Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams, 2021. arXiv:2006.16824.

[32] Anirudh Som, Hongjun Choi, Karthikeyan Natesan Ramamurthy, Matthew P. Buman, and Pavan K. Turaga. Pi-net: A deep learning approach to extract topological persistence images. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 3639–3648, 2020.

[33] Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 701–713, New York, NY, USA, 2021. Association for Computing Machinery.

[34] Naoya Tanabe, Shizuo Kaji, Susumu Sato, Tomoo Yokoyama, Tsuyoshi Oguma, Kiminobu Tanizawa, Tomohiro Handa, Takashi Sakajo, and Toyohiro Hirai. A homological approach to a mathematical definition of pulmonary fibrosis and emphysema on computed tomography. *J. Appl. Physiol.*, 1985:601–612, 2021.

[35] Renata Turkeš, Jannes Nys, Tim Verdonck, and Steven Latré. Noise robustness of persistent homology on greyscale images, across filtrations and signatures. *PLOS ONE*, 16(9):1–26, 09 2021.

[36] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 547–558, New York, NY, USA, 2020. Association for Computing Machinery.

A Appendix

A.1 Computational cost of PH

The worst-case complexity of the standard matrix reduction algorithm for computing persistent homology is K^3, where K is the number of columns, which is linear with respect to the number of pixels in the cubical complex. The average time of computation for persistent homology and its persistence image as in the PH-PI pretraining for a synthesised image described in Sec. 3.1 in different sizes is shown in Tab. 3. The increase rate looks almost linear with respect to the number of pixels in this particular set of images.

A.2 Transitions of statistics during the second learning

The transition of the accuracy during the second learning of the models selected from Table 1 for the ImageNet-1k dataset is given in Figure 4. We observe a similar tendency to Figure 4 for the CIFAR100 dataset.

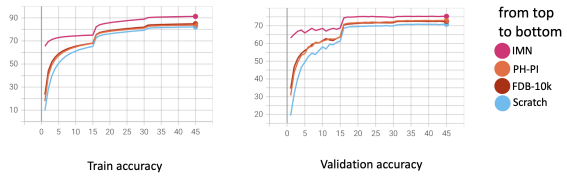

Figure 5: Transition of the training and validation accuracy during the second learning of the selected models with ResNet50 for the ImageNet-1k dataset.

To see if the proposed scheme is still effective when the CNN architecture is changed to a more powerful one, we performed an experiment with ResNet101. The experiment configuration was the same as in Table 1 except for the model architecture. Figure 6 shows the transition of the accuracy during the second learning of the selected models with ResNet101 for the CIFAR100 dataset. We observe a similar tendency to Figure 5 for ResNet50. Note that due to the large computational time required for training **FDB-10k**, we could not include **FDB-10k** in the figure.

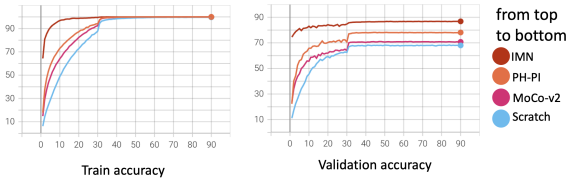

Figure 6: Transition of the training and validation accuracy during the second learning of the selected models with ResNet101 for the CIFAR100 dataset.

A.3 What is really learned?

Understanding and explaining what the model is learning is a difficult problem, which forms one of the central topics in machine learning research. We note that there

Table 3: Average computation time for PH and persistence image

image size	256×256	512×512	1024×1024	2048×2048
computation time in ms	22.62	89.85	436.93	2181.13

is currently no solid understanding on the visual representation obtained by convolutional neural networks even by supervised learning with ImageNet or by SSL schemes. Here, we give some evidence that indicates the difference between the models trained in our scheme and in the classical supervised manner with real images and manual labels. First, we observe a distinction in the weights of the first few convolution layers among different models. Figure 7 shows that the model learns different low-level features through our scheme.

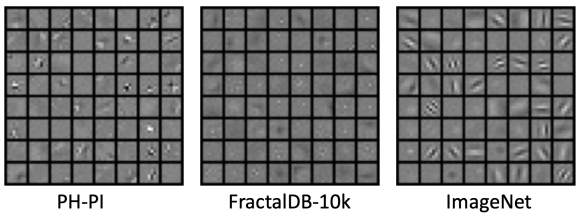

Figure 7: Visualisation of the filters of the first convolutional layer of PH-PI, FractalDB-10k, and ImageNet-1k trained models.

To understand this phenomenon, we conducted an experiment "the other way around" to see if the visual representation acquired through supervised learning with real images is sufficient to approximate persistent homology. Specifically, we replaced the fully-connected layer of the ImageNet-1k trained model with a randomly initialised one. Then, the model was trained for the PH-regression task as in the PH-PI pretraining while the weights of all but fully-connected layers were frozen (we call this model **IMG_fr**). The model was compared with the one trained from scratch (by updating all layers), which was precisely in the same manner as **PH-PI** in Sec. 3.3. Figure 8 shows that the convergence of the validation loss, as well as the training loss, of **IMN_fr** is much worse than that of **PH-PI**, indicating that the image features learned by the convolutional layers through a supervised training with ImageNet-1k is not sufficient for approximating PH. This explains what we have observed in Figure 7; the convolutional layers of **PH-PI** learn features specific to PH computation, which are different from those features learned with a supervised image classification task with ImageNet. This behaviour sharply contrasts with the fact that the ImageNet-1k trained model can be finetuned for various tasks by updating only the weights of fully-connected layers. As the visual representation obtained by the usual classification is insufficient for approximating PH, training a CNN specifically with the PH-regression task as in our scheme supplements to acquire more versatile visual representation.

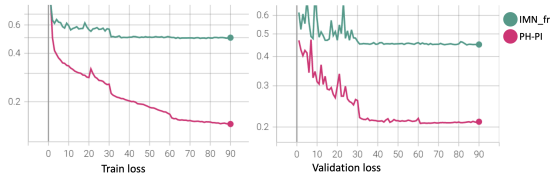

Figure 8: Transition of the loss for the PH-regression task for the ImageNet-1k trained model with the weights of all but fully-connected layers frozen (**IMN_fr**) and a model with randomly initialised weights (**PH-PI**).

To investigate this direction further, we performed another experiment of the PH-regression with a different set of synthesised images with simple random blobs which had virtually no texture information (Figure 9 (Top)). Note that this set of images is very different from the ones Figure 3 used for PH-PI. We used the ImageNet-1k trained (**IMN_fr**) and the PH-PI trained (**PH-PI_fr**) models while the weights of all but fully-connected layers were frozen. The transition of the loss values is shown in Figure 10. The validation loss of **PH-PI_fr** converges to a lower value than that of **IMN_fr**. This shows the visual representation learned by the convolutional layers through **PH-PI** is useful to approximate the PH of a different type of image than was used for pretraining. On the contrary, the visual representation learned by the supervised learning with ImageNet-1k is not sufficient to approximate PH, even for very simple images. Figure 10 also shows the transition of the loss values for **IMN** and **PH-PI**, where all layers including convolutional ones were updated during the second learning. Both models had better convergence than the ones updating only the fully-connected layers, but **PH-PI** had lower loss values than **IMN** indicating that updating all layers did not completely override the learned features during the first learning.

The visualisation of Grad-CAM++ [7] (with the target of the last convolution layer) of the two models, **PH-PI_fr** and **IMN_fr**, is shown in the middle and bottom rows of Figure 9. In each of the first, third, and fourth column from the left, the single large connected component corresponds to a single lump in **PH-PI_fr** (Bottom), whereas two or more lumps cover the edge of the component in **IMN_fr** (Middle). We may interpret this as that **PH-PI_fr** tends to identify the large shape as it is whereas **IMN_fr** finds it as a collection of edges.

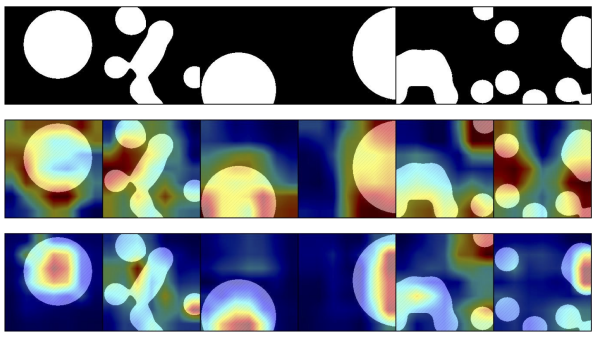

Figure 9: (Top) randomly synthesised blob images (Middle) Grad-CAM++ visualisation of **IMN_fr**, and (Bottom) **PH-PI_fr**

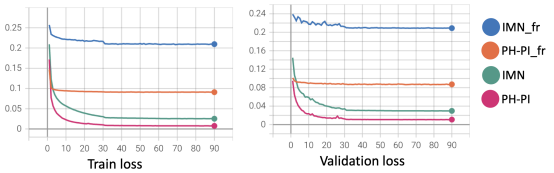

Figure 10: Transition of loss for PH-regression task with the random blob images. The ImageNet-1k trained model (**IMN**) and the PH-PI model (**PH-PI**) trained with the synthetic dataset described in Section 3.1 are further trained for the PH-regression task for another set of synthetic images (Figure 9(Top)). Corresponding models with the weights of all but fully-connected layers frozen during the second learning are denoted by **IMN_fr** and **PH-PI_fr**, respectively.

