FINAL MILESTONE REPORT
TOPOLOGICAL CONTROL FOR TIME AWARE MACHINE INTELLIGENCE

Report Date: July 30, 2022

Contract No: HR00112190040

Contract PoP: 12/14/2020 – 06/14/2022

Program Name: Time-Aware Machine Intelligence (TAMI)

Reporting Period: 4/30/2022 - 6/14/2022

Prepared by:

Bradley J. Nelson
University of Chicago
5801 South Ellis Avenue, Chicago, IL 60637
Phone: 773-702-4263

Distribution List

Agreements Officer (AO):
Lydia Richards
Phone: (703) 526-4129
Email: Lydia.Richards@darpa.mil

DARPA Program Manager (PM)
Lt Col David Lewis
Email: charlton.lewis@darpa.mil

Agreements Officer's Representative (AOR)
Dr. MaryAnne Fields
Tel: (919) 549-4350
Email: mary.a.fields22.civ@mail.mil

Support Staff
Hunter Gabbard: hunter.gabbard.ctr@darpa.mil
Lee Pele: Lee.Pele.ctr@darpa.mil
Olivia Large: Olivia.Large.ctr@darpa.mil

DARPA Closeout Team:
Email: CMO_closeout@darpa.mil

Defense Technical Information Center
Attn: DTIC-BCS
8725 John J. Kingman Road, Suite 0944
Fort Belvoir, VA 22060-0944

OVERVIEW

In this final milestone report, we detail our work to address task 8 in our program:

8) Report on refinements to theory, methodology, and analysis tools. Final report demonstrating understanding of neural networks by topological means as part of a time-aware machine intelligence system. Compare to SOAand document contributions.

1. SUMMARY OF ACTIVITIES

In this reporting period, we have continued experiments outlined in our Phase 2 research plan. We have primarily focused on using topological constructions to understand how neural networks operate on video data, and in incorporating Lorentzian geometry into neural networks using equivariant layers. We also detail our work to improve computational performance persistent homology, which we use extensively in our topological investigations, and persistent path homology, which is a form of persistent homology which is particularly amenable to directional data such as causal graphs.

Proposed Milestone/Objective	Accomplishment
Final report demonstrating understanding of neural networks by topological means as part of a time-aware machine intelligence system.	This report
Software used to build topological control apparatus, neural networks, machine intelligence system.	Software uploaded to Okta

TABLE 1. Breakdown of task 8: Refer to Revised TDD for agreement No.: HR001121900040

2. DETAIL DESCRIPTIONS

2.1. **Lorentz-Equivariant Neural Networks.** Our first main line of work was to incorporate Lorentz equivariance into neural networks in order to preserve causal structure in the input data.

2.1.1. *Equivariant Neural Networks.* A feedforward neural network, assuming constant width, is a map $f\colon \mathbb{R}^n \to \mathbb{R}^n$ given by alternately composing matrices $A_i \in \mathbb{R}^{n\times n}$ with pointwise activations $\sigma_i(v) := \max(v, b_i)$ where $b_i \in \mathbb{R}^n$ and max is taken coordinatewise, i.e.,

$$f(v) = A_k\sigma_{b_{k-1}}A_{k-1}\cdots\sigma_{b_2}A_2\sigma_{b_1}A_1v.$$

A G-equivariant neural network requires that for some chosen matrix group $G \subseteq \mathbb{R}^{n\times n}$, we have $f(Xv) = Xf(v)$ for all $X \in G$ and $v \in \mathbb{R}^n$. Since

$$\begin{aligned} f(Xv) &= X(X^{-1}A_kX)(X^{-1}\sigma_{b_{k-1}}X)(X^{-1}A_{k-1}X)\cdots(X^{-1}A_2X)(X^{-1}\sigma_{b_1}X)(X^{-1}A_1X)v \\ &= XA_k'\sigma_{b_{k-1}}'A_{k-1}'\cdots\sigma_{b_2}'A_2'\sigma_{b_1}'A_1'v \end{aligned}$$

and the last expression equals $Xf(v)$ if we have

$$A_i' = X^{-1}A_iX = A_i, \quad \sigma_{b_i}' = X^{-1}\sigma_{b_i}X = \sigma_{b_i} \tag{1}$$

for all $i = 1, \ldots, k$, the conditions in (1) vastly narrow down the range of possible weights A_i and biases b_i, a huge computational saving in modern applications. Incorporating group representations allows us to extend this to variable-width neural networks. This refreshing interpretation of equivariant neural network is the focus of our invited contribution to the widely read "What-is" series in the Notices of the AMS [39].

2.1.2. *Lorentzian Geometry.* Lorentzian geometry is the geometry of space and time where information propagation has a speed limit (the speed of light in physics, but other speed limits may apply in different applications). The geometry is defined by a non-standard dot product

$$((x_0, t_0), (x_1, t_1)) = x_0^T x_1 - c^2 t_0^T t_1 \tag{2}$$

where c is the speed of light in physics, or otherwise a parameter to be chosen. This dot product defines a notion of (pseudo-)distance $d(a, b) = \sqrt{(a-b, a-b)}$.

In Lorentzian geometry, vectors are partitioned into three classes, producing a *causal structure* on space-time. If $(a, a) < 0$, then a is *time-like*, meaning there can be a causal relationship between the vector a and the origin. If $(a, a) > 0$, then a is *space-like* and there can be not causal relationship. If $(a, a) = 0$, then a is *light-like*. These classes are determined in part by the value of the parameter c in the inner-product.

2.1.3. *Lorentz Equivariance.* The Lorentz group is the set of transformations of an input space with Lorentzian geometry which preserve the indefinite inner product, and thus, causal structure. Define $I_{1,d-1} \in \mathbb{R}^{d\times d}$ to be the diagonal matrix with diagonal $(1, -1, -1, \ldots, -1)$. The Lorentz group is equivalent to the matrix group

$$SO(1, d-1) := \{G^T I_{1,d-1} G = I_{1,d-1}, G \in \mathbb{R}^{d\times d}\}.$$

In order to create networks that are "time aware", we have incorporated Lorentz equivariance into neural network models. This has been recently proposed in the context of particle physics [5]. In our work, we seek to use this on a wider variety of temporal data.

As an example, we can consider the problem of learning the solution of 1-dimensional wave equation:

$$u_{xx} = u_{tt}.$$

The solution of wave equation is equivariant with respect to the Lorentz group $\mathrm{SO}(1, 1)$. It should be noted that the equivariance does not hold for one equation. Quite the contrary, it is the space of the solutions of the wave equation that has the symmetry $SO(1, 1)$. To be more specifically, if we are give a solution $u(x, t)$ of the wave equation, then $u(g(x, t))$ is also a solution where g is any element in $\mathrm{SO}(1, 1)$.

For comparison, consider the structure of physics-informed (PI) neural network [52]. In a PI network, the loss function adds the constraints of the derivatives and has the extra term:

$$L_{physics-informed} = \sum \left(\frac{\partial^2}{\partial x^2} u_w(x_i, t_i) - \frac{\partial^2}{\partial t^2} u_w(x_i, t_i) \right)^2.$$

However, previous applications of PI only work for one specific problem at a time and requires requires much higher sampling rate (see the experimental details in [52]).

The data set is constructed as follows. For each sample, we randomly generate a solution of the wave equation

$$u(x, t) = \sum_{i=1}^{5} (a_n \sin nx + b_n \cos nx) \sin nt.$$

Then generate 101 points $\{(x_k, t_k)\}_{1\le k\le 101}$ in the domain $x, t \in [0, 1]$ together with the function value $\{u(x_k, t_k)\}_{1\le k\le 101}$. In each sample, the output consists of one function value $u(x_{101}, t_{101})$, while the input consists of the position of all 101 points and the function value of first 100 points. The training and testing data sets each have 10,000 samples.

We use the equivariant multilayer perception (EMLP) structure [22] which allows simple construction of Lorentz equivariant networks. A network with 5 layers and 16 width can achieves relatively good performance compared to linear regression (LR) and vanilla multilayer perception (MLP). As we can see in the results, it is possible to use Lorentz equivariant neural network to learn a complicated map using relatively small number of samples. It is also possible to combine physics-informed network with EMLP structure. However, itthe training takes much longer time and does not demonstrate significant improvement in the loss.

	LR	MLP	EMLP
Train MSE	0.681	0.079	0.095
Test MSE	0.725	3.368	0.310

TABLE 2. Train and test mean square error (MSE) for different models, including linear regression (LR), multilayer perception (MLP), equivariant multilayer perception (EMLP).

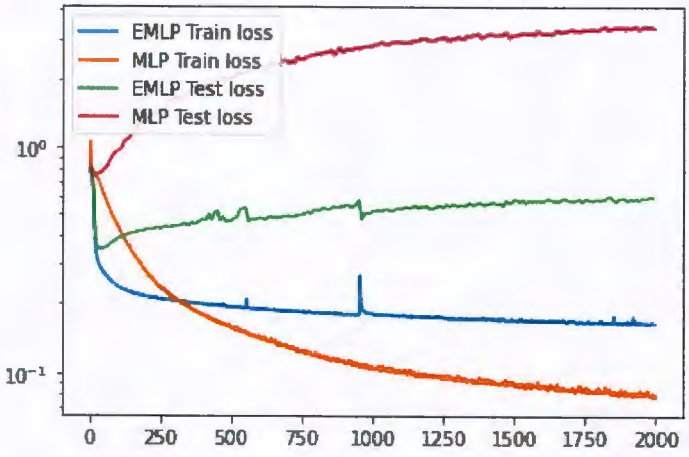


FIGURE 1. Train and test mean square error against number of epochs.

