ICLR 2021 CHALLENGE FOR COMPUTATIONAL GEOMETRY & TOPOLOGY: DESIGN AND RESULTS

Nina Miolane*
UC Santa Barbara, USA
ninamiolane@ucsb.edu

Matteo Caorsi*
L2F SA, Switzerland
matteocao@gmail.com

Umberto Lupo*
EPFL, Switzerland
umberto.lupo@epfl.ch

Marius Guerard*
Geomstats, USA

Nicolas Guigui*
Inria Sophia-Antipolis, France

Johan Mathe*
Atmo, USA

Yann Cabanes*
Institute of Mathematics of Bordeaux and Thales, France

Wojciech Reise*
Inria Paris-Saclay, France

Thomas Davies**
University of Southampton, UK

António Leitão**
NOVA IMS, Portugal

Somesh Mohapatra**
MIT, USA

Saiteja Utpala**
Mastercard, India

Shailja Shailja**
UC Santa Barbara, USA

Gabriele Corso**
Cambridge, UK

Guoxi Liu**
Clemson University, USA

Federico Iuricich**
Clemson University, USA

Andrei Manolache**
Bitdefender and University of Bucharest, Romania

Mihaela Nistor, Matei Bejan**, Armand Mihai Nicolicioiu**, Bogdan-Alexandru Luchian****
University of Bucharest, Romania

Mihai-Sorin Stupariu**
University of Bucharest, Romania

Florent Michel, Khanh Dao Duc****
University of British Columbia, Canada

Bilal Abdulrahman**
CUNY Graduate Center, USA

Elodie Maignant**
Inria and ENS Paris-Saclay, France

Zhiyuan Liu**
University of North Carolina, USA

Marek Černý**
Charles University, Czech Republic

Martin Bauw, Santiago Velasco-Forero, Jesus Angulo**
Mines ParisTech, PSL Research University, France

Yanan Long**
University of Chicago, USA

* : Organizers and external jury; ** : Participants

ABSTRACT

This paper presents the computational challenge on differential geometry and topology that happened within the ICLR 2021 workshop "Geometric and Topological Representation Learning". The competition asked participants to provide creative contributions to the fields of computational geometry and topology through the open-source repositories Geomstats and Giotto-TDA. The challenge at-

1

tracted 16 teams in its two month duration. This paper describes the design of the challenge and summarizes its main findings.

Code: `https://github.com/geomstats/challenge-iclr-2021`.

1 INTRODUCTION

The motivation behind the organization of the ICLR 2021 Computational Geometry and Topology challenge was two-folds: first, to push forward the fields of computational differential geometry and topology; and second, to foster *reproducible research* in mathematics by encouraging the use, development and maintenance of open-source repositories.

Reproducible Research   The *reproducibility* of a (computational) experiment is widely regarded as a requirement to establish a scientific claim or to demonstrate the applicability of a technology. Different authors have suggested different levels of *reproducibility* (Dalle, 2012; Stodden et al., 2013; Schnell, 2018). These levels range from *repeatability* – the ability of the same team to repeat the same experiment with the same methodological set up – to the stronger notion of *reproducibility* – the ability of a different team to reproduce the results with a different methodological set up. In computational and mathematical sciences, such "methodological setup" refers to the software environment, the data, and raw code.

Despite many valuable initiatives to improve reproducibility in science, minimum standards are rarely met – even in the mathematical sciences (David Redish et al., 2018). This so-called *reproducibility crisis* has led researchers, funding agencies, politicians, and the wider audience to question the reliability of the scientific enterprise. Among other attempts to address the crisis, the workshop "ICERM Workshop on Reproducibility in Computational and Experimental Mathematics (2012)" laid out several recommendations to the research community. In particular, the workshop committee claimed that "appropriate tools" should be taught as standard operating procedure in relation to computational aspects of research (Stodden et al., 2013).

Computational notebooks   What are today's "appropriate tools" in modern computational and mathematical sciences? The most commonly used tools may not be the most appropriate anymore. The publication process has barely evolved since the 17th century, as the research paper (or its electronic form) still represents the principal mean of diffusion for mathematical ideas. As James Sommers writes: "Scientific results today are as often as not found with the help of computers. [...] And yet by far the most popular tool we have for communicating these results is the PDF - literally a simulation of a piece of paper. Maybe we can do better."

Many new "appropriate tools" are available to mathematical sciences to go beyond the sole diffusion format of the research paper. Among them, the computational notebook is an excellent candidate that can complement the traditional PDF paper (Oakes et al., 2019). Even a very theoretical mathematical paper could be completed by a computational notebook, that would: (i) use symbolic computation software to automatically check equations, (ii) leverage packages to check the veracity of a theorem on specific examples, (iii) provide interactive visualizations of special cases for the theoretical concepts exposed. This computational notebook could also be run automatically by the editor board of the corresponding journal, hence relieving some aspects of the review process and directly fostering reproducibility.

Open-source packages   Computational notebooks would ideally heavily leverage a shared implementation of the core concepts of a given field of mathematics. This implementation would be materialized as a piece of open-source software, whose computations would be constantly checked by its community. As such, well-maintained open-source software and computational notebooks represent the foundational "appropriate tools" that can foster reproducibility in mathematical research and with it, improve the efficiency and reliability of the research enterprise. Many open-source packages have made code available to foster reproducible research in their respective fields of mathematics.