A.4 Class-wise inspection on the CIFAR100 classification

Here, we investigate what kind of images are better classified with our scheme. We looked at the predictions for the CIFAR100 validation dataset (100 classes $\times$ 100 images) of the three models, **Label**, **PH-PI**, and **IMN** selected from Table 1, after second learning. We computed the F_1-score for each of the 100 classes and sorted the result in terms of (F_1-score of **PH-PI**) - (F_1-score of **IMN**). Having a positive value of this means that **PH-PI** performed better than **IMN** for the class. The result is shown in Table 4. Although it is not shown on the list, we note that **PH-PI** had improved F_1-scores than **Label** for all 100 classes.

Figure 11 shows sample images from the six classes that appear in Table 4. We selected three images from each class in the following manner. For the classes of bear, snake, and shrew, with which **IMN** performed better then **PH-PI**, we selected those images where the prediction of **PH-PI** was incorrect and that of **IMN** was correct. Then, we chose three images with the lowest ranking of the true label in the prediction of **PH-PI**. For the classes of plate, road, and girl, with which **PH-PI** performed better than **IMN**, we did the same by swapping the role of **PH-PI** and **IMN**. The caption below each sample image is the prediction of the incorrect model. How the models make incorrect predictions is interesting and it seems to agree with our claim that the model trained with the PH-regression task (**PH-PI**) put more emphasis on the shape than the texture. Especially the plate class provides an illustrative example; **PH-PI** focuses on the round shape to answer correctly, while **IMN** looks at the texture of the plate. We give a few more interpretations of the result: The snake in the second row can be mistaken as a bottle if we look at its silhouette without its distinctive texture. The girl in the third row wears colourful striped clothes but cannot be mistaken as a lizard by a human. As all images of road consist mainly of large and simple geometric shapes, which are easily characterised by persistent homology, they are correctly answered by PH-PI.

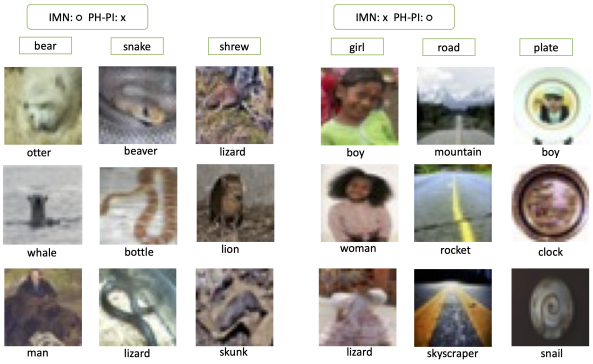

Figure 11: Sample images of the classes in Table 4. The label of each image indicates the (incorrect) prediction by **PH-PI** (for the classes bear, snake, and shrew) and by **IMN** (for the classes girl, road, and plate).

A.5 Image synthesis method

There are many different ways to synthesise images, such as various noise generation methods. We have not conducted a thorough test on dataset creation since the focus in the design of our pretext task is not on the distribution of the data. The key is in the process of approximating a certain computation (in our case, persistent homology) based on mathematical structure. We list some reasons why we chose the particular image synthesis method. We can see from Equation (1): (i) It is rotation invariant, except for the fact that the image is rectangular. (ii) It is scale (image size) independent. (iii) It produces various frequency profiles: The lifetime profile of the persistent homology is a very important factor of the topology of the image. With the signed distance transform, the total persistence (the sum of lifetime of all cycles) is bounded so that we can have either many short lifetime cycles or a small number of long lifetime cycles. In natural

Table 4: F_1-scores of the predictions of selected models from Table 1 for the validation dataset of C100. The list is sorted in the increasing order of the difference in the F_1-score between **PH-PI** and **IMN** and we show the first three and the last three classes.

class	Label	PH-PI	IMN	(PH-PI)-(IMN)	(PH-PI)-(Label)
bear	0.524	0.653	0.854	-0.201	0.129
snake	0.617	0.670	0.831	-0.161	0.053
shrew	0.545	0.573	0.725	-0.152	0.028
...					
plate	0.710	0.777	0.772	0.006	0.068
road	0.906	0.927	0.916	0.011	0.020
girl	0.472	0.653	0.623	0.030	0.181

images, short lifetime cycles arise mainly from noise or texture, which are local. The denominator of Equation (1) discourages the resulting image from having many high-frequency components so that there are more long lifetime cycles, which represent global topology. (iv) It is computationally cheap. We emphasise that the fractal model used in Formula-driven Supervised Learning [24] requires tremendous computational power and their datasets are prepared on a computing cluster. Creation of our dataset with 400,000 images takes less than twenty minutes on a personal computer, and is generated on the fly during the first epoch of training.