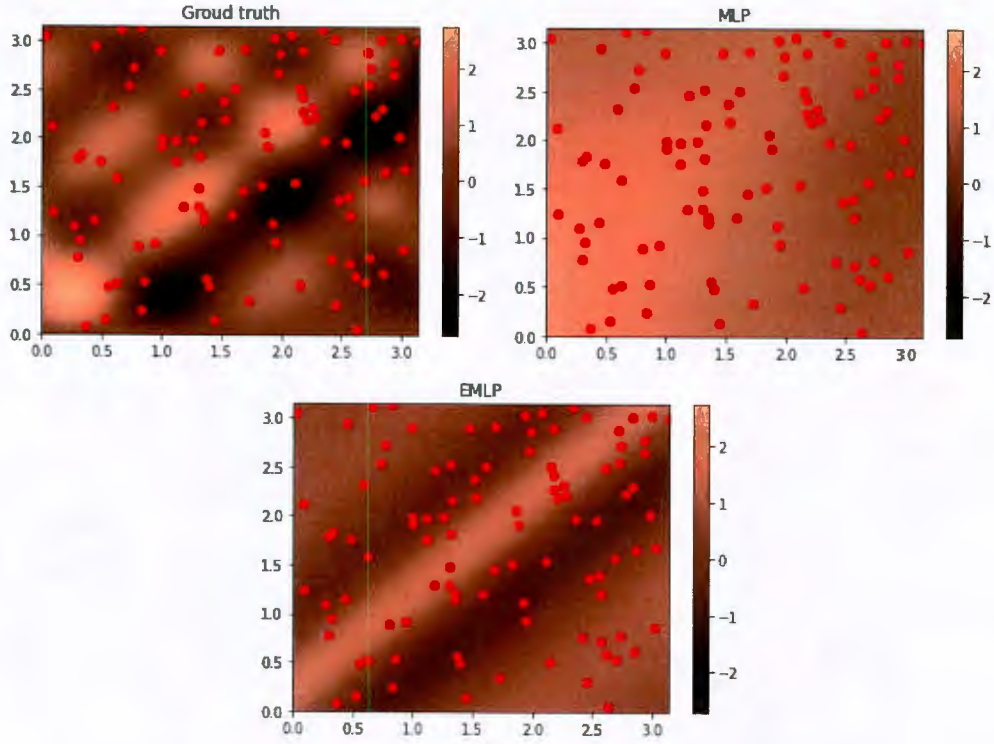


FIGURE 2. Examples of heatmaps of learned function against sampled points.

2.2. **Temporal Prediction Tasks in Computer Vision.** Our second main line of work is to use topology to investigate how information propagates through neural networks trained on video data. This builds on our prior work understanding and regularizing topology in dense prediction tasks for static images [23]

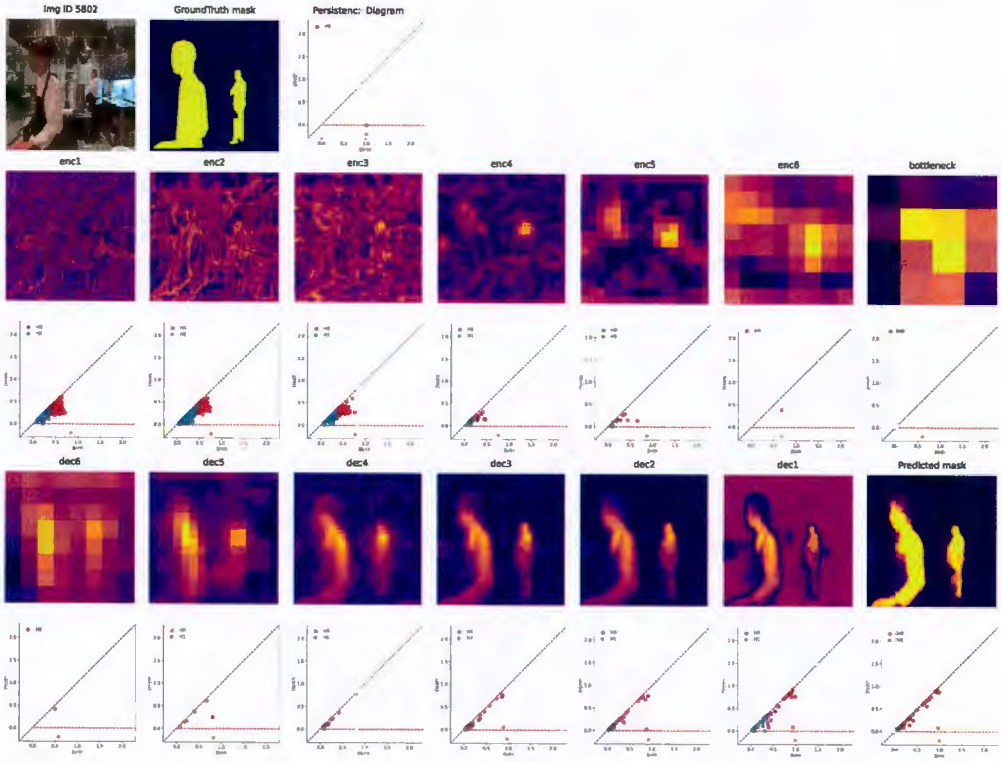


FIGURE 3. **Visualizations and Persistence Diagrams of Internal Activations.** The first row shows the input image, the ground truth segmentation mask, and the mask's persistence diagram. The 2nd and 4th row show the visualizations of the magnitude of internal activations. On the 3rd and 5th row, each figure shows the persistence diagrams of the corresponding internal activations above. Points further from the diagonal are more robust features. Both visualizations and diagrams show that the level-set topology can emerge as early as in the 4th encoder layer, and it remains nearly consistent in the decoder layers. No topological regularization is used on this example.

2.2.1. *Internal Activations.* First, we provide experimental results which demonstrate that topology can appear in the internal activations of trained neural networks. We train a convolutional neural network on binary semantic segmentation task which assigns values to each pixel indicating whether this pixel is part of a human torso or not.

Model architecture. We use the U-Net [54] architecture, which is a popular baseline for segmentation tasks. Our U-Net adopts 6 phases of encoders and 6 phases of decoders, where each block consists of two layers of convolutions, batch normalizations, and ReLU activations. Figure 3 shows the flow of the network, along with visualizations of each layer, and the network progresses as $\texttt{Input} \to \texttt{enc1} \to \cdots \to \texttt{enc6} \to \texttt{bottleneck} \to \texttt{dec6} \to \cdots \to \texttt{dec1} \to \texttt{Output}$ with skip connections.

Dataset. We use the COCO dataset [40], specifically COCO-2014, which contains 83K training images and 41K validation images. From its provided semantic segmentation annotations, we make a subset where all images contain human annotations and further process them to only annotate pixels as human/non-human. This commonly used data set contains human objects and [65, 28] have recently studied bias in captions. However, our semantic segmentation task does not use these potentially problematic labels.

Training objectives and hyper-parameters. The training objective is the MSE loss

$$\mathcal{L}_{\text{MSE}}(y, \hat{y}) = \sum_{i=1}^{h} \sum_{j=1}^{w} (y_{i,j} - \hat{y}_{i,j})^2 \tag{3}$$

where h and w are the height and width of the image. We train the network for 100 epochs with initial learning rate of 0.01. We use SGD as the optimizer with learning rate decaying by half every 40 epochs.

2.2.2. *Topological Regularization.* The particular regularization we use is in the family of functionals based on algebraic functions of the birth-death pairs [1]. Specifically, we penalize all but the k longest birth-death pairs in dimension 0:

$$\mathcal{L}_{\text{Topology}}(\{b_i, d_i\}) = \sum_{i>k} |d_i - b_i|^2. \tag{4}$$

This encourages at most k local maxima in the function f over the image channels.

2.2.3. *Binary Semantic Segmentation.* We experimented using the same architecture as described above and trained the networks for 50 epochs with cosine learning rate schedule, but varying the choice of regularization. We regularize `dec4` with $k = 8$ to penalize more than 8 connected components on this intermediate layer. As shown in Figure 4, there is a improvement on `mIoU`, e.g., mean Intersection-over-Union accuracy, when we regularize the second decoder layer with the proposed topological regularizer. Also shown in Figure 4, there are also improvements on convergence speed with the assistance of the proposed regularization.

	mIoU (%)
No Regularization	51.64
Topological Regularization	**52.54**

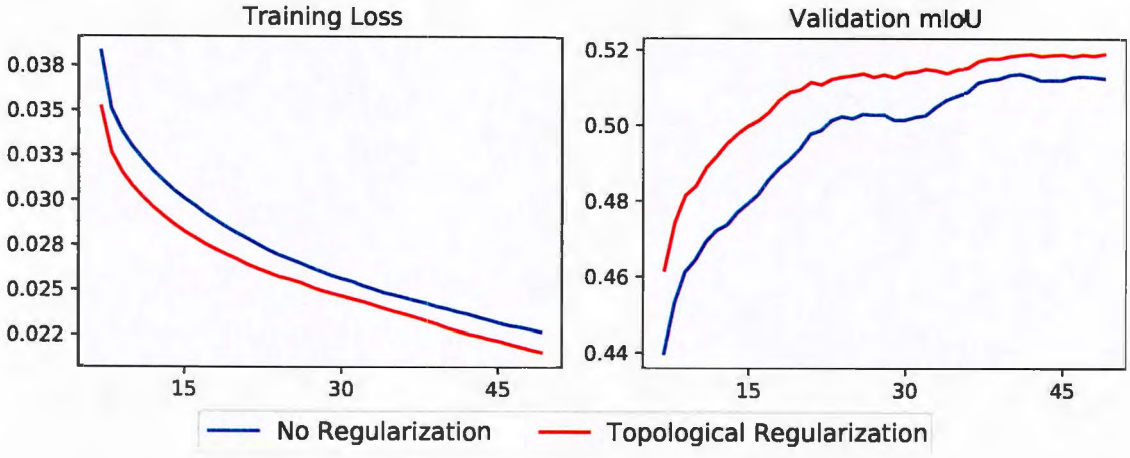


FIGURE 4. Convergence and Convergence Improvement by Topology Regularizer

2.2.4. *Monocular Depth Estimation.*
Training Objectives. Our proposed training objective is a weighted sum of three loss functions and two regularization terms:

$$\begin{aligned}\mathcal{L}(y, \hat{y}) = \lambda_{\text{d}} \cdot \mathcal{L}_{\text{depth}}(y, \hat{y}) + \lambda_{\text{g}} \cdot \mathcal{L}_{\text{gradient}}(y, \hat{y}) + \lambda_{\text{s}} \cdot \mathcal{L}_{\text{SSIM}}(y, \hat{y}) \\ + \lambda_{\text{tv}} \cdot \mathcal{L}_{\text{TotalVariation}}(\hat{h}^{(a)}) + \lambda_{\text{top}} \cdot \mathcal{L}_{\text{Topology}}(\hat{h}^{(b)})\end{aligned} \tag{5}$$

where y represents the ground-truth label of depth, $\hat{y}$ represents the predicted depth, $\hat{h}^{(a)}$ and $\hat{h}^{(b)}$ represents some intermediate layers of the network. Each loss term is defined as follows,
Depth Loss. We use the RMSE loss in log scale which empirically converges faster than L1 or L2 loss.

$$\mathcal{L}_{\text{depth}}(y, \hat{y}) = \sqrt{\sum_{i=1}^{h} \sum_{j=1}^{w} (\log y_{i,j} - \log \hat{y}_{i,j})^2} \tag{6}$$

Gradient Loss. The horizontal and vertical image gradients, $\nabla_{\parallel}$ and $\nabla_{\perp}$, are computed by a Sobel filter [56, 30]. This further helps align the edges of the ground-truth and the prediction.

$$\mathcal{L}_{\text{gradient}}(y, \hat{y}) = \sum_{i=1}^{h} \sum_{j=1}^{w} \left(|\nabla_{\perp} y_{i,j} - \nabla_{\perp} \hat{y}_{i,j}| + |\nabla_{\parallel} y_{i,j} - \nabla_{\parallel} \hat{y}_{i,j}| \right) \tag{7}$$

Structural Similarity Loss. This loss uses the Structure Similarity Index Measure (SSIM) [63] which is shown to be a good loss term for depth estimation tasks [26].

$$\mathcal{L}_{\text{SSIM}}(y, \hat{y}) = \frac{1 - \text{SSIM}(y, \hat{y})}{2} \tag{8}$$

Total Variation Regularization. It's often used in imaging tasks where the expected output is piece-wise constant. We apply this regularization term to the last layer $\hat{h}^{(a)}$ before the output layer.

$$\begin{aligned} \mathcal{L}_{\text{TotalVariation}}(\hat{h}^{(a)}) = & \sum_{k=1}^{c} \sum_{i=1}^{h-1} \sum_{j=1}^{w} \left(\hat{h}^{(a)}_{i+1,j,k} - \hat{h}^{(a)}_{i,j,k} \right)^2 \\ & + \sum_{k=1}^{c} \sum_{i=1}^{h} \sum_{j=1}^{w-1} \left(\hat{h}^{(a)}_{i,j+1,k} - \hat{h}^{(a)}_{i,j,k} \right)^2 \end{aligned} \tag{9}$$

Topological Regularization. We apply this regularizer to the second decoder layer $\hat{h}^{(b)}$. We first use the L2-norm projection to project this internal activation to $\tilde{h} \in \mathbb{R}^{h \times w}$ and compute its birth-death pairs (b_i, d_i) using super-level set filtration and persistent homology. Afterwards, we formulate the loss, given these birth-death pairs, through Equation (4), e.g., $\mathcal{L}_{\text{Topology}}(\{b_i, d_i\}) = \sum_{i>k} (d_i - b_i)^2$. For our application, we choose $k = 8$ to penalize internal activations to have more than 8 connected components or local extrema.

Experiment Setup and Results. We trained the model on DIODE [61] datasets with two backbone networks: U-Net and DenseDepth [2]. For U-Net, models are trained for 100 epochs with batch size 16, a cosine learning rate schedule, a momentum of 0.9 and a weight decay of 1e-4. For DenseDepth, to achieve the best performance, we use Adam optimizer [32] with initial learning rate 0.0001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Models are trained for 20 epochs with batch size 12 and a cosine learning rate schedule. For regularization settings, both models are regularized as such: the total variation regularization is enforced onto the last decoder layer and the topological regularization is enforced onto the second decoder layer.

Quantitative comparisons can be found in table 3 and fig. 5. Qualitative comparisons of internal layers is shown in fig. 6.

2.3. Dense Prediction for Video Data.

2.3.1. *Fully Convolutional Neural Network.* A *fully convolutional neural network* (FCN) [55] is a convolutional neural network without any fully connected layer. In other words, it is composed of convolutional layers and max pooling layers. Fully convolutional neural networks achieve good performance in many computer vision tasks, such as image segmentation, video semantic segmentation, and object tracking. Furthermore, FCNs are the backbone of many state of the art architectures used in computer vision. Usually, an FCN is used as an initial embedding of the input images, and is followed by other architectural components such as transformers. We want to study how the topology of an image changes as it goes through the convolutional layers in a FCN.

In figure 7, we show the activations and persistent diagrams of the picture of a bear, as it goes through the layers in a FCN. From this figure, we see that the topology of the image is simplified as it goes through the layers in FCN.

2.3.2. *Temporal Memory Attention Network (TMANet).* Results in the previous section motivate us to consider complicated architectures that use FCN as a backbone. In this section, we consider a state of the art model called *Temporal Memory Attention Network* (TMANet) [62], which is used in video semantic segmentation tasks. TMANet achieves new state of the art performance on Cityscapes and CamVid datasets.

	Level of Regularization	lower is better					higher is better		
		mae	rmse	abs rel	mae $\log_{10}$	rmse $\log_{10}$	δ_1	δ_2	δ_3
U-Net	None	4.2776	6.5386	0.4524	0.1706	0.2181	0.4336	0.6778	0.8128
	Total Variation	4.0952	6.3145	0.4311	0.1649	0.2103	0.4455	0.6915	0.8184
	Topology	4.0548	6.2168	**0.4206**	0.1651	0.2121	0 4388	0.6914	**0.8295**
	TV + Topology	**4.0138**	**6.2044**	0.4269	**0.1614**	**0.2069**	**0.4565**	**0.7020**	0.8232
DenseDepth	None	3.6554	5.9900	**0.3648**	0.1660	0.2452	0 5088	0.7481	0.8625
	Total Variation	3.5073	5.5763	0.3922	0.1427	0.1884	0 5151	0.7444	0.8665
	Topology	3.5857	5.7030	0.3921	0.1435	0.1887	0.5006	0.7418	0.8668
	TV + Topology	**3.4065**	**5.4196**	0.3908	**0.1395**	**0.1849**	**0.5197**	**0.7582**	**0.8745**

TABLE 3. **Quantitative Comparison.** UNet and DenseDepth Performance on DIODE. Each model is compared internally with varying regularization choices. Numbers in Red indicate the best score whereas in Blue shows the second best.

In video semantic segmentation tasks, there are two types of topology to consider: the 2D topology and the 3D topology. For 2D topology, we consider a fixed frame in a video. Then, we consider how the topology of that frame changes as it goes through the layers in TMANet. For 3D topology, we take ten consecutive frames in a video and consider these frames as a 3D tensor. The three dimensions in this tensor are height, weight, and time. Then we look at how the topology of this 3D tensor changes as it goes through TMANet. We generate persistent diagrams for both 2D and 3D topology of images in CamVid.

From figure 8 and figure 9, we see that both 2D topology and 3D topology are simplified as the images go through TMANet.

2.3.3. *Topological Regularization.* Results in the previous section motivate us to add topological regularization to TMANet. The idea is to explicitly force the network to learn a predictor that simplifies the 3D topology. Our goal is to show that this 3D topological regularization would improve generalization of TMANet. Currently, we modified the code for the original TMANet paper [62] to incorporate topological regularization. However, the computation on GPU is too much to run on our machine. We are currently modifying the code to let it run on CPU instead.

2.4. **Updating Persistent Homology.** Topological regularization can be a computational bottleneck in our work on images/video as well as in other deep learning problems [51]. We have developed schemes for accelerating these computations using updates. Our full paper is available in [41].

2.4.1. *Background and Motivation.* Most algorithms for computing persistent homology are based on computing a factorization of filtered boundary matrix D_q (meaning its rows and columns are arranged in the order of appearance of cells in the filtration):

$$D_q V_q = R_q, \tag{10}$$

where V_q is upper-triangular and R_q is *reduced*, which means that it has unique *low pivots*, i.e. the index of the last non-zero row of each column (if it exists) is unique. The original algorithm for finding such a matrix factorization takes worst-case time cubic in the number of simplices. This can be a bottleneck in problems that require the computation of persistent homology of many similar topological spaces. Examples include feature generation for data in machine learning tasks [10, 25, 9, 24] as well as in continuous optimization problems with persistent homology included in the objective [17, 6, 36, 7, 16, 31]. In these computationally intensive scenarios, we wish to be able to reuse computation, particularly factorization, to the largest extent possible.

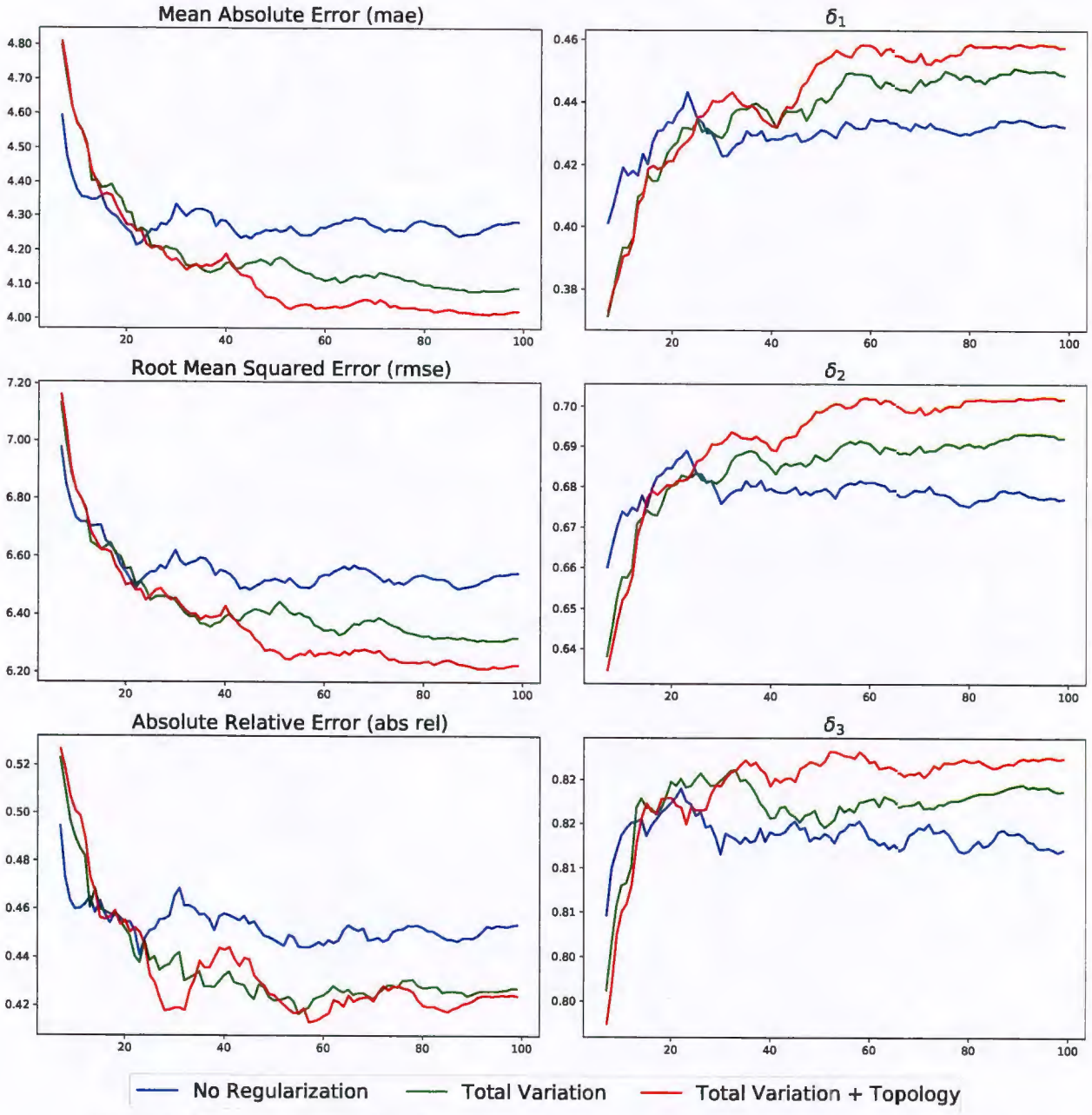
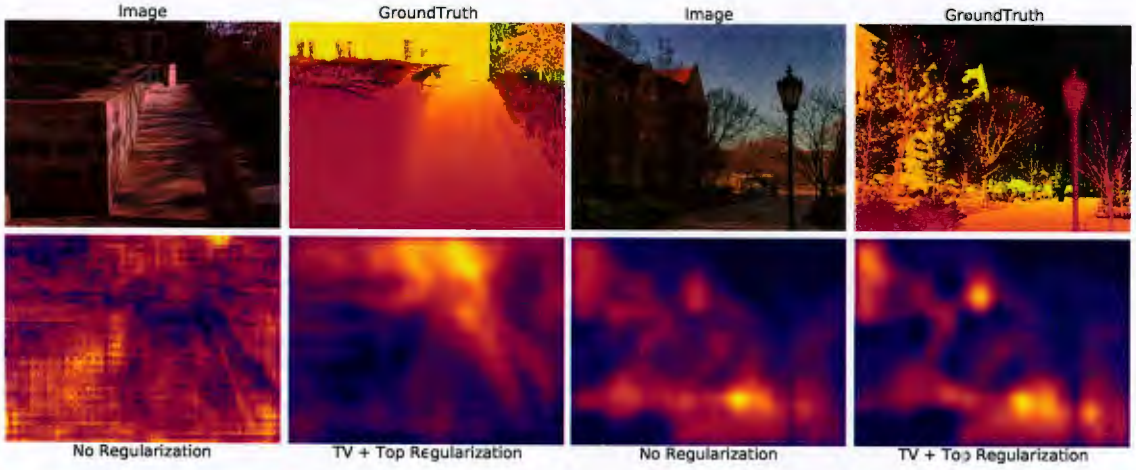


FIGURE 5. **Improvements on U-Net Convergence.** Blue lines (without regularization) demonstrates some degree of overfitting as loss increases and accuracy decreases after 20 epochs. Green lines (with total variation regularization) alleviates overfitting a bit, and Red lines (with both proposed regularizations) further overcomes overfitting and converges to a better optimum.

2.4.2. *Main Contribution.* We propose an approach to accelerate iterated persistent homology computations based on updating associated matrix factorizations. Our approach improves the update scheme of [20] for permutations by additionally handling addition and deletion of cells in a filtered topological space and by processing changes in a single batch. We show that the complexity of our scheme scales with the number of elementary changes to the filtration which as a result is often less expensive than the full persistent homology computation. We additionally adapt our approach to the cohomology algorithm and clearing optimizations. Finally, we perform computational experiments demonstrating practical speedups in several situations including feature generation and optimization guided by persistent homology.



(A) Effects on U-Net. (B) Effects on DenseDepth

FIGURE 6. **Effects of Topological Regularization on Interval Activations.** For each architecture, we visualize the second decoder layer. Our proposed regularization helps the network concentrate better on regions of interest.

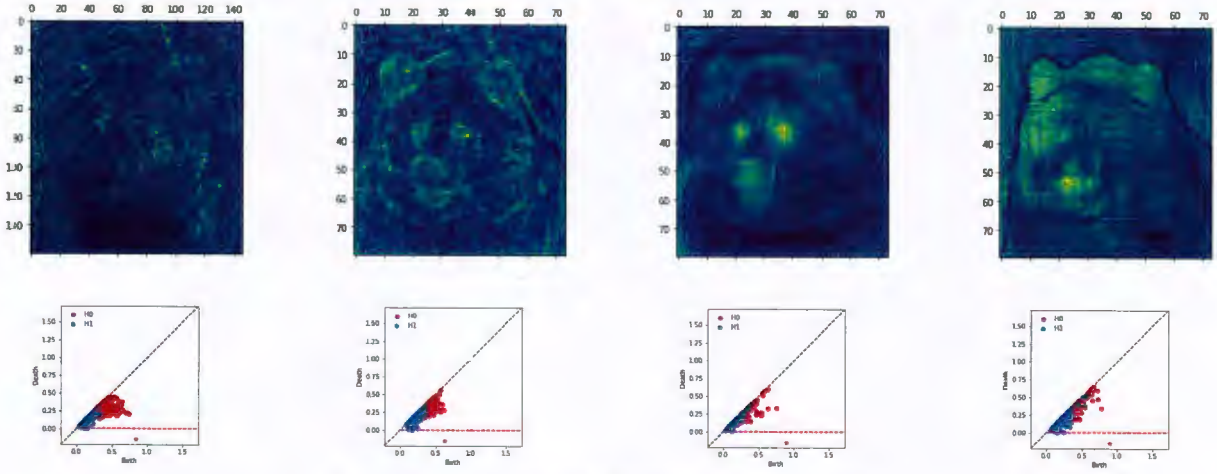


FIGURE 7. Activations and Persistent Diagrams in FCN

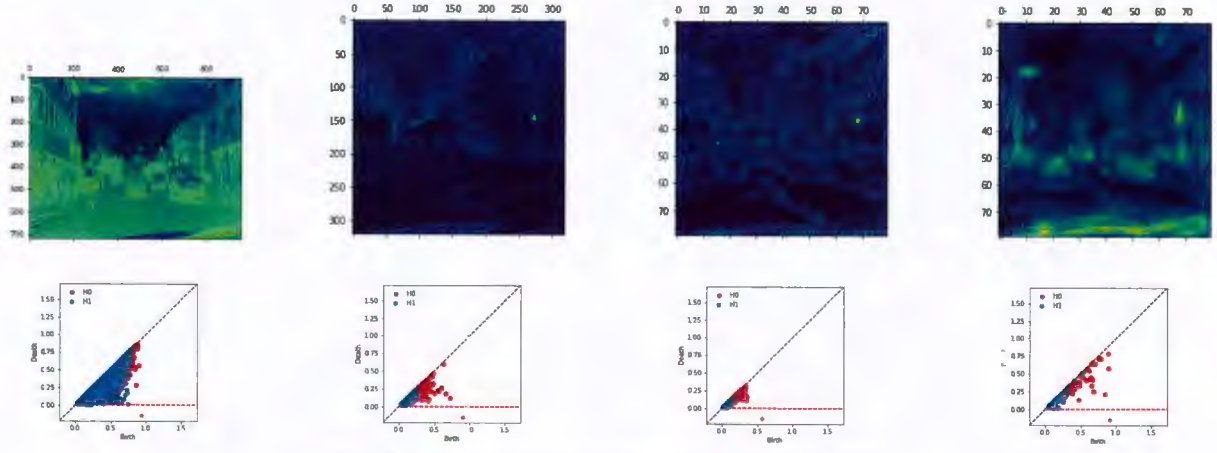


FIGURE 8. 2D Activations and Persistent Diagrams in TMANet

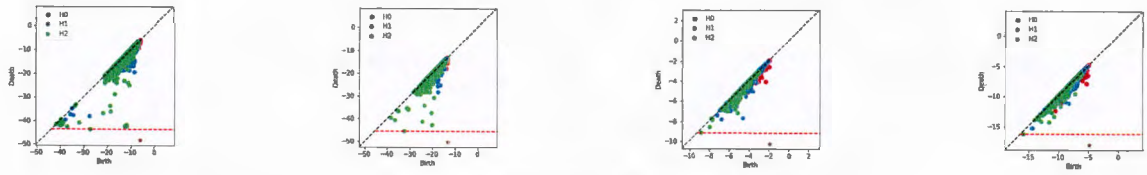


FIGURE 9. 3D Persistent Diagrams in TMANet

2.4.3. *Data and Experiment Results.* We investigate our methods on both synthetic and real data sets. We list those real ones below.

- **MNIST** [34]: A collection of handwritten digit images contains a training set of 60,000 examples, and a test set of 10,000 examples.
- **Vert-64**: A 3-dimensional rotational angiography scan of a head with an aneurysm used for benchmarking persistent homology in [49].
- **Bunny**: The Bunny model comes from the Stanford Computer Graphics Laboratory [58]. We use one of its 3D scan picture with size 40256 points in $\mathbb{R}^3$.
- **Dragon**: It is a 3-dimensional scan of a dragon from the Stanford Dragon graphic model [58] and in [49] consists of 1000 and 2000 points sampled uniformly at random.
- **H3N2**: The data set from [49] contains 2722 different genetic sequences of H3N2 influenza, where each sequence is a vector in $\mathbb{R}^{1173}$.

We provide a table of persistent homology computation of level set filtration below. More details and explanation are available in https://arxiv.org/abs/2108.05022. Overall, our update scheme gives a 2 to 3 times speedup compared to a full persistent homology computation.

		MNIST	Vert-64	S2D(0.01)	S2D(0.1)	S3D(0.01)	S3D(0.1)
Freudenthal	d_K	1.9×10^{-1}	–	3.3×10^{-3}	2.9×10^{-2}	1.4×10^{-2}	3.0×10^{-2}
	Ripser	2.2×10^{-3}	–	4.0×10^{-2}	$\mathbf{4.3 \times 10^{-2}}$	–	–
	Dionysus	3.0×10^{-3}	–	1.2×10^{-1}	1.3×10^{-1}	1.9×10^{0}	1.9×10^{0}
	Gudhi	4.6×10^{-3}	–	1.9×10^{-1}	2.0×10^{-1}	2.2×10^{0}	2.2×10^{0}
	BATS(c)	2.1×10^{-3}	–	6.6×10^{-2}	6.9×10^{-2}	8.1×10^{-1}	$\mathbf{8.7 \times 10^{-1}}$
	BATS(u,s)	$\mathbf{1.5 \times 10^{-3}}$	–	$\mathbf{3.9 \times 10^{-2}}$	6.1×10^{-2}	$\mathbf{6.9 \times 10^{-1}}$	9.8×10^{-1}
	BATS(u,c)	$\mathbf{1.5 \times 10^{-3}}$	–	$\mathbf{3.9 \times 10^{-2}}$	6.0×10^{-2}	7.0×10^{-1}	1.0×10^{0}
Cubical	d_K	2×10^{-1}	4.5×10^{-2}	3.3×10^{-3}	3.0×10^{-2}	1.4×10^{-2}	3.0^{-2}
	Gudhi	2.7×10^{-3}	2.7×10^{0}	3.0×10^{-2}	3.3×10^{-2}	$\mathbf{2.0 \times 10^{-1}}$	$\mathbf{2.1 \times 10^{-1}}$
	BATS(c)	2.2×10^{-3}	4.1×10^{0}	6.1×10^{-2}	7.3×10^{-2}	4.4×10^{-1}	4.7×10^{-1}
	BATS(u,s)	$\mathbf{1.2 \times 10^{-3}}$	1.3×10^{1}	$\mathbf{2.1 \times 10^{-2}}$	$\mathbf{3.2 \times 10^{-2}}$	2.1×10^{-1}	2.8×10^{-1}
	BATS(u,c)	$\mathbf{1.2 \times 10^{-3}}$	$\mathbf{2.1 \times 10^{0}}$	$\mathbf{2.1 \times 10^{-2}}$	$\mathbf{3.2 \times 10^{-2}}$	2.2×10^{-1}	2.7×10^{-1}

TABLE 4. Average time in seconds to recompute or update persistent homology of super-level set filtrations on synthetic and real data, using either Cubical complexes or the Freudenthal triangulation of a grid. d_K is the normalized Kendall–tau distance between the initial and updated filtrations averaged over experiments. Ripser [4], Dionysus [45], GUDHI [43], and BATS(c) [3] recompute persistent homology. Gudhi and Ripser both use cohomology, and Dionysus and BATS both use homology. BATS(c) uses clearing and does not form the basis V. BATS(u,s) updates the RU decomposition from the standard reduction algorithm and BATS(u,c) updates the RU decomposition obtained from clearing. Timings are averaged over 1000 updates for MNIST (using Image init. for the updating schemes), 1 update for Vert-64, 100 updates for S2D columns, and 20 updates for S3D columns. Timings for the Freudenthal triangulation of the Vert-64 data set are excluded due to memory constraints.

2.5. **Topology Preserving Dimension Reduction.** We have also developed a tool for maximizing the topological fidelity of visualizations [47]. Our work develops a notion of selection errors in persistent homology, and shows how to use the interleaving distance to control for selection errors.

2.5.1. *Background and Motivation.* Dimension reduction is an important component of many data analysis tasks, but can be potentially problematic as it may "reveal" structure in data which is not truly present. In inference this can be addressed by principled use of a withheld test set or an analysis which addresses model selection more directly. However, in exploratory data analysis it can be difficult to address selection problems incurred by exploration of different dimension reduction techniques and subsequent parameter tuning, such as whether visualized structures are really present or an artifact of the chosen embedding. In this project, we are considered about the topological correctness of an embedding and develop the use of the interleaving distance for the purpose of quantifying the extent to which *topological* features of an embedding relate to features in the original data set.

2.5.2. *Main Contribution.*

(1) We show how the interleaving distance can be used to quantify a scale at which topological features in X and features in Y are in correspondence, and be used to select homological features of Y in correspondence with features in X.
(2) We show how to incorporate the interleaving distance explicitly into the optimization of the embedding Y and prove the existence of descent directions under mild conditions. Explicitly, we consider the orthogonal optimization with bottleneck distance between two persistent diagrams of Vietoris–Rips Filtrations as its objective:

$$\underset{P \in \mathbb{R}^{d \times k}}{\text{minimize}} \, \mathrm{d_B}(\mathrm{dgm}(X), \mathrm{dgm}(PX)) \tag{11}$$

$$\text{subject to } P^T P = I \tag{12}$$

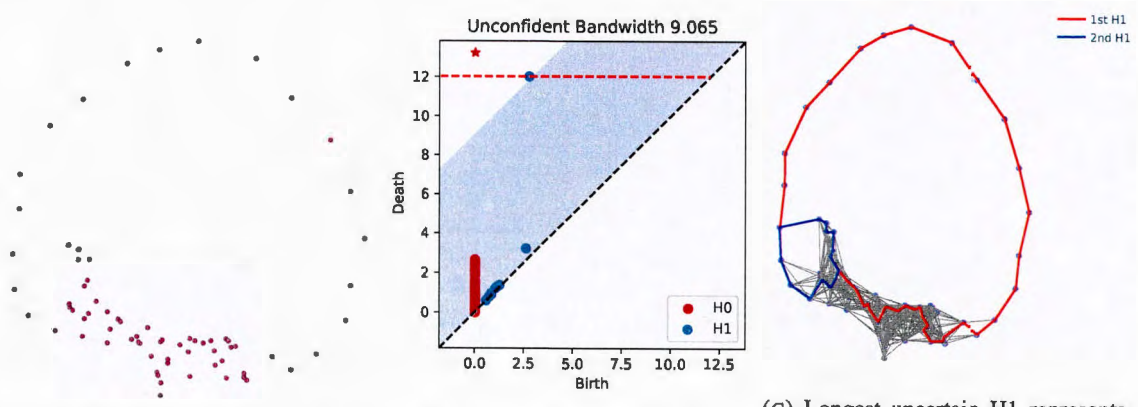
where $X \in \mathbb{R}^{n \times d}$ is a a high dimensional data set and k is a human readable dimension (2 in our case).
(3) We demonstrate this technique in finding optimal linear projections of the data set X to preserve the bottleneck distance on several examples with interesting topology.

2.5.3. *Data and Experiment Results.* Our real data sets includes

- The Columbia Object Image Library (COIL-100) [48] dataset contains 7200 colorful images of 100 objects, where each object has 72 128×128 images with 3 color channels taken at pose intervals of 5 degrees.
- Natural image patches are a well-studied data set with interesting topological structures at various densities [15]. We follow the data processing procedure of [35] to sample 3×3 patches from the van Hateren natural images database [60]. We further refine a sub-sample of 50,000 patches using the co-density estimator of [21] with $k = 5, p = 40\%$ to obtain a data set of 20,000 patches which resembles the "three-circle" model of [21].

We demonstrate our technique by assessing how well any transformation of point cloud data X preserves topology. In particular, we can compute the bottleneck distance $\epsilon = d_I(H_k(X), H_k(Y))$, the number of features of $H_k(Y)$ with $|d - b| > 2\epsilon$ and the number of features of $H_k(X)$ with $|d - b| > 4\epsilon$. In the context of dimension reduction, it would be desirable to minimize the interleaving distance in order to maximize the number of features we can identify which are in correspondence with features in the original data set.

In table 5, we compare several algorithms for dimension reduction on a set of images from COIL-100 [48]. Methods compared include PCA, MDS [33], and ISOMAP [57] with our method based on minimizing the bottleneck distance denoted by PH, and a hybrid PH + PCA. Because every Vietoris–Rips filtration has a single H_0 pair with death at ∞, at least one H_0 feature will always be selected. Only PH + PCA allows for the selection of an H_1 feature. A more detailed visualization of the PH + PCA embedding can be found in fig. 10.



(A) Dimension reduction result

(B) Persistence diagram with unconfident band

(C) Longest uncertain H1 representative (Blue) and certain H1 representative (Red)

FIGURE 10. Dimension reduction results of Tomato dataset by first using PCA to reduce dimension to 10 and then using our PH optimization to reduce to 2.

Method	max H_1	d_I H_0	d_I H_1	H_0	H_1
PCA	4.854	5.148	10.852	1	0
MDS	5.626	4.782	10.217	1	0
ISOMAP	113.968	1.935	108.623	1	0
PH	7.543	5.295	5.295	1	0
PH + PCA	9.234	4.689	4.532	1	1

TABLE 5. Selection of topological features. max H_1 is the length of the largest H_1 pair in Y. The last two columns indicate the number of features which are in correspondence with the original data in H_0 and H_1 via the interleaving.

2.6. **Persistent Path Homology.** Persistent path homology [18] is a variant of persistent homology which is applicable to directed graphs such as causal graphs. At a high level, it measures the number of distinct paths up to a version of homotopy [27]. Unfortunately, existing implementations are quite inefficient compared to software for computing persistent homology.

First we briefly describe how the original reduction algorithm can be applied in the persistent path homology setting.

Definition 2.1. An elementary p-path $[v_0, \ldots, v_p]$ is

- **regular** if $v_i \neq v_{i+1}$ for $i \in \{0, \ldots, p-1\}$.
- **irregular** if it is not regular.

Set

$$\mathcal{R}_p(V) = \text{span}\{\sigma \in \Lambda_p(V) : \sigma \text{ is regular}\}$$

and

$$\mathcal{I}_p(V) = \text{span}\{\sigma \in \Lambda_p(V) : \sigma \text{ is irregular}\}.$$

Definition 2.2. Let $G = (V, E)$ be a digraph. An elementary p-path $[v_0, \ldots, v_p]$ is **allowed** if $(v_i, v_{i+1}) \in E$ for each $i = 0, \ldots, p-1$. We denote the space of allowed p-paths as $\mathcal{A}_p$.

Definition 2.3. Given a digraph $G = (V, E)$ and a positive integer p, we define the **the space of ∂-invariant paths on** G to be

$$\Omega_p(G; \mathbb{K}) = \{\sigma \in \mathcal{A}_p : \partial_p(\sigma) \in \mathcal{A}_{p-1}\}.$$

We also set $\Omega_{-1} = \mathcal{A}_{-1}$ and $\Omega_{-2} = \mathcal{A}_{-2}$.

The path homology of the digraph G is the homology of the chain complex

$$H_p(\Omega(G)) = \ker \partial_p / \operatorname{img} \partial_{p+1}$$

and persistent path homology $PPH(G)$ is the persistent homology of a filtered chain complex $\Omega(G)$.

Suppose we have the boundary matrix $R_p : \mathcal{R}_p \to \mathcal{R}_{p-1}$. Ordinarily, we would already require the matrix to be in filtration order, which means that the allow times are sorted increasing from left to right (columns are elements of $\mathcal{R}_p$) and increasing from top to bottom (rows are elements of $\mathcal{R}_{p-1}$). Usually, there will be many elements of $\mathcal{R}_{p-1}$ that are in $\mathcal{A}_p^\infty$. Any column that ends up with a pivot where the corresponding cell is in $\mathcal{A}_p^\infty$ is obviously not in Ω_p. However, we can still have a natural interpretation where it corresponds to a bar that is both created and destroyed at time infinity (and thus does not correspond to any feature in the barcode).

We compare an implementation of the reduction algorithm that does the reduction via BATS [3] versus the original implementation of persistent path homology [19]. For simplicity we compute homology over $\mathbb{F}_2$. We computed 1-D PPH on simulated data introduced in [19] called "randNet," a complete digraph with edges added in a random order (used in [19] to demonstrate the original implementation) Note that with the

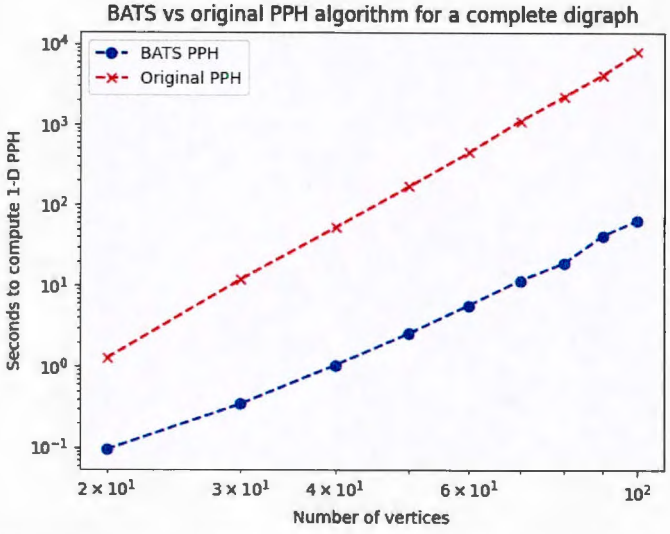


FIGURE 11. A comparison of the runtime of the BATS-powered PPH implementation with the original algorithm. We compute 1-D PPH over $\mathbb{F}_2$ on a complete digraph, with edges added in random order.

complete digraph on 100 vertices (with 9900 directed edges), the BATS implementation was over 120 times faster than the original implementation. By looking at the log-log plot (bottom of Figure 11) and regressing the slopes, we note that the BATS implementation has an exponent of 4.10 compared to 5.40 for the original implementation.

One next step would be to explore *zigzag path homology*. *Zigzag homology* (introduced in [13, 14] with further advances in [11, 44, 50, 42, 46]) was developed to generalize persistence homology to the setting where instead of a filtration of complexes, the inclusion maps can be in either direction. Given our complexes K_i, rather than each inclusion map looking like

$$\cdots \to K_{i-1} \to K_i \to K_{i+1} \to \ldots,$$

the inclusion maps can look like

$$\cdots \to K_{i-1} \to K_i \leftarrow K_{i+1} \to \ldots.$$

A (discrete) *temporal graph* is a sequence of graphs $G_1, \ldots, G_n$ of graphs that share a vertex set V. We also we require that either $G_i \subseteq G_{i+1}$ or $G_i \supseteq G_{i+1}$, which gives inclusion maps in one of two directions between each G_i and G_{i+1}. We interpret the graphs G_i to be changing over discrete time intervals, so that

upon each "update" edges are either added or deleted. This gives rise to a sequence of inclusion maps that can be studied via zigzag techniques.

ATAS already has robust tools to compute ordinary persistence, a natural next step would be to develop *zigzag path homology*, although some care must be ensured given that we must account for both the allow time and entry time for each path.

2.7. **Additional Funded Work.** In previous sections, we covered several lines of work which we have pursued in the aim of developing methods for topological and geometric methods for time aware machine intelligence. Both PIs and students have also produced related work in topology, geometry, and neural networks while funded by this grant.

In computational topology, PI Nelson [29] has developed new methods for parameterizing and compressing expensive topological constructions on point cloud data. This may find applications to accelerating optimization of persistent homology, a method used in our work. In joint work with G. Carlsson, he has also developed computational methods for solving the topological evasion problem [12]. While not directly applicable to neural networks, this work solves a class of temporal evasion problems that arise naturally in sensor networks.

Co-PI Lim has developed methods to work with data in subspaces of varying dimensions [64, 8, 59]. He has also found a way to optimally approximate neural networks of fixed sizes [38], obtained new methods for multiparameter eigenvalue problems [53] and written an extensive review of tensors in computations [37].

3. Conclusion & Plan Forward

Throughout the Time-Aware Machine Intelligence program, we have made progress in several directions that use topology and geometry to understand and enhance the capabilities of time-aware neural networks. Highlight of our work include

(1) Incorporation of Lorentz equivariance into spatio-temporal prediction tasks in neural networks. Our results currently demonstrate large improvements over non-equivariant networks of similar depth.
(2) Use of persistent homology to understand and regularize activations in dense prediction tasks in images/video. This has allowed us to improve the performance state-of-the-art networks.
(3) A variety of improvements to algorithms to support these topological and geometric tools, including acceleration of persistent homology in optimization, improved topological data visualization, and greatly accelerated persistent path homology.

We plan to pursue the following projects which build on the work that we were able to begin thanks to this program:

Lorentz Equivariant Networks. We plan to incorporate equivariance into physics-informed neural networks [52] which will represent a non-trivial application of the ideas that we have developed and tested on simpler network architectures. Another promising direction of future work will be to develop methods to quantify the failure of equivariance in non-equivariant networks architectures, and use this as a measurement of performance of neural networks on problems with symmetry.

Topology in Dense Prediction. Our work using topology in dense prediction problems represents one of the first uses of persistent homology to regularize large classes of neural networks out of the box, without specifically tailoring architectures to use topological methods. A promising direction of future work is to improve the performance of topological regularization; while not cost prohibitive, optimizations such as moving to GPU and incorporating algorithmic improvements may result in much lower computational overhead. Dense prediction in video data offers a larger of variety of ways in which topology may be used for understanding and regularization. While we have made promising progress on this task, future work would involve comparing our 3-dimensional image approach with different distances (such as bottleneck or Wasserstein distances) on fixed frames of the video.

Topological Calculations. We have made several substantial improvements to algorithms and implementations for persistent homology (PH) and persistent path homology (PPH). Moving forward, our PPH implementation can be used for analysis of larger temporal graphs than have been previously been feasible. We also plan to extend to zigzag path homology for application to sliding temporal windows. Our PH implementations will allow for accelerated use of persistent homology in optimization objectives. Our work on accelerating persistent homology calculations with warm starts might also be adapted to the setting of PPH and used in optimization over causal graphs.

REFERENCES

[1] Aron Alcock, Erik Carlsson, and Gunnar Carlsson. "The ring of algebraic functions on persistence bar codes". en. In: *Homology, Homotopy and Applications* 18.1 (2016), pp. 381–402. ISSN: 15320073, 15320081. DOI: 10.4310/HHA.2016.v18.n1.a21. URL: http://www.intlpress.com/site/pub/pages/journals/items/hha/content/vols/0018/0001/a021/ (visited on 07/03/2018).

[2] Ibraheem Alhashim and Peter Wonka. "High Quality Monocular Depth Estimation via Transfer Learning". In: *ArXiv* abs/1812.11941 (2018).

[3] *Basic Applied Topology Subprograms (BATS)*. URL: https://github.com/bnels/BATS.

[4] Ulrich Bauer. "Ripser: efficient computation of Vietoris–Rips persistence barcodes". In: *Journal of Applied and Computational Topology* (2021). DOI: 10.1007/s41468-021-00071-5.

[5] Alexander Bogatskiy et al. "Lorentz group equivariant neural network for particle physics". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 992–1002.

[6] Rickard Brüel Gabrielsson and Gunnar Carlsson. "Exposition and Interpretation of the Topology of Neural Networks". In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). Dec. 2019, pp. 1069–1076. DOI: 10.1109/ICMLA.2019.00180.

[7] Rickard Brüel-Gabrielsson et al. "ATopology Layer for Machine Learning". In: *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*. The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS). Apr. 24, 2020. arXiv: 1905.12200. URL: http://arxiv.org/abs/1905.12200 (visited on 05/18/2020).

[8] Yuhang Cai and Lek-Heng Lim. "Distances between probability distributions of different dimensions". In: *IEEE Trans. Inform. Theory* 68.6 (2022), pp. 4020–4031. ISSN: 0018-9448.

[9] Zixuan Cang and Guo-Wei Wei. "Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction". In: *International Journal for Numerical Methods in Biomedical Engineering* 34.2 (2018). ISSN: 2040-7947. DOI: 10.1002/cnm.2914.

[10] Gunnar Carlsson. "Topology and data". In: *Bulletin of the American Mathematical Society* 46.2 (Jan. 29, 2009), pp. 255–308. ISSN: 0273-0979. DOI: 10.1090/S0273-0979-09-01249-X. URL: http://www.ams.org/journal-getitem?pii=S0273-0979-09-01249-X (visited on 10/22/2019).

[11] Gunnar Carlsson, Anjan Dwaraknath, and Bradley J. Nelson. *Persistent and Zigzag Homology: A Matrix Factorization Viewpoint*. Available at arXiv:1911.10693. 2019. DOI: 10.48550/ARXIV.1911.10693. URL: https://arxiv.org/abs/1911.10693.

[12] Gunnar Carlsson and Bradley J. Nelson. *Computational Spanier-Whitehead Duality and the Structure of Complements*. In preparation. 2022.

[13] Gunnar Carlsson and Vin Silva. "Zigzag Persistence". In: *Foundations of Computational Mathematics* 10 (Dec. 2008), pp. 367–405. DOI: 10.1007/s10208-010-9066-0.

[14] Gunnar Carlsson, Vin de Silva, and Dmitriy Morozov. "Zigzag Persistent Homology and Real-Valued Functions". In: *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*. SCG '09. Aarhus, Denmark: Association for Computing Machinery, 2009, pp. 247–256. ISBN: 9781605585017. DOI: 10.1145/1542362.1542408. URL: https://doi.org/10.1145/1542362.1542408.

[15] Gunnar Carlsson et al. "On the Local Behavior of Spaces of Natural Images". In: *International Journal of Computer Vision* 76.1 (Jan. 2008), pp. 1–12. ISSN: 0920-5691, 1573-1405 DOI: 10.1007/s11263-007-0056-x.

[16] Mathieu Carrière et al. "PersLay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures". Preprint: http://arxiv.org/abs/1904.09378. Mar. 8, 2020. arXiv: 1904.09378. URL: http://arxiv.org/abs/1904.09378 (visited on 06/10/2020).

[17] Chao Chen et al. "A Topological Regularizer for Classifiers via Persistent Homology". In: *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*. The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS). 2019. arXiv: 1806.10714. URL: http://arxiv.org/abs/1806.10714 (visited on 06/10/2020).

[18] C. Chowdhury and F. Memoli. "Persistent Path Homology of Directed Networks". In: *Symposium on Discrete Algorithms (SODA)*. 2018.

[19] Samir Chowdhury and Facundo Mémoli. *pypph*. https://github.com/samirchowdhury/pypph. 2018.

[20] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. "Vines and Vineyards by Updating Persistence in Linear Time". en. In: (2006), p. 8.

[21] Vin de Silva and Gunnar Carlsson. "Topological Estimation Using Witness Complexes". In: *Proc. Sympos. Point-Based Graphics*. June 2004. DOI: 10.2312/SPBG/SPBG04/157-166.

[22] Marc Finzi, Max Welling, and Andrew Gordon Wilson. "A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3318–3328.

[23] Deqing Fu and Bradley J. Nelson. "Topological Regularization for Dense Prediction". In: *arXiv:2111.10984 [cs, math]* (Nov. 2021). arXiv: 2111.10984 [cs, math].

[24] Kaifu Gao et al. "Repositioning of 8565 Existing Drugs for COVID-19". In: *The Journal of Physical Chemistry Letters* (June 2020). DOI: 10.1021/acs.jpclett.0c01579.

[25] Chad Giusti, Robert Ghrist, and Danielle S. Bassett. "Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data". In: *Journal of Computational Neuroscience* 41.1 (Aug. 2016), pp. 1–14. ISSN: 0929-5313, 1573-6873. DOI: 10.1007/s10827-016-0608-6. URL: http://link.springer.com/10.1007/s10827-016-0608-6 (visited on 03/27/2020).

[26] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 6602–6611.

[27] A Grigor'yan et al. "Homotopy Theory for Digraphs". In: *Pure and Applied Mathematics Quarterly* 10.4 (2014), pp. 619–674.

[28] Zeyuan Hu and Julia Strout. "Exploring Stereotypes and Biased Data with the Crowd". In: *arXiv* abs/1801.03261 (2018).

[29] Bradley J. Nelson. *Parameterized Vietoris-Rips Filtrations via Covers*. Preprint. 2022.

[30] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. "Design of an image edge detection filter using the Sobel operator". In: *IEEE Journal of solid-state circuits* 23.2 (1988), pp. 358–367.

[31] Kwangho Kim et al. "Efficient Topological Layer based on Persistent Landscapes". Preprint: http://arxiv.org/abs/2002.02778. Feb. 7, 2020. arXiv: 2002.02778. URL: http://arxiv.org/abs/2002.02778 (visited on 06/10/2020).

[32] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2015).

[33] J. B. Kruskal. "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis". In: *Psychometrika* 29.1 (Mar. 1964), pp. 1–27. ISSN: 1860-0980. DOI: 10.1007/BF02289565.

[34] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[35] Ann B. Lee, Kim S. Pedersen, and David Bryant Mumford. "The Nonlinear Statistics of High-Contrast Patches in Natural Images". In: *International Journal of Computer Vision* 54.1-3 (2003), pp. 83–103. ISSN: 09205691. DOI: 10.1023/A:1023705401078.

[36] Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. "AFramework for Differential Calculus on Persistence Barcodes". In: *Foundations of Computational Mathematics* (July 2021) ISSN: 1615-3383. DOI: 10.1007/s10208-021-09522-y.

[37] Lek-Heng Lim. "Tensors in computations". In: *Acta Numer.* 30 (2021), pp. 555–764. ISSN: 0962-4929. DOI: 10.1017/S0962492921000076. URL: https://doi.org/10.1017/S0962492921000076.

[38] Lek-Heng Lim, Mateusz Michałek, and Yang Qi. "Best k-layer neural network approximations". In: *Constr. Approx.* 55.1 (2022), pp. 583–604. ISSN: 0176-4276. DOI: 10.1007/s00365-021-09545-2. URL: https://doi.org/10.1007/s00365-021-09545-2.

[39] Lek-Heng Lim and Bradley J. Nelson. "What is an Equivariant Neural Network?" submitted. 2022.

[40] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *ECCV*. 2014.

[41] Yuan Luo and Bradley J. Nelson. "Accelerating Iterated Persistent Homology Computations with Warm Starts". In: *arXiv preprint arXiv:2108.05022* (2021).

[42] Clément Maria and Steve Y. Oudot. "Zigzag Persistence via Reflections and Transpositions". In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Agorithms*. SODA'15. San Diego, California: Society for Industrial and Applied Mathematics, 2015, pp. 181–199.

[43] Clément Maria et al. "The Gudhi Library: Simplicial Complexes and Persistent Homology". In: *Mathematical Software – ICMS 2014*. Ed. by Hoon Hong and Chee Yap. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2014, pp. 167–174. ISBN: 978-3-662-44199-2. DOI: 10.1007/978-3-662-44199-2_28.

[44] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. "Zigzag Persistent Homology in Matrix Multiplication Time". In: *Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry*. SoCG '11. Paris, France: Association for Computing Machinery, 2011, pp. 216–225. ISBN: 9781450306829. DOI: 10.1145/1998196.1998229. URL: https://doi.org/10.1145/1998196.1998229.

[45] Dmitiry Morozov. *Dionysus2*. Software available at https://www.mrzv.org/software/dionysus2/.

[46] Bradley J. Nelson. "Parameterized Topological Data Analysis". PhD dissertation. Stanford University, June 2020.

[47] Bradley J. Nelson and Yuan Luo. "Topology-Preserving Dimensionality Reduction via Interleaving Optimization". In: *arXiv:2201.13012 [cs, math]* (Jan. 2022). arXiv: 2201.13012 [cs, math].

[48] Sameer A Nene, Shree K. Nayar, and Hiroshi Murase. *object image library (COIL-100*. Tech. rep. 1996.

[49] Nina Otter et al. "Aroadmap for the computation of persistent homology". In: *EPJ Data Science* 6.1 (Dec. 2017). ISSN: 2193-1127. DOI: 10.1140/epjds/s13688-017-0109-5. arXiv: 1506.08903. URL: http://arxiv.org/abs/1506.08903 (visited on 02/08/2018).

[50] Steve Y. Oudot and Donald R. Sheehy. "Zigzag Zoology: Rips Zigzags for Homology Inference". In: *Proceedings of the Twenty-Ninth Annual Symposium on Computational Geometry*. SoCG '13. Rio de Janeiro, Brazil: Association for Computing Machinery, 2013, pp. 387–396. ISBN: 9781450320313. DOI: 10.1145/2462356.2462371. URL: https://doi.org/10.1145/2462356.2462371.

[51] R. Brüel-Gabrielsson et al. "ATopology Layer for Machine Learning". In: *AISTATS, Proceedings of Machine Learning Research*. Vol. 108. 2020, pp. 1553–1563.

[52] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: Adeep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". en. In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. ISSN: 0021-9991.

DOI: 10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125 (visited on 05/05/2022).

[53] Jose Israel Rodriguez et al. "Fiber product homotopy method for multiparameter eigenvalue problems". In: *Numer. Math.* 148.4 (2021), pp. 853–888. ISSN: 0029-599X. DOI: 10.1007/s00211-021-01215-6. URL: https://doi.org/10.1007/s00211-021-01215-6.

[54] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *MICCAI*. 2015.

[55] Evan Shelhamer, Jonathan Long, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.4 (2016), pp. 640–651.

[56] Irwin Sobel and G. M. Feldman. "An Isotropic 3×3 image gradient operator". In: 1990.

[57] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (Dec. 2000), pp. 2319–2323. ISSN: 00368075, 10959203. DOI: 10.1126/science.290.5500.2319.

[58] *The Stanford 3D Scanning Repository*. http://graphics.stanford.edu/data/3Dscanrep/. Accessed: 2021-07-23.

[59] Brian St. Thomas et al. "Learning Subspaces of Different Dimensions". In: *J. Comput. Graph. Statist.* 31.2 (2022), pp. 337–350. ISSN: 1061-8600. DOI: 10.1080/10618600.2021.2000420. URL: https://doi.org/10.1080/10618600.2021.2000420.

[60] J. H. van Hateren and A van der Schaaf. "Independent Component Filters of Natural Images Compared with Simple Cells in Primary Visual Cortex". In: *Proceedings. Biological Sciences* 265.1394 (Mar. 1998), pp. 359–366. ISSN: 0962-8452. DOI: 10.1098/rspb.1998.0303.

[61] Igor Vasiljevic et al. "DIODE: A Dense Indoor and Outdoor DEpth Dataset". In: *CoRR* abs/1908.00463 (2019). URL: http://arxiv.org/abs/1908.00463.

[62] Hao Wang, Weining Wang, and Jing Liu. "Temporal memory attention for video semantic segmentation". In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 2254–2258.

[63] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13 (2004), pp. 600–612.

[64] Ke Ye, Ken Sze-Wai Wong, and Lek-Heng Lim. "Optimization on flag manifolds". In: *Math. Program.* 194.1-2, Ser. A(2022), pp. 621–660. ISSN: 0025-5610. DOI: 10.1007/s10107-021-01640-3. URL: https://doi.org/10.1007/s10107-021-01640-3

[65] Dora Zhao, Angelina Wang, and Olga Russakovsky. "Understanding and Evaluating Racial Biases in Image Captioning". In: *International Conference on Computer Vision (ICCV)*. 2021.