Towards Robust Perception using Topological Invariants

Romie Banerjee[1] , **Feng Liu**[1] and **Pei Ke**[1]
[1]Huawei Munich Research Center
{romie.banerjee, feng.liu1, peike}@huawei.com,

Abstract

Persistent homology is a tool from topological data analysis that can be used to define functions that extract topological features from images. These derivatives of persistent homology are invariant under continuous symmetries in the image space (for example translation or rotation). We provide theoretical argument for using such functions in conjunction with convolutional filters to improve perception algorithms. We discuss using these functions as a method for adding topological inductive priors on the convolutional network. We propose a coupled UNet architecture for learning sematic segmentation in the presence of topological priors involving restrictions on the persistent betti values of the segments. We run experiments on perception data for autonomous driving and are able to report qualitative improvements.

1 Introduction

Autonomous driving presents a variety of perception problems. Traditionally deep learning methods involving convolutional neural networks have been employed with great success to address these problems. In spite of its tremendous success deep learning methods are known to be extremely brittle. The source of the brittleness is the dependence of the resulting algorithm on the training data. As a result the trained models are sensitive to distributional shifts in the test data. We apply methods from topology to address this problem in the case of semantic segmentation. The key insight is to use topological loss functions during training. Loss functions derived from persistent homology can be used to put topological constraints on the output of the network. Classical machine learning leverages local geometry, as can be seen from the Euclidean loss functions. Topological Data Analysis (TDA) produces invariants of the global geometry of the input. The constraints introduced through topological loss can be utilized as topological prior assumptions on the model or to reduce distance in topological feature space to the training images. Persistent homology has two important properties of invariance under continuous symmetries and Lipschitz stability. The topological features we consider are therefore very well suited to the robust vision problem.

1.1 Main contributions

In this paper we address the lack of robustness of current convolutional neural networks when applied to test data obtained from a distribution different from the training data distribution. As a concrete example we consider the task of semantic segmentation of street scenes in adverse weather conditions with a neural network trained on images in good weather. We start with a baseline convolutional network, the UNet ([Ronneberger *et al.*, 2015]), and propose a coupled UNet architecture that allows training with topological priors and losses. We provide empirical evidence that the topological coupling improves performance of the baseline UNet on both the regular driving test data and adverse weather shifted test data.

1.2 Related Work

Topological signals have been previously used in deep learning, in particular applied to auto-encoders ([Wang *et al.*, 2020]) and generative adversarial networks ([Moor *et al.*, 2020]). Topological persistence has also been used in parallel to deep learning based image classifiers ([Dey *et al.*, 2017]). In [Hu *et al.*, 2019], topology preserving loss function has incorporated into deep neural networks for image segmentation of medical images.

The novelty of our method is twofold. Firstly, we introduce the coupled architecture which is offers flexibility in terms of how the new topological features can affect the learning process of the convolutional network. Secondly, our method can make use of the topological losses to introduce both topological priors and posteriors during learning. To our knowledge this is the first time topological features have been applied to perception related learning tasks in driving scenarios.

2 Brief Overview of Topological Data Analysis

In this section we give a brief review of the fundamental ideas from Topological data analysis, and in particular the persistent homology algorithm. For references see ([Carlsson, 2009], [Chazal and Michel, 2017], [Carlsson and Zomordian, 2005]) Topological data analysis (TDA) emerged in an attempt to import invariants defined in classical algebraic topology, e.g. homology, to the discrete world of datasets. Homology is a machine which when fed a topological space (often modelled as a simplicial complex) measures various topologically invariant properties of the space like connectedness or presence of higher dimensional loops, in terms of the rank of the algebraic object it generates. The broad problem TDA attempts to address is this: it it possible to recover the topology of the underlying space which the data has been sampled from? This requires generalizing the homology algorithm from spaces to datasets. However naively applying homology to a dataset (which is a finite metric space) is uninteresting, as the dataset carries no connectivity information. Persistent homology views the dataset at different scales evolving through time and models it as a nested topological space, on which ordinary homology is applied. The resulting object, called a persistence diagram, contains topological information at various scales, the more salient ones are the ones which persist more. The bottleneck distance between persistence diagrams makes possible to do analysis over the space of such diagrams. Real valued loss function defined on the persistence diagrams allows us to run optimization over the space of the input data.

2.1 Persistent Homology overview

Persistence diagrams

Persistent homology (PH) is an algorithm to encode the evolution of homology groups of nested topological space across the scales. The nested topological space is modelled as a *filtered simplicial complex*. The PH algorithm extends the algorithm for computing ordinary *simplicial homology* of a simplicial complex to that of a filtered simplicial complex. The homology in dimension k of a simplicial complex X is an abelian group $H_k(X)$. The rank of $H_k(X)$ is also known as the k-th Betti number β_k and represents the number of k-dimensional holes in X. The PH in dimension k of a filtered simplicial complex $X = \{\phi = X_0 \subset X_1 \subset ... \subset X_i \subset\}$ is a sequential diagram of abelian groups $PH_k(X) = H_k(X_0) \to H_k(X_1) \to \dots H_k(X_n) \to \dots$. The persistence diagram PD_k is a way to encode the birth and death of every homology cycle that appears in the diagram. The longer a homology cycle survives the more persistent it is. The full information can be represented as a multi-set of pairs (b, d) where b and b are the birth and death filtration indices of a homology class. We denote $PD_k(X) = \{(b_i, d_i)\}_{i\in I_k}$. We order the indexing of points by decreasing lifetimes i.e. $i < j$ if $d_i - b_i \geq d_j - b_j$. Let's denote the space of persistence diagrams by $\mathcal{PD}$.

Let $\mathcal{PD}_k$ be the subspace of k-dimensional persistence diagrams. The space $\mathcal{PD}_k$ has a a metric defined by the bottleneck distance. For $D_1, D_2 \in \mathcal{PD}_k$,

$$\mathbb{B}(D_1, D_2) = \inf_m \max_{(p,q)\in m} \|p - q\|_\infty \tag{1}$$

where m is the set of all possible pairwise matchings of the points in D_1 and D_2

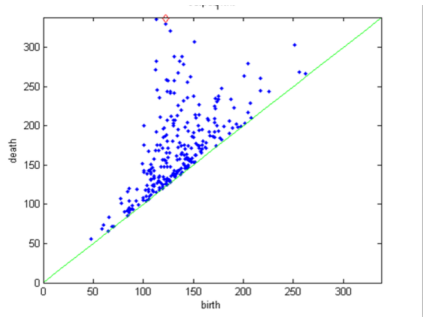

Figure 1: A Persistence Diagram representation obtained by plotting homology cycles by their birth-death values on the co-ordinate plane. The lifetime of a cycle is given by the vertical distance from the diagonal.

Functions on the space of persistence diagrams

The space of persistence diagrams is however not amenable to statistical analysis or even calculus, as they don't even form a vector space. There has been some alternative formulations of persistence diagrams, like persistence landscapes ([Bubenik, 2015]), to introduce additive structures on them. In order to get around this problem, we make use of a real-valued cost function on the space of persistence diagrams ([Gabrielsson *et al.*, 2020]) $\mathcal{E}(p, q, i_0; \texttt{dim}) : \mathcal{PD}_{\texttt{dim}} \to \mathbb{R}$

$$\mathcal{E}(p, q, i_0; \texttt{dim})(D) = \sum_{(b_i,d_i)\in D} |d_i - b_i|^p \left(\frac{d_i + b_i}{2}\right)^q \tag{2}$$

The parameter p and q define a polynomial function on the points (b_i, d_i) of a diagram D in $\mathcal{PD}_{\texttt{dim}}$. The output is the sum of lifetimes of all points in the diagram D after skipping the first i_0 points. The map $\mathcal{E}(p, q, i_0, \texttt{dim})$ is continuous given the topology induced by the bottleneck distance on $\mathcal{PD}_{\texttt{dim}}$.

In this paper we also consider the following special cases of the functional $\mathcal{E}$,

$$\mathcal{S}(\texttt{dim}) := \mathcal{E}(1, 0, 0; \texttt{dim}) \tag{3}$$

$$\mathcal{PS}(\texttt{dim}, \texttt{skip}) := \mathcal{E}(1, 0, \texttt{skip}; \texttt{dim}) \tag{4}$$

and,

$$\mathcal{T}(\texttt{dim}, \texttt{k}) = [|d_i - b_i| : 0 \leq i \leq \texttt{k}] \tag{5}$$

They stand for the functions `SumBarcodeLengths`, `PartialSumBarcodeLengths` and `TopKBarcodeLengths` defined in ([Gabrielsson and Nelson, 2019]).

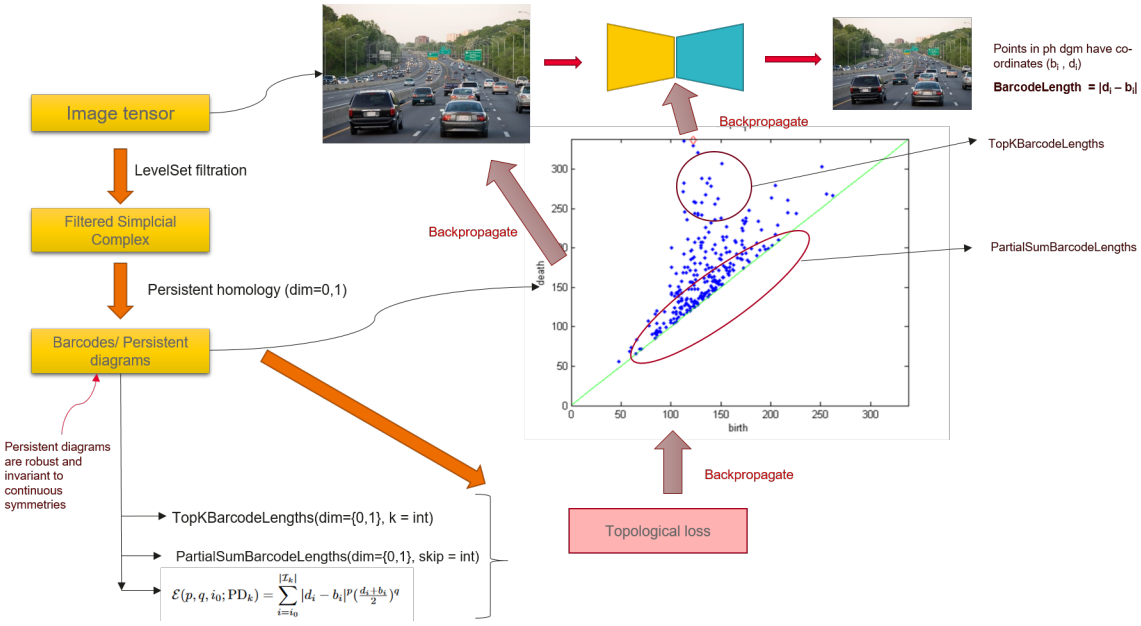

Figure 2: Topological features map pipeline: persistent homology is applied to the natural level-set filtration on image tensors. Composed with differentiable functionals on the space of persistence diagrams, this produces topology feature maps from space of image tensors to tensors. Differentiability allows for back-propagation to optimize either in image space of the weight space of a neural network.

The function $\mathcal{S}(\texttt{dim})$ sums up the lifetimes of all the homology cycles in dimension `dim`. The function $\mathcal{PS}(\texttt{dim}, \texttt{skip})$ sums up the lifetimes of all homology cycles in dimension `dim` skipping the first `skip` ones. The function $\mathcal{T}(\texttt{dim}, \texttt{k})$ returns a vector of length `k` whose components are the lifetimes of the first `k` homology cycles in dimension `dim`.

Differentiation

Given a filtered simplicial complex, the authors of ([Gabrielsson *et al.*, 2020]) introduce a method for computing the gradient of a functional of a persistence diagram $\mathcal{E}(p, q, i_0; \texttt{dim})$. This is done by observing that every birth-death pair can be mapped to the cell that respectively created and destroyed the homology class, defining an inverse map from the persistent diagram to pair of simplices.

3 Topological features from images

3.1 Sublevel set filtration

Given a real valued continuous function on a topological space $f : M \to \mathbb{R}$, we define the *sublevel set filtration* on M by increasing the parameter α, with $M_\alpha = f^{-1}(-\infty, \alpha]$. Let us denote the k-th persistent homology of the level-set filtration associated where (M, f) by $PD_k(f)$.

Consider a simplicial complex K with vertex set V and a function $f : V \to \mathbb{R}$. The function f can be extended to all simplices of K by $f([v_0, \ldots, v_k]) = \max\{f(v_i) : i = 1, \ldots, k\}$ for any simplex $\sigma = [v_1, \ldots, v_k] \in K$ and the family of sub-complexes $K_r = \{\sigma \in K : f(\sigma) \leq r\}$ defines a filtration called the sublevel filtration of f.

3.2 From Images to persistence diagrams

An image can be seen as a real valued function $f : \mathbb{I} \to [-1, 1]$ on the space $\mathbb{I} \simeq [0, 1]^2$. The graph of this function is a surface embedded in $\mathbb{I} \times [-1, 1]$. The level sets of this graph at various heights $F_r = f^{-1}(-1, r]$ produces a filtration F_r on $\mathbb{I}$. The persistent homology of this filtration is a way of summarizing of the evolving topologies of the level sets. We need to formulate this in the discrete language of tensors and simplicial complexes.

A gray-scale image can be represented as 2-d tensor. For an image X of dimension $n \times m$ the tensor is a function $X : [1..n] \times [1..m] \to \mathbb{R}$ Let $V = [1..n] \times [1..m]$ be the vertex set of a 2-dimensional simplicial complex homeomorphic to $\mathbb{I}$. Call this complex $\mathbb{V}$ and the lifted function $\tilde{X} : \mathbb{V} \to \mathbb{R}$. The sublevel filtration of X is the filtered simplicial complex $X_r = \{\sigma \in \mathbb{V} : X(\sigma) \leq r\}$. Let $\mathcal{I}$ denote the space image tensors. In our case $\mathcal{I}$ is a subspace of $\mathbb{R}^{n \times m}$.

$$\mathcal{I} \to \text{filtered simplicial complexes}$$

$$X \mapsto \{X_r\}_{r \in I} \tag{6}$$

The `dim`-dimensional persistent homology applied to the filtered simplicial complex $\{X_r\}_{r \in R}$ produces a persistent diagram $PD_{\texttt{dim}}(X) \in \mathcal{PD}_{\texttt{dim}}$.

$$\mathcal{I} \to \mathcal{PD}_{\texttt{dim}}$$

$$X \mapsto PD_{\texttt{dim}}(\{X_i\}) \tag{7}$$

3.3 Topological feature maps

Given any functional tensor valued functional $\mathcal{F}$ on $\mathcal{PD}$, we can compose to form topological feature maps as follows.

$$\widetilde{\mathcal{F}} : \mathcal{I} \to \text{tensors}$$

$$X \mapsto \mathcal{F}(PD_{\texttt{dim}}(\{X_i\}_{i\in I})) \tag{8}$$

The result is a topological feature map called $\widetilde{\mathcal{F}}$. In this paper our working examples of topological feature maps are $\widetilde{\mathcal{S}}(\texttt{dim})$, $\widetilde{\mathcal{PS}}(\texttt{dim}, \texttt{skip})$ and $\widetilde{\mathcal{T}}(\texttt{dim}, \texttt{k})$. For example the map $\widetilde{\mathcal{E}}(1, 0, 1; 0)$ is the sum of lifetimes of all homology 0-cycles skipping the first one.

3.4 Invariance and stability properties of topological features

Invariance

Simplicial homology is a homotopy invariant. Given a homotopy equivalence $f : X \to Y$ of simplicial complexes, the induces map of homology $f_* : H_n(X) \to H_n(Y)$ is an isomorphism. The notion of homotopy equivalence of filtered simplicial complexes is given by a commutative diagram of simplicial homotopy equivalences.

$$\begin{array}{ccccccc} X_0 & \xrightarrow{\subset} & X_1 & \xrightarrow{\subset} & X_2 & \xrightarrow{\subset} & \cdots \\ \simeq\downarrow f_0 & & \simeq\downarrow f_1 & & \simeq\downarrow f_2 & & \\ Y_0 & \xrightarrow{\subset} & Y_1 & \xrightarrow{\subset} & Y_2 & \xrightarrow{\subset} & \cdots \end{array} \tag{9}$$

Let us define two images to be topologically equivalent if there is a homotopy equivalence between the corresponding sublevel set filtrations. This is the case for example if $X(a, b) = Y(\phi(a, b))$ for some Euclidean transformation ϕ on I^2. This includes translational and rotational symmetries of the image. In general a topological symmetry between images is not necessarily of this form however.

Stability

Persistent homology satisfies a Lipschitz stability property in the following sense. Let $f, g : X \to \mathbb{R}$ be two real-valued functions defined on a topological space M that are q-tame. Then for any dimension k,

$$\mathbb{B}(PD_k(f), PD_k(g)) \leq \|f - g\|_\infty = \sup_{x\in M} |f(x) - g(x)| \tag{10}$$

Filtration of finite simplicial complexes are always tame. As a consequence, given any two images tensors $X, Y : \mathbb{I} \to [-1, 1]$, the associated persistence diagrams in any dimension k satisfies the Lipschitz stability property

$$\mathbb{B}(PD_k(X), PD_k(Y)) \leq \|X - Y\|_\infty \tag{11}$$

where $||X - Y||_\infty$ can be interpreted as the L_∞ distance between the unrolled tensors obtained from the tensors X and Y.

This Lipschitz stability property means persistent homology is robust to perturbations in the image space.

3.5 Back-propagation through topological feature maps

The space of persistence diagrams forms a continuous space, unlike the classical betti numbers. The map $PD_k : \mathcal{I} \to \mathcal{PD}$ is a continuous map. Therefore continuous morphisms in the image space, i.e. continuously transforming one image into another image will result in a continuous morphism of one persistence diagram into another. Composition with an continuous function into some additive and continuous space $\mathcal{F} : \mathcal{PD} \to \mathcal{A}$ could result in a differentiable topology feature map. The gradient of this map could used to continuously transform an image to optimise the topological feature map.

The maps $\widetilde{\mathcal{F}}$ are differentiable and the derivative has be computed by ([Gabrielsson *et al.*, 2020]) using an inverse map from points in the persistence diagram to pairs of simplices in the filtered simplicial complex responsible for the birth and death of the homology cycle. This also depends on a total ordering of all the simplices, and is not always possible due to simultaneous appearance of simplices. There are approaches to replace the total order with a strict order. Hence the gradient $\frac{\partial \widetilde{\mathcal{F}}}{\partial \sigma}$ is depends on a strict ordering of the σ's.

With this differentiable mechanism in place we can optimize in the the space of images $\mathcal{I}$ to maximize of minimize any desired topological feature map. In the examples in Figure 3 the input image X has been continuously morphed to find the closest image $\overline{X}$ which minimizes $\widetilde{\mathcal{E}}(1, 0, 10; 0)$ and $\widetilde{\mathcal{E}}(1, 0, 10; 1)$ (in top figure) and maximizes $\sum(\widetilde{\mathcal{T}}(\texttt{dim} = (0, 1), \texttt{k} = 20))$ and $\sum(\widetilde{\mathcal{T}}(\texttt{dim} = (0, 1), \texttt{k} = 20))$ (in bottom figure).

In notation,

$$\overline{X} = \text{argmin}_{X\in\mathcal{I}} \widetilde{\mathcal{E}}(1, 0, \texttt{skip}; \texttt{dim})(X) \tag{12}$$

$$\overline{X} = \text{argmax}_{X\in\mathcal{I}} \left\| \widetilde{\mathcal{T}}(\texttt{dim}, \texttt{k})(X) \right\|_1 \tag{13}$$

where $\texttt{dim} = \{0, 1\}$, $\texttt{skip} = 10$ and $\texttt{k} = 20$.

The effect of the first operation is a *topological smoothing* by moving the points in persistence diagram PD_0, except for the first 10, closer to the diagonal. This results in killing the homology cycles that are not too significant in terms of lifetimes. The effect of the second operation is a *topological sharpening* by pushing the top $\texttt{k}$ points in the persistence diagrams further away from the diagonal. This results in increasing the lifetimes of the the top $\texttt{k}$ homology cycles.

In the following section we apply this to a neural network optimizer. Given an neural network N_θ applied to an image tensor, the aim is to optimize the pair N_θ to maximize or minimize the topological feature maps. For example,

$$\overline{\theta} = \text{argmin}_\theta \mathbb{E}_X \widetilde{\mathcal{E}}(p, q, i_0; \texttt{dim})(N_\theta(X)) \tag{14}$$

4 Combining topology feature maps with Convolutional networks

The optimization task in Figure 3 can be alternative expressed as a optimizing a neural network. Consider

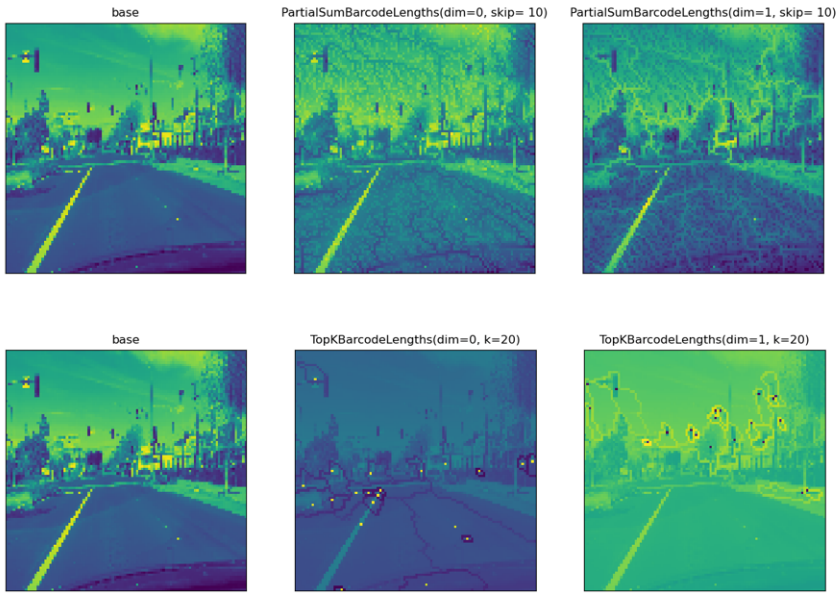

Figure 3: Top: Visualizing `PartialSumBarcodeLengths` using gradient-descent; Bottom: visualizing `TopKBarcodeLengths` using gradient-ascent

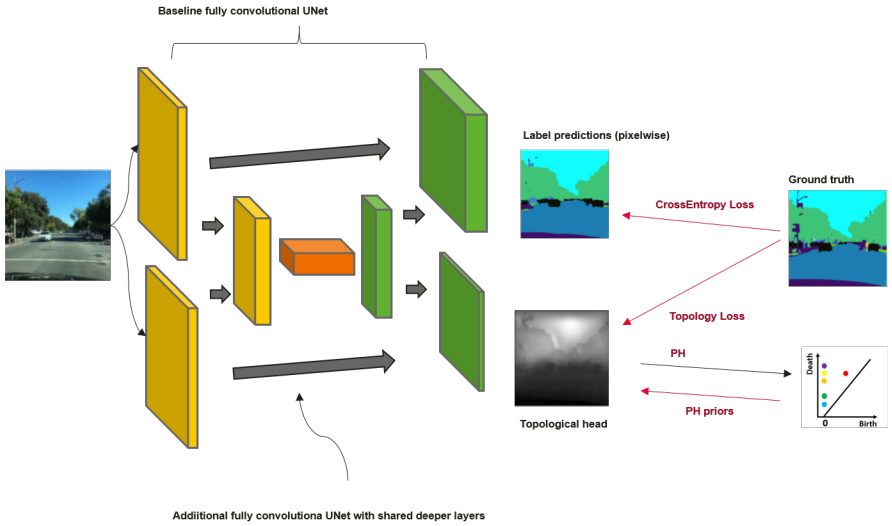

Figure 4: Topological Coupled-UNet architecture

an UNet architecture ([Ronneberger *et al.*, 2015]) with fully convolutional ReLU encoder E_θ and decoder D_ψ. Further suppose the encoder and decoder has layers

$$E_\theta = E^1_{\theta_n} \circ \ldots \circ E^n_{\theta_1}$$

and

$$D_\theta = D^1_{\psi_n} \circ \ldots \circ D^n_{\psi_1}.$$

The UNet architecture uses skip augmentations

$$E^k \to D^k \ (1 \leq k \leq n)$$

The topological feature map can be augmented serially at the end of the decoder network. The training process aims to solve the following optimization problem.

$$\overline{\theta}, \overline{\psi} = \text{argmin}_{\theta,\psi} \mathbb{E}_X \widetilde{\mathcal{E}}(1, 0, \texttt{skip}; \texttt{dim})(D_\psi \circ E_\theta(X)) \tag{15}$$

$$\overline{\theta}, \overline{\psi} = \text{argmax}_{\theta,\psi} \mathbb{E}_X \widetilde{\mathcal{T}}(\texttt{dim}, \texttt{k})(D_\psi \circ E_\theta(X)) \tag{16}$$

4.1 Coupled UNet architecture

In this section we define the coupled UNet architecture. We apply this to semantic segmentation. Semantic segments have very distinct topological properties. We wish to combine the UNet in the previous section trained to optimize certain topological features characteristic of semantic segments with a standard UNet used for semantic segments. The two UNets are combined in parallel with shared weights in the deeper dimensions.

Let us continue with the same notation for a UNet represented by fully-convolutional ReLU encoder-decoder pair (E_θ, D_ψ). Let $U_1 = (E_{\theta^1}, D_{\psi^1})$ and $U_2 = (E_{\theta^2}, D_{\psi^2})$ be two structurally identical UNets. The weights are shared in the middle. See Figure 4.

$$\theta^1_k = \theta^2_k \ (1 \leq k \leq K), \ K < n \tag{17}$$

$$\psi^1_k = \psi^2_k \ (1 \leq k \leq K), \ K < n \tag{18}$$

The training goal:

$$\overline{\theta^1}, \overline{\psi^1}, \overline{\theta^2}, \overline{\psi^2} = \text{argmin}_{\theta^1,\psi^1,\theta^2,\psi^2}(\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4) \tag{19}$$

where,

$$\mathcal{L}_1 = \mathbb{E}_X \texttt{CrossEntropy}(D_{\psi^1} \circ E_{\theta^1}(X), X) \tag{20}$$

$$\mathcal{L}_2 = \mathbb{E}_X \left\| (\widetilde{\mathcal{T}}(\texttt{dim}, \texttt{k})(D_{\phi^1} \circ E_{\theta^1}(X)) - \widetilde{\mathcal{T}}(\texttt{dim}, \texttt{k})(X)) \right\|_2 \tag{21}$$

$$\mathcal{L}_3 = \mathbb{E}_X \widetilde{\mathcal{E}}(1, 0, \texttt{skip}; \texttt{dim})(D_{\psi^2} \circ E_{\theta^2}(X)) \tag{22}$$

$$\mathcal{L}_4 = -\mathbb{E}_X \left\| \widetilde{\mathcal{T}}(\texttt{dim}, \texttt{k})(D_{\psi^2} \circ E_{\theta^2}(X)) \right\|_2 \tag{23}$$

The variables K (number of layers of the coupled UNet with shared weights), `skip`, `t` and `dim` are hyperparameters.

5 Experiments

5.1 Model specifications

Our baseline is the UNet model defined in [Ronneberger *et al.*, 2015]. Our model has the same architectural pattern used in this paper, with changes made to the input/output image dimensions and number of output channels (depending of number of segmentation classes). We shall denote this model as UNET

The coupled UNet consists of two copies of the UNET with shared weights in the middle layers. The second copy of the UNET has a single output channel. The number of layers in the idle with shared weights is a hyperparameter. Define CUNET to the coupled UNet where all but the final layers have shared weights. Define CUNET-wc (weakly coupled) to be the coupled UNet in which 7 layers (about 70%) in the middle of the UNETs have shared weights.

The network architectures are implemented in PyTorch. The persistent homology functional differentiable computational block is implemented using the Topology Layer PyTorch library ([Gabrielsson and Nelson, 2019]). We trained our network on Nvidia Tesla P100 GPU, operating system Ubuntu 18.04.02 LTS.

5.2 Datasets and Evaluation metrics

The **BDD100k** (Berkeley Deep Drive) dataset ([Yu *et al.*, 2020]) consists of video and image data of urban street scenes from diverse locations in the United States. The database covers different weather conditions, including sunny, overcast, and rainy, as well as different times of the day. For training our models we using the training dataset of 7000 RGB images and test our results on the validation set of 1000 images. The resolution chosen for training is 256*256.

The **Cityscapes** dataset ([Cordts *et al.*, 2016]) consists of images of urban street scenes from 50 different cities in Germany, captured in daytime and good/medium weather conditions. The annotations consist of dense semantic segmentation features with 30 class labels. We have trained our models on the training set consisting of 2975 images from all cities combined. The tests are carried out on the validation set consisting of 500 images. The resolution chosen for training is 256*256.

The **ACDC** (Adverse Conditions Dataset with Correspondences) dataset ([Sakaridis *et al.*, 2021]) consists of urban street scenes, highways and rural areas from Switzerland. The main focus is on adversarial visual conditions. The dataset consists of image-level correspondences between adverse-condition and normal -condition recordings via matching perspective GPS data. There are 4006 adverse-condition images with fine pixel-level annotations split into 100 foggy, 1006 nighttime, 1000 rainy and 1000 snowy images. The classes labels are identical to those of the Cityscapes dataset. We test our Cityscapes trained models on the ACDC dataset to demonstrate the robustness of the Coupled UNet on

adversarial weather images compared with the baseline models.

We use the metrics *Pixel-level Accuracy* (Accuracy) and *Intersection-over-Union* (IOU) to get quantitative comparisons of our models. The IOUs are computed by taking the mean of the IOUs over all classes.

5.3 Qualitative results

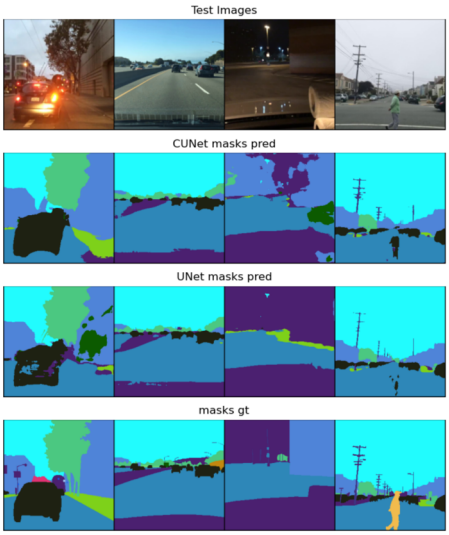

Figure 5: Semantic segmentation using topological Coupled UNet on BDD100 dataset. Topological loss of the CUNET forces simply connected segments.

The figure 5 shows output semantic segmentation of UNET and CUNET trained on **BDD100k** training dataset and evaluated on **BDD100k** validation dataset. The figure 6 shows output semantic segmentation of UNET, CUNET and CUNET-wc trained on **Cityscapes** training data and evaluated on **Cityscapes** validation data.

Both examples show marked improvement in segmentation quality in the CUNets compared to the baseline UNETs. The precise nature of this improvement is the preservation of the topological properties of the class segments. Examples of topological properties of segments are connected components and number of 1-dimensional loops. The topological loss during the training process of the CUNET helps to learn these topological features on top of the usual convolutional features. It is also worth noting that segmentation quality in the weakly coupled CUNET is marginally better than the strongly coupled CUNET

5.4 Quantitative results

In this section we present the quantitative results of our experiments. The scores are based on pixel-wise accuracy and Intersection-over-Union. Table 1 show comparison of scores between UNET and CUNET on the

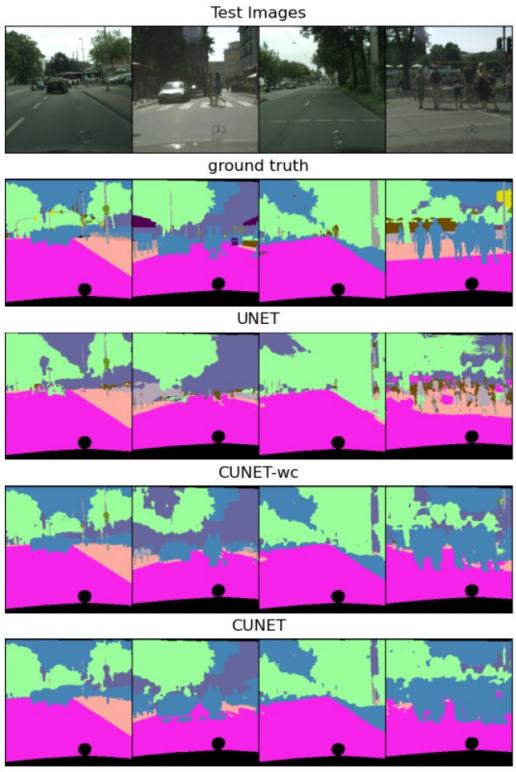

Figure 6: Semantic segmentation using strongly and weakly Coupled UNet on the Cityscapes dataset.

Dataset	Method	Accuracy	IOU
BDD100k	UNET	0.9843	0.6731
	CUNET	**0.9865**	**0.7034**
Cityscapes	UNET	0.9823	0.7839
	CUNET	**0.9909**	**0.8387**

Table 1: Quantitative results for the baseline UNET and the Topological Coupled UNET models on BDD100k and Cityscapes datasets

BDD100K and **Cityscapes** datasets. Our model outperforms the baseline in both cases.

In table 2 we produce results comparing performance of the CUNET with the UNET on the **ACDC dataset**. Here both models were trained on the Cityscapes (clean weather) training data and evaluated on the **ACDC** (adverse weather) validation data. Our model CUNET is consistently better than the baseline UNET except for the nighttime scenes. These results demonstrate the capacity of the topological coupled network to perform better than the baseline when distribution shift via adverse weather is introduced during inference. This is due to the fact the global topological features learned during training are more robust to distribution shifts than convolutional features alone.

In table 3 we produce the results comparing the performance of the CUNET with a weakly-coupled

Dataset	Method	Accuracy	IOU
Cityscapes	UNET	0.9878	0.7456
	CUNET	**0.9934**	**0.7920**
ACDC(fog)	UNET	0.9361	0.6740
	CUNET	**0.9575**	**0.7387**
ACDC(rain)	UNET	0.9377	0.6822
	CUNET	**0.9609**	**0.7403**
ACDC(snow)	UNET	0.9364	0.6632
	CUNET	**0.9571**	**0.7321**
ACDC(night)	UNET	**0.9418**	**0.7210**
	CUNET	0.9259	0.7188

Table 2: Quantitative results for the baseline UNET and the Topological Coupled UNET models on ACDC adversarial weather conditions dataset

Dataset	Method	Accuracy	IOU
Cityscapes	CUNET-wc	**0.9935**	**0.8031**
	CUNET	0.9934	0.7920
ACDC(fog)	CUNET-wc	0.9598	0.7333
	CUNET	**0.9575**	**0.7387**
ACDC(rain)	CUNET-wc	0.9508	**0.7496**
	CUNET	**0.9609**	0.7403
ACDC(snow)	CUNET-wc	0.9384	0.7263
	CUNET	**0.9571**	**0.7321**
ACDC(night)	CUNET-wc	**0.9261**	**0.7192**
	CUNET	0.9259	0.7188

Table 3: Comparison of performance of weakly and strongly coupled UNETs on Cityscapes and ACDC dataset

CUNET-wc. Both models are trained on the Cityscapes. The results show improvement in the weakly-coupled architecture when evaluated on the Cityscapes validation dataset. The results are however mixed and inconclusive when adverse weather distribution shift is introduced.

6 Summary and Conclusions

Persistent homology captures global topological information. Their definitions are based on algebraic-topological invariants called homology adapted to point-cloud data. RGB image tensors can be represented as scalar functions on the domain and the natural filtration can be used to define persistent homology. This is captured by persistent homology functionals, and not detected by standard convolutional filters. These topological features are furthermore invariant under continuous transformations. Lipschitz stability property satisfied by persistent homology also means it is robust under perturbations in the image space.

Semantic segments of images are have very well defined topological features and serve as a good test case for experimenting with these new features. Our experiments show that learning topological features on top of a convolutional features improves performance of current semantic segmentation models on urban road scenes. We also empirically show that the robustness of the topological features help the new model perform better when distribution shift is introduced during inference via adverse weather conditions.

References

[Bubenik, 2015] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(3):77–102, 2015.

[Carlsson and Zomordian, 2005] Gunnar Carlsson and Afra Zomordian. Computing persistent homology. *Discrete and Computational Topology*, 33:249–274, 2005.

[Carlsson, 2009] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, April 2009.

[Chazal and Michel, 2017] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Journal de la Société Française de Statistique 2017*, 2017.

[Cordts *et al.*, 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Dey *et al.*, 2017] Tamal Krishna Dey, Sayan Mandal, and William Varcho. Improved Image Classification using Topological Persistence. In Matthias Hullin, Reinhard Klein, Thomas Schultz, and Angela Yao, editors, *Vision, Modeling & Visualization*. The Eurographics Association, 2017.

[Gabrielsson and Nelson, 2019] Rickard Brüel Gabrielsson and Bradley J. Nelson. TopologyLayer: A Topology Layer for Machine Learning : Persistent Homology + Features for PyTorch, 2019.

[Gabrielsson *et al.*, 2020] Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1553–1563. PMLR, 26–28 Aug 2020.

[Hu *et al.*, 2019] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Moor *et al.*, 2020] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In Hal Daumé III and Aarti Singh, editors,

Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 7045–7054. PMLR, 13–18 Jul 2020.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[Sakaridis *et al.*, 2021] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. *ArXiv e-prints*, 2021.

[Wang *et al.*, 2020] Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. Topogan: A topology-aware generative adversarial network. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 118–136, Cham, 2020. Springer International Publishing.

[Yu *et al.*, 2020] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.