In the field of differential geometry, we find `Pygeometry` (Censi, 2012), `Manopt` (Boumal et al., 2013), `Pyquaternion` (Wynn, 2014), `Pyriemann` (Barachant, 2015), `PyManopt` (Townsend et al., 2016), `TheanoGeometry` (Kühnel & Sommer, 2017), `Geoopt` (Kochurov et al., 2019), `Geomstats` (Miolane et al., 2020), the `SageManifold` project within the package `SageMath`

(Developers et al., 2020), `Tensorflow Manopt` (Smirnov, 2021), and `JuliaManifolds` (Axen et al., 2021), among others. In the field of topology, we find `Perseus` (Nanda, 2012), `Dipha` (Kerber et al., 2014), `Javaplex` (Tausz et al., 2014), `TDA` (Fasy et al., 2015), `Dionysus` (Morozov, 2015), `Eireen` (Henselman & Ghrist, 2016), `PHAT` (Bauer et al., 2017), the Topology ToolKit `TTK` (Tierny et al., 2018), `RedHom` (Brendel et al., 2019), `Scikit-tda` (Saul & Tralie, 2019), `Giotto-TDA` (Tauzin et al., 2020), `HomCloud` (Obayashi et al., 2020), `Diamorse` (Robins & Delgado-Friedrichs, 2020), `Gudhi` (The GUDHI Project, 2021), `GDA-public` (Geom-Data, 2021), and `Ripser` (Bauer, 2021), to cite a few.

Despite the existence of these packages, computational notebooks do not always accompany the submission of a mathematical research paper. Furthermore, recruiting maintainers to ensure the validity of the code on these platforms is often difficult. Both issues can be explained by a lack of incentives in the associated scientific communities.

Incentives Computational notebooks and open-source software – such as the ones referenced in the previous paragraph – are gaining popularity in several fields of research, for example within machine learning communities. However, they might be still under-used in the mathematical sciences for three main reasons. First, traditional mathematical training rarely introduce notebooks and software engineering as part of the curriculum. In differential geometry for example, textbooks may lack coding exercises or an associated interactive library. As a result, mathematicians do not necessarily master the tools available to use or write code associated with their findings. Second, many fields of mathematics lack a reference platform, such as a designated software, where researchers can share their computations and together contribute to their field. Third, there are only few incentives that motivate junior researchers in the mathematical sciences to learn good practices. The "publish-or-perish" pressure can make it difficult for junior researchers to consider taking additional time to (learn to) implement and share their results. As a consequence, and specifically in differential geometry and topology, it can be challenging to reproduce results, even if they were produced by the same team.

Computational Geometry and Topology Challenge The ICLR 2021 Computational Geometry and Topology Challenge aimed to address these issues by encouraging researchers to delve into open-source implementations of differential geometry and topology. The participants were asked to create computational notebooks using the open-source software `Geomstats` (Miolane et al., 2020) and `Giotto-TDA` (Tauzin et al., 2020). The goal was to showcase some of the aforementioned "appropriate tools" for modern research in the mathematical sciences. The participants of the challenge were rewarded by the publication of the present paper and with prizes for the three winning teams.

Outline and contributions The remainder of this paper is organized as follows. Section 2 introduces the setup and guidelines of the challenge. Section 3 summarizes the submissions to the challenge. Section 4 presents the main features used by the participants within the packages `Geomstats` and `Giotto-TDA`. Section 5 presents the limitations of the packages as reported by the participants. Section 6 provides a list of new features proposed by the participants that aim to enhance current implementations of computational geometry and topology. Section 7 describes the challenge's evaluation process and gives the final ranking of the submissions to the challenge.

2 SETUP OF THE CHALLENGE

The challenge was held in conjunction with the workshop "Geometric and Topological Representation Learning" of the International Conference on Learning Representations (ICLR) 2021.

Guidelines The participants were asked to submit a `Jupyter Notebook` to provide "the best data analysis, computational method, or numerical experiment relying on state-of-the-art geometric and topological Python packages": `Geomstats` and `Giotto-TDA`. The participants submitted their `Jupyter Notebooks` via Pull Requests (PR) to the GitHub repository of the challenge [1]. Teams were accepted and there was no restriction on the number of team members. The current

---

[1]`https://github.com/geomstats/challenge-iclr-2021`

principal developers of `Geomstats` and `Giotto-TDA`, i.e. the co-authors of the published papers (Miolane et al., 2020; Tauzin et al., 2020), were not allowed to participate.

Each submission was requested to respect the following structure: (i) Introduction and motivation, (ii) Analysis/Experiment, (iii) Benchmark, (iv) Limitations and perspectives. Guidelines were also giving examples of possible submissions:

- Data analysis with geometric and topological methods,
- Implementation of a research paper with `Geomstats`/`Giotto-TDA`
- Implementation of a feature to merge into `Geomstats`/`Giotto-TDA` codebases,
- Implementation of visualization tools for `Geomstats`/`Giotto-TDA`,
- Benchmarking/profiling on geometric and topological methods against other methods for a public dataset.

This list was completed by the `submission-example-*` folders on the GitHub repository, to help participants understand the packages and design their submission.

Evaluation criterion: fostering creativity   The evaluation criterion was: "how does the submission help push forward the fields of computational geometry and topology?". The submissions were ranked according to this evaluation criterion, through a voting procedure relying on the Condorcet method, see Section 7.

The choice of this evaluation criterion was motivated by several reasons. First, the criterion did not require participants to submit novel research. The main focus was the implementation, which could for instance be the reproduction of published research. Such a criterion can allow the participants to focus on producing clean code and to provide a hands-on explanation of the mathematical concepts at hand.

This criterion also did not bias participants towards showcasing "positive results" such as a new method beating the state of the art. "Negative results" were considered just as valuable as positive results. In particular, submissions criticizing the available packages, or showing examples where geometric and topological representations did not help the analysis were also significantly rewarded.

Lastly, such criterion encouraged participants to be generally creative. Most machine learning challenges are conducted by ranking the participants according to a quantitative metric on a test dataset. This can induce biases in the contributions of the participants, since methods that do not perform on that specific metric are not rewarded. While they have many other advantages, such criteria may hide interesting research. In contrast, our evaluation criterion, relying on a voting system through the Condorcet method, was meant to also reward creative submissions.

Software engineering practices   The participants were also encouraged to use software engineering best practices. Their code should be compatible with Python 3.8 and make an effort to respect the Python style guide PEP8. The `Jupyter notebooks` were automatically tested when a Pull Request was submitted and the tests were required to pass. If a dataset was used, the dataset had to be public and referenced. Participants could raise GitHub issues and/or request help or guidance at any time through `Geomstats` slack and `Giotto-TDA` slack. The help/guidance was be provided modulo availability of the maintainers.

3 SUBMISSIONS TO THE CHALLENGE

Sixteen teams submitted code before the deadline and participated in the challenge. This section provides a summary of their submissions.

Noise Invariant Topological Features   This submission analyzes data topological structure whilst being robust to various data corruptions. Examples of perturbed data are noisy point clouds, photos taken from different views, or dynamic modeling. This submission showcases the pipeline for extracting Perturbed Topological Signatures (PTS) by using `Geomstats` and `Giotto-TDA` (Som et al., 2018). The topological properties are studied by using distance metrics and kernels defined on the Stiefel and Grassmann manifolds (Edelman et al., 1999; Hamm & Lee, 2009; Som et al.,

2018). Experiments are performed on three datasets: SHREC 2010 (Veltkamp et al., 2010), Princeton COS429 (COS429, 2009), and MNIST (LeCun et al., 2010).

Estimators of Means of Symmetric Positive Matrices　This submission investigates estimators of means of Symmetric Positive Definite (SPD) matrices. In the first notebook, the efficiency of the recursive estimation of the Karcher mean (Ho et al., 2013), and of the K-means algorithm relying on it, are benchmarked and improved. In the second notebook, the Shrinkage Estimator is implemented (Yang et al., 2020), and the notebook shows how it improves on the maximum likelihood estimator. Experiments rely on synthetic datasets on the manifold of SPD matrices using sampling methods on manifolds (Schwartzman, 2016).

Visualization of Kendall Shape Spaces for Triangles　This submission introduces visualization methods for Kendall shape spaces of triangles. An object's shape can be described by locating relevant points on it, called landmarks (Dryden & Mardia, 2016). The Kendall shape space of triangles in dimension $m = 2, 3$ is the space of triangles quotiented by the group of rotations, translations and dilatations of $\mathbb{R}^m$ (Kendall, 1984; Le & Kendall, 1993). This submission presents two new visualization methods of these Kendall shape spaces, demonstrates their use and compares them with an alternative visualization method for this dataset: the non-exact visualization of multidimensional scaling (MDS). The experiments are performed on synthetic data of triangles in $\mathbb{R}^m$ for $m = 2, 3$, and on the dataset of optic nerve heads shapes from `Geomstats' datasets` module.

Map your Topology to Different Geometries　This submission implements a method to map a set of points from one geometry of choice onto another while preserving the topology. In the context of this notebook, a "geometry" refers to a Riemannian manifold such as Euclidean space, Hyperbolic space, Hypersphere, manifold of Symmetric Positive Definite (SPD) matrices, among others. The method uses gradient descent on Riemannian manifolds with a loss function introduced in (Moor et al., 2020) that has been used in Deep Learning (Moor et al., 2020; Gabrielsson et al., 2020). Experiments are run on synthetic data generated on the Euclidean plane, the sphere and the Poincare ball.

Naive Image Anomaly Detection on Fashion MNIST　This submission evaluates the possibility to achieve anomaly detection (AD) in image databases with naive distances to centroids and norms using Euclidean and Riemannian representations. The notebook considers simple AD setups where the objective is to discriminate between two classes of the Fashion MNIST dataset (Xiao et al., 2017). A general approach to embed images into the space of covariance matrices is introduced based on (Calvo & Oller, 1990). The best performances are achieved by the method relying on the norm of the negated geodesic principal component analysis (PCA) with the Fréchet mean as PCA base point (Rippel et al., 2020), using the Log-Euclidean Riemannian metric.

Shape Analysis of Bone Cancer Cells　This submission studies osteosarcoma (bone cancer) cells and the impact of drug treatment on their morphological shapes. The analysis uses cell images obtained from fluorescence microscopy. The corresponding dataset has been added into the `Geomstats'` module `datasets` by the participants. Cell shapes are modelled as discrete (open) curves. The submission uses the Riemannian elastic metric on discrete curves to compare cell shapes (Jermyn et al., 2011). The biological assumption is that such measures of irregularity and spreading of cells allow accurate classification and discrimination between cancer cell lines treated with different drugs (Elaheh et al., 2019). The submission studies to which extent this Riemannian metric can detect how the cell shape is associated with the response to treatment.

Repurposing Peptide Inhibitors for SARS-Cov-2 Spike Protein　This submission develops an approach combining physico-chemical parameter analysis and topological featurization to train robust one-class classifiers to predict protein-protein interactions (PPIs). PPIs form the molecular basis of processes that equally sustain life and drive development of disease, such as SARS-Cov-2. Peptides have garnered therapeutic interest due to their potential to disrupt clinically-relevant PPIs, apart from synthetic accessibility and better targeting modalities (Tsomaia, 2015; Mohapatra et al., 2020; Schissel et al., 2020). The submission uses the top-performing model to screen the peptides in the current dataset against SARS-Cov-2 receptor binding domain protein. The Peptide Binding DataBase (PepBDB) is used for model training (Wen et al., 2018).

Shape analysis with skeletal models and Principal Nested Spheres This submission considers anatomical shape analysis with skeletal representations (s-reps) (Liu et al., 2021) and Principal Nested Spheres (PNS) (Jung et al., 2012; Kim et al., 2020). The s-rep of a given shape consists of the shape's skeleton and two functions defined on the skeleton: a radial vector field and a radius function. PNS is a manifold learning method that addresses the non-Euclidean properties of shape data. PNS fits a *hierarchy* of submanifolds – subspheres – to some input data. The notebook applies this method to s-reps of toy data and to the classification problem of the hand skeleton shape dataset available in `Geomstats' datasets` module, comparing s-reps and PNS to Euclidean and Riemannian alternatives from the literature. The best classification performance is obtained by using the Kendall Riemmanian metric (Le & Kendall, 1993) on the hand skeleton shapes.

Riemannian mean-shift algorithm This submission implements a Riemannian version of the mean-shift algorithm (Subbarao & Meer, 2009; Caseiro et al., 2012). Classic (Euclidean) mean shift works by sliding a window (a ball whose radius is called "bandwidth") over the dataset, iteratively adjusting the center of the window until convergence to the estimated mode of the data. Mean shift is used for clustering, with several advantages over K-means. This notebook implements the method and shows its applicability on toy datasets on the sphere and hyperbolic plane.

Intrinsic Disease Maps Using Persistent Cohomology This submission uses persistent cohomology to investigate and visualize two infectious disease progression datasets: physiological data on Malaria in mice (Cumnock et al., 2018) and humans (Torres et al., 2016), and data on Hepatitis C in humans (Rosenberg et al., 2018). The submission reiterates the work of (Daniel Amin, 2021) and computes circular coordinates using the methodology introduced in (de Silva & Vejdemo-Johansson, 2009). The generated circular coordinate function provides an intrinsic disease phase coordinate that maps out the disease progression in the full data space.

Neural Sequence Distance Embeddings This submission presents Neural Sequence Distance Embeddings (`NeuroSEED`), a general framework to embed biological sequences in geometric vector spaces that reflect their evolutionary distance. The notebook illustrates the effectiveness of the hyperbolic space that captures the hierarchical structure and provides an average 38% reduction in embedding RMSE against the best competing geometry. The capacity of the framework and the significance of these improvements are then demonstrated devising supervised and unsupervised `NeuroSEED` approaches to multiple core tasks in bioinformatics. Benchmarked with common baselines, the proposed approaches display significant accuracy and/or runtime improvements on real-world datasets (Clemente et al., 2015; Zheng et al., 2019).

Analyzing Representative Cycles for Persistent Homology This submission aims to simplify the use of cycles for the analysis of persistent homology. The persistence diagram is often the only representation that software packages for TDA provide to visualize persistent homology information. Visualizing where each homology class appeared in the domain space can be very challenging for a user. This submission provides a collection of functions to simplify the interactive visualization and analysis of the homology class by enriching the information contained in the persistence diagram with cycles. Cycles are computed with an external library Iuricich (2020) which uses Discrete Morse theory Robins et al. (2011); Mischaikow & Nanda (2013) to achieve scalability. The notebook demonstrates the use of cycles on a subset of the MNIST dataset (LeCun et al., 2010) and provides an overview of applications using the direct visualization of cycles for exploratory data analysis.

Investigating CNN weights with Giotto Vectorization This submission provides a topological analysis of convolutional neural networks (CNN) weights. Transforming each layer to a new auxiliary space predicts network properties on a non-trivial supervised classification task. The notebook uses the Small CNN Zoo dataset (Unterthiner et al., 2021), shows how to compute the persistence diagrams of the auxiliary space, vectorizes the diagrams into Silhouettes (Chazal et al., 2014), and finally runs several regression and classification experiments. The results are particularly encouraging in terms of anomaly detection.

Brain Connectomes Comparison using Geodesic Distances This submission investigates the performance of geodesic distances on manifolds to assess brain connectome similarity between pairs

of twins in terms of their structural networks at different network resolutions. The notebook uses the brain structural connectomes of 412 human subjects in five different resolutions and two edge weights (Kerepesi et al., 2016). The notebook investigates the performance of geodesic distances on manifolds and compares them with Euclidean distances within a Wilcoxon rank sum non-parametric test (Cuzick, 1985).

Fuzzy c-Means Clustering for Persistence Diagrams and Riemannian Manifolds This submission implements Fuzzy c-Means clustering for persistence diagrams and Riemannian manifolds (Davies et al., 2020). Many real world problems are fuzzy; that is, data points can have partial membership to several clusters, rather than a single "hard" labelling to only one cluster (Campello, 2007). The notebook describes the fuzzy c-means algorithm, highlights the convergence results, and demonstrates fuzzy clustering on two simple datasets.

Reweighting Vectors for Graph Convolutional Neural Networks via Poincaré Embedding and Persistence Images This submission demonstrates how to incorporate local graph topological properties (e.g. connected components, cycles) into persistence enhanced graph neural networks (GNN) (Zhao et al., 2020) for graph and node classification tasks. The notebook converts unweighted graphs to weighted graphs by embedding them using the Poincaré ball model and using the resulting Riemannian distances as the weights. Then, the persistence images of resulting weighted graphs are computed and the resulting matrix is used to re-weight the GNN weights for enhanced performances.

4   FEATURES USED IN THE PACKAGES

This section presents the features used in both packages and the limitations outlined by the participants. The numbers in parentheses refer to the number of submissions in which a given feature has been used.

Features used in `Geomstats` The differential geometric structures used in the submissions are the following: `Hypersphere` ($\times 4$), Kendall's `PreShapeSpace` ($\times 3$) with associated `KendallShapeMetric` ($\times 3$), the space of symmetric positive definite matrices `SPDMatrices` ($\times 2$) with associated Bures-Wasserstein metric `SPDMetricBuresWasserstein` ($\times 2$), or alternative Riemannian metrics such as the `SPDMetricAffine` ($\times 1$) and the `SPDMetricLogEuclidean` ($\times 1$), the Lie group of rotations `SpecialOrthogonal` ($\times 2$), the hyperbolic space in its `PoincareBall` ($\times 2$) representation and associated `PoincareBallMetric` ($\times 1$), the space of `Matrices` ($\times 1$), the general class of `RiemannianMetric` ($\times 1$), the manifold of `DiscreteCurves` ($\times 1$) with associated square-root velocity metric `SRVMetric` ($\times 1$), the `Grassmanian` ($\times 1$) with the `GrassmanianCanonicalMetric` ($\times 1$), `Stiefel` ($\times 1$) with the `StiefelCanonicalMetric` ($\times 1$). The algorithms of geometric statistics that have been used are: the estimation of the `FrechetMean` ($\times 6$), the tangent principal component analysis `TangentPCA` ($\times 3$), the `RiemannianKMeans` and the Riemannian version of the `KNearest NeighbordsClassifier`. In terms of differential geometric datasets, the cell shapes dataset, hand skeleton shape dataset and the optical nerve shape dataset were used. The visualization module was also used through its `Sphere` ($\times 3$), `PoincareDisk` ($\times 2$), `KendallDisk` ($\times 1$), `KendallSphere` ($\times 1$). The main features that have not been used are the mathematical structures and functions related to information geometry, such as the manifolds of Beta distributions, Dirichlet distributions and Normal distributions; and the more involved learning procedures such as the `expectation_maximization` or the `KalmanFilter` on Lie groups.

Features used in `Giotto-TDA` The topological features used in the submissions are the following. `VietorisRipsPersistence` ($\times 6$) was the most used class for computing persistent homology, followed by `CubicalPersistence` ($\times 1$). In the `diagrams` module, the computation of distance matrices between persistence diagrams via `PairwiseDistance` ($\times 3$) was used but not preferred to the vectorisation of persistence diagrams via `PersistenceImage` ($\times 2$), `PersistenceEntropy` ($\times 1$), `NumberOfPoints` ($\times 1$), `Amplitude` ($\times 1$) and `Silhouette` ($\times 1$). Additionally, `Scaler` ($\times 1$), from the same module, was used. A few pre-processing utilities

for images were explored, namely `Binarizer` (×2) and the following classes for creating filtrations from 2D or 3D greyscale images: `RadialFiltration` (×2), `DilationFiltration` (×1), `ErosionFiltration` (×1) and `DensityFiltration` (×1). The visualization module `plotting` was used through its functions `plot_diagram` (×3) and `plot_point_cloud` (×2). Among the main modules that have not been used, we find the modules related to time-series and curves.

5   LIMITATIONS OF THE PACKAGES

Participants were asked to report on the limitations of the packages. This section provides a summary of their findings.

Limitations of `Geomstats`   First, some participants reported bugs.   For example, there was an issue to project points from the `Grassmannian` or the `Stiefel` manifold to the tangent space using default canonical metrics or the participants' custom metrics.   Both `preprocessing.ToTangentSpace` and `Grassmannian.to_tangent` failed with the same issue. A similar problem was encountered while trying to project points from the manifold of `DiscreteCurves` on its tangent space at the `FrechetMean` using the square-root velocity metric `SRVMetric`. In this case, the implementation of the `FrechetMean` itself was failing.

Second, participants did not find some implementations in the package or struggled to understand the existing code. For example, a participant reported that the metric for the `HyperbolicSpace` was missing, and another tried to use the abstract class `RiemannianMetric` without providing a definition of inner-product or a metric matrix.   Another participant would have liked to use product manifold and product metrics but did not realize that it was implemented in the library.   Other participants wanted to use several backends but could not find the way to use both in one script: `os.environ["GEOMSTATS_BACKEND"] = "pytorch"; importlib.reload(geomstats.backend)`.

These issues come from a lack of completeness of the current documentation of the package, misleading error messages and possible erroneous existing documentation. Other participants did report several problems with the current documentation, which could be improved with more detailed descriptions and an index with short summaries. There are classes such as the `PoincareBall` that are found in tutorials, but not find in the documentation website.

Lastly, participants reported the lack of integration between the modules related to graph and hyperbolic spaces in `Geomstats`, and the formalism of modern libraries using graphs. In this case, a refactoring is needed to allow a better integration between geometric statistics through `Geomstats` and packages of geometric learning such as `pytorch-geometric` (Fey & Lenssen, 2019), `networkx` (Hagberg et al., 2008) and `dgl` (Wang et al., 2019).

Limitations of `Giotto-TDA`   First, some participants reported bugs. For example, computing persistence images on persistence diagrams formed by a single persistence pair outputs an empty persistence image.

Some participants pointed out that the rigid input requirements for `PairwiseDistance` – in particular, the fact that all persistence diagrams must formally have the same number of homology dimensions and of birth-death pairs in each homology dimension – can be limiting in applications. This indicates that a utility function for converting collections of persistence diagrams into a format accepted by `PairwiseDistance` should be added to the package. Possibly related to this, some participants surmised (incorrectly, in this case) that persistent homology transformers such as `VietorisRipsPersistence` cannot handle collections of point clouds of different cardinalities. This might indicate that the aforementioned tight requirements in `PairwiseDistance` are at odds with the more permissive character of many other components of `Giotto-tda`.

The next reported limitation came from the architecture of the package itself.   Some participants reported that the package's high-level API did not allow for manipulations of attributes and usage of methods that are present in the objects that it runs underneath.   For example, `VietorisRipsPersistence` runs using `Ripser` but does not allow one to access the cocycles that are otherwise accessible using `Ripser`'s API directly.

Lastly, in terms of performance, some participants reported that the computational runtime for `PersistenceImage` was very irregular on their data in comparison to the performance of e.g. `Silhouette`.

6   PROPOSED FEATURES FOR THE PACKAGES

This section lists the features, suggested by the participants, that could be implemented in packages of computational geometry and topology such as `Geomstats` and `Giotto-TDA`. These are implementations that they would judge useful in order to push forwards the fields of computational geometry and topology.

Proposed features for `Geomstats`   As the Stiefel and the Grassmann manifolds are becoming popular in the Machine Learning and Computer Vision community, more geometric features on these manifolds (such as various metrics) could offer powerful tools for solving a wide range of learning tasks.

In the same vein, symmetric positive definite (SPD) matrices are raising more and more interest in the same communities. While `Geomstats` has a module that processes time-series into SPD matrices, further SPD-dedicated preprocessing to SPD matrices for various data types would be helpful. Trainable SPD representations would also be of interest. For instance, some participants specifically asked for an implementation of SPD neural networks, the so-called second order neural networks.

Then, the module on the geometry of discrete curves could be improved in different ways. `Geomstats` only provides spaces of open curves, a restriction which is also not necessary clear through reading the documentation only. Adding the implementation of the space of closed curves would be interesting. Furthermore, only the elastic metric on these spaces of curves is implemented, while generalizations of this metric exist that could be interesting. Lastly, when looking at shapes of curves, it is interesting to quotient out not only the reparameterization of the curve (done with the elastic metric), but also the rotations and translations of these curves. This is not obvious from the current documentation, and indications of this aspect, together with a recommendation to use the module of Kendall shape space for this, could be helpful.

Lastly, the visualization module could be further improved, by allowing more interactive visualizations and adding a visualization for the space of SPD matrices that could allow to visualize the differences between the different metrics on SPD matrices (of low dimensions).

Proposed features for `Giotto-TDA`   First, participants suggested to add an implementation that computes the pairwise distance matrix of diagrams with different number of homology groups in each dimension, without having to first perform some laborious manual padding. Some participants also offered to add tools to compute the cohomology persistence, and for example circular coordinates.

For machine learning applications, some participants suggested to use a tensor backend instead of NumPy, such as Tensorflow or PyTorch.

7   FINAL RANKING

This section provides the final ranking of the challenge's submissions. The Condorcet method was used to rank the submissions based on the single evaluation criterion: "how does the submission help push forward the fields of computational geometry and topology?"

Each of the 16 teams had the opportunity to vote for the 3 best submissions. Each team received only one vote, even if there were several participants in the team. In addition, 8 external reviewers, chosen among `Geomstats` and `Giotto-TDA` core maintainers and all from different institutions, also voted for the 3 best submissions. The 3 preferences had to be all 3 be different: e.g. one could not select the same `Jupyter Notebook` for both first and second place. The submissions were anonymized, the votes remained secret, only the final ranking is published here. Ties are represented by bullet points in the ranking below.

1. Noise Invariant topological features
2. • Estimators of Means of Symmetric Positive Matrices
   • Visualization of Kendall Shape Spaces for Triangles
3. Neural Sequence Distance Embeddings
4. Repurposing Peptide Inhibitors for SARS-Cov-2 Spike Protein
5. • Map your Topology to Different Geometries
   • Intrinsic Disease Maps Using Persistent Cohomology
6. Shape analysis with skeletal models and Principal Nested Spheres
7. • Riemannian mean-shift algorithm
   • Fuzzy c-Means Clustering for Persistence Diagrams and Riemannian Manifolds
8. • Naive Image Anomaly Detection on Fashion MNIST
   • Shape Analysis of Bone Cancer Cells
   • Brain Connectomes Comparison using Geodesic Distances
   • Investigating CNN weights with Giotto Vectorization
9. Analyzing Representative Cycles for Persistent Homology
10. Reweighting Vectors for Graph Convolutional Neural Networks via Poincaré Embedding and Persistence Images

Regardless of this final ranking, we would like to stress that all the submissions were of very high quality. We warmly congratulate all the participants.

AUTHOR CONTRIBUTIONS

ACKNOWLEDGMENTS

8 CONCLUSION

This white paper presented the motivations behind the organization of the "Computational Geometric and Topological Challenge" at the ICLR 2021 workshop "Geometric and Topological Representation Learning", and summarized the findings from the participants' submissions.

REFERENCES

Seth Axen, Mateusz Baran, and Ronny Bergmann. Manifolds.jl: a library of Riemannian manifolds in Julia, 2021. URL https://github.com/JuliaManifolds/Manifolds.jl.

Alexandre Barachant. PyRiemann: Python package for covariance matrices manipulation and Biosignal classification with application in Brain Computer interface, 2015. URL https://github.com/alexandrebarachant/pyRiemann.

Ulrich Bauer. Ripster: efficient computation of vietoris-rips persistence barcodes, February 2021. Preprint.

Ulrich Bauer, Michael Kerber, Jan Reininghaus, Hubert Wagner, and Bryn Keller. Phat: Persistent homology algorithm toolbox, 2017. URL https://bitbucket.org/phat-code/phat/src/master/.

Nicolas Boumal, Bamdev Mishra, P. A. Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2013. ISSN 15337928. doi: 10.1080/13583883.2004.9967123. URL http://arxiv.org/abs/1308.5200.

Piotr Brendel, Paweł Dłotko, Grzegorz Jabłoński, Mateusz Juda, Andrzej Krajniak, Marian Mrozek, Hubert Wagner, Przemysław Witek, and Natalia Żelazna. Capd::redhom - simplicical and cubical homology., 2019. URL `http://capd.sourceforge.net/capdRedHom/`.

Miquel Calvo and Josep M Oller. A distance between multivariate normal distributions based in an embedding into the siegel group. *Journal of multivariate analysis*, 35(2):223–242, 1990.

Ricardo José Gabrielli Barreto Campello. A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recogn. Lett.*, 28(7):833–841, May 2007. ISSN 0167-8655. doi: 10.1016/j.patrec.2006.11.010. URL `https://doi.org/10.1016/j.patrec.2006.11.010`.

Rui Caseiro, João F. Henriques, Pedro Martins, and Jorge Batista. Semi-intrinsic mean shift on riemannian manifolds. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (eds.), *Computer Vision – ECCV 2012*, pp. 342–355, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33718-5.

Andrea Censi. PyGeometry: Library for handling various differentiable manifolds., 2012. URL `https://github.com/AndreaCensi/geometry`.

Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, and Larry Wasserman. Stochastic convergence of persistence landscapes and silhouettes. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, pp. 474–483, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450325943. doi: 10.1145/2582112.2582128. URL `https://doi.org/10.1145/2582112.2582128`.

Jose Clemente, Erica Pehrsson, Martin Blaser, Kuldip Sandhu, Zhan Gao, Bin Wang, Magda Magris, Glida Hidalgo, Monica Contreras, Óscar Noya-Alarcón, et al. The microbiome of uncontacted amerindians. *Science advances*, 2015.

Princeton Course COS429, 2009. URL `https://www.cs.princeton.edu/courses/archive/fall09/cos429/assignment3.html`.

Katherine Cumnock, Avni S. Gupta, Michelle Lissner, Victoria Chevee, Nicole M. Davis, and David S. Schneider. Host energy source is important for disease tolerance to malaria. *Current Biology*, 28(10):1635–1642.e3, 2018. ISSN 0960-9822. doi: https://doi.org/10.1016/j.cub.2018.04.009. URL `https://www.sciencedirect.com/science/article/pii/S0960982218304378`.

Jack Cuzick. A wilcoxon-type test for trend. *Statistics in Medicine*, 4(4):543–547, 1985. doi: https://doi.org/10.1002/sim.4780040416. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780040416`.

Olivier Dalle. On reproducibility and traceability of simulations. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pp. 1–12, 2012. doi: 10.1109/WSC.2012.6465284.

Mikael Vejdemo-Johansson Daniel Amin. Intrinsic disease maps using persistent cohomology. *Foundations of Data Science*, 0, 2021.

A. David Redish, Erich Kummerfeld, Rebecca Lea Morris, and Alan C. Love. Reproducibility failures are essential to scientific inquiry. *Proceedings of the National Academy of Sciences of the United States of America*, 115(20):5042–5046, 2018. ISSN 10916490. doi: 10.1073/pnas.1806370115.

Thomas Davies, Jack Aspinall, Bryan Wilder, and Long Tran-Thanh. Fuzzy c-means clustering for persistence diagrams. *CoRR*, abs/2006.02796, 2020. URL `https://arxiv.org/abs/2006.02796`.

Vin de Silva and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Proceedings of the 25th annual symposium on Computational geometry - SCG '09*, 2009. doi: 10.1145/1542362.1542406. URL `http://dx.doi.org/10.1145/1542362.1542406`.

The Sage Developers, William Stein, David Joyner, David Kohel, John Cremona, and Burçin Eröcal. Sagemath, version 9.0, 2020. URL `http://www.sagemath.org`.

Ian Dryden and Kantilal Mardia. *Statistical Shape Analysis, with Applications in R. Second Edition.* John Wiley and Sons, Chichester, 2016.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, April 1999. ISSN 0895-4798. doi: 10. 1137/S0895479895290954. URL https://doi.org/10.1137/S0895479895290954.

Alizadeh Elaheh, Xu Wenlong, Castle Jordan, Foss Jacqueline, and Prasad Ashok. Tismorph: A tool to quantify texture, irregularity and spreading of single cells. *PLoS ONE*, 14(6), 2019. URL https://doi.org/10.1371/journal.pone.0217346.

Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the r package tda, 2015.

Matthias Fey and Jan Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1553–1563. PMLR, 26–28 Aug 2020. URL http://proceedings.mlr.press/v108/gabrielsson20a.html.

GeomData. Gda-public: Open-source toolbox of easy-to-use topological data analysis tools., 2021. URL https://github.com/geomdata/gda-public.

Aric Hagberg, Daniel Schult, and Pieter Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

Jihun Hamm and Daniel Lee. Extended grassmann kernels for subspace-based learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2008/file/e7f8a7fb0b77bcb3b283af5be021448f-Paper.pdf.

Gregory Henselman and Robert Ghrist. Matroid Filtrations and Computational Persistent Homology. *ArXiv e-prints*, June 2016.

Jeffrey Ho, Guang Cheng, Hesamoddin Salehian, and Baba Vemuri. Recursive karcher expectation estimators and geometric law of large numbers. In Carlos M. Carvalho and Pradeep Ravikumar (eds.), *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 325–332, Scottsdale, Arizona, USA, 05 2013. PMLR. URL http://proceedings.mlr.press/v31/ho13a.html.

Federico Iuricich. PersistenceCycles: a C++ package for visual exploration of persistent homology, 2020. URL https://github.com/IuricichF/PersistenceCycles.

Ian Jermyn, Sarang Joshi, Eric Klassen, and Anuj Srivastava. Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(07):1415–1428, 07 2011. ISSN 1939-3539. doi: 10.1109/TPAMI.2010.184.

Sungkyu Jung, Ian Dryden, and James Stephen Marron. Analysis of principal nested spheres. *Biometrika*, 99(3):551–568, 2012.

David G. Kendall. Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 03 1984. ISSN 0024-6093. doi: 10.1112/blms/16.2.81. URL https://doi.org/10.1112/blms/16.2.81.

Michael Kerber, Jan Reininghaus, and Ulrich Bauer. Dipha: A distributed persistent homology algorithm, 2014. URL https://github.com/DIPHA/dipha.

Csaba Kerepesi, Balazs Szalkai, Bálint Varga, and Vince Grolmusz. The braingraph.org database of high resolution structural connectomes and the brain graph tools. *Cognitive Neurodynamics*, 11, 10 2016. doi: 10.1007/s11571-017-9445-1.

Byungwon Kim, Jörn Schulz, and Sungkyu Jung. Kurtosis test of modality for rotationally symmetric distributions on hyperspheres. *Journal of Multivariate Analysis*, 178(C), 2020. doi: 10.1016/j.jmva.2020.10460. URL `https://ideas.repec.org/a/eee/jmvana/v178y2020ics0047259x19306141.html`.

Maxim Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian Adaptive Optimization Methods with pytorch optim, 2019. URL `https://github.com/geoopt/geoopt`.

Line Kühnel and Stefan Sommer. Computational Anatomy in Theano. In *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*, pp. 164–176, Cham, 2017. Springer International Publishing. URL `https://bitbucket.org/stefansommer/theanogeometry`.

Huiling Le and David G. Kendall. The riemannian structure of euclidean shape spaces: A novel environment for statistics. *The Annals of Statistics*, 21(3):1225–1271, 1993. ISSN 00905364. URL `http://www.jstor.org/stable/2242196`.

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

Zhiyuan Liu, Jun-Pyo Hong, Jared Vicory, James N. Damon, and Stephen M. Pizer. Fitting unbranching skeletal structures to objects. *Medical Image Anal.*, 70:102020, 2021. doi: 10.1016/j.media.2021.102020. URL `https://doi.org/10.1016/j.media.2021.102020`.

Nina Miolane, Nicolas Guigui, Alice Le Brigant, Johan Mathe, Benjamin Hou, Yann Thanwerdas, Stefan Heyder, Olivier Peltre, Niklas Koep, Hadi Zaatiti, Hatem Hajri, Yann Cabanes, Thomas Gerald, Paul Chauchat, Christian Shewmake, Daniel Brooks, Bernhard Kainz, Claire Donnat, Susan Holmes, and Xavier Pennec. Geomstats: A python package for riemannian geometry in machine learning. *Journal of Machine Learning Research*, 21:1–9, 2020. ISSN 15337928.

Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, September 2013. ISSN 0179-5376, 1432-0444. doi: 10.1007/s00454-013-9529-6.

Somesh Mohapatra, Nina Hartrampf, Mackenzie Poskus, Andrei Loas, Rafael Gómez-Bombarelli, and Bradley L Pentelute. Deep Learning for Prediction and Optimization of Fast-Flow Peptide Synthesis. *ACS Central Science*, nov 2020. ISSN 2374-7943. doi: 10.1021/acscentsci.0c00979. URL `https://doi.org/10.1021/acscentsci.0c00979`.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7045–7054. PMLR, 07 2020. URL `http://proceedings.mlr.press/v119/moor20a.html`.

Dmitriy Morozov. Dionysus, a c++ library for computing persistent homology., 2015. URL `https://www.mrzv.org/software/dionysus/`.

Vidit Nanda. Perseus, the persistent homology software, 2012. URL `http://www.sas.upenn.edu/~vnanda/perseus`.

Bentley James Oakes, Romain Franceschini, Simon Van Mierlo, and Hans Vangheluwe. The computational notebook paradigm for multi-paradigm modeling. *Proceedings - 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion, MODELS-C 2019*, pp. 449–454, 2019. doi: 10.1109/MODELS-C.2019.00072.

Ippei Obayashi, Tomohide Wada, Tian Tunhua, Jun Miyanaga, and Yasuaki Hiraoka. Homcloud - a data analysis software based on persistent homology., 2020. URL `https://homcloud.dev/index.en.html`.

Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pp. 6726–6733. IEEE, 2020. doi: 10.1109/ICPR48806.2021.9412109. URL `https://doi.org/10.1109/ICPR48806.2021.9412109`.

Vanessa Robins and Olaf Delgado-Friedrichs. Diamorse - digital image analysis using discrete morse theory and persistent homology., 2020. URL `https://github.com/AppliedMathematicsANU/diamorse`.

Vanessa Robins, Peter John Wood, and Adrian P. Sheppard. Theory and algorithms for constructing discrete morse complexes from grayscale digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1646–1658, 2011. doi: 10.1109/TPAMI.2011.95.

Brad Rosenberg, Marion Depla, Catherine A. Freije, Denis Gaucher, Sabrina Mazouz, Maude Boisvert, Nathalie Bédard, Julie Bruneau, Charles Rice, and Naglaa Shoukry. Longitudinal transcriptomic characterization of the immune response to acute hepatitis c virus infection in patients with spontaneous viral clearance. *PLOS Pathogens*, 14(9):1–24, 09 2018. doi: 10.1371/journal.ppat.1007290. URL `https://doi.org/10.1371/journal.ppat.1007290`.

Nathaniel Saul and Chris Tralie. Scikit-tda: Topological data analysis for python, 2019. URL `https://doi.org/10.5281/zenodo.2533369`.

Carly Schissel, Somesh Mohapatra, Justin Wolfe, Colin Fadzen, Kamela Bellovoda, Chia-Ling Wu, Jenna Wood, Annika Malmberg, Andrei Loas, Rafael Gómez-Bombarelli, and Bradley Pentelute. Interpretable Deep Learning for De Novo Design of Cell-Penetrating Abiotic Polymers. *bioRxiv*, 2020. doi: 10.1101/2020.04.10.036566. URL `https://www.biorxiv.org/content/10.1101/2020.04.07.029884v1`.

Santiago Schnell. "Reproducible" Research in Mathematical Sciences Requires Changes in our Peer Review Culture and Modernization of our Current Publication Approach. *Bulletin of Mathematical Biology*, 80(12):3095–3105, 2018. ISSN 15229602. doi: 10.1007/s11538-018-0500-9.

Armin Schwartzman. Lognormal distributions and geometric averages of symmetric positive definite matrices. *International Statistical Review*, 84(3):456–86, 2016. doi: 10.1111/insr.12113.

Oleg Smirnov. Tensorflow manopt: a library for optimization on riemannian manifolds, 2021.

Anirudh Som, Kowshik Thopalli, Karthikeyan Natesan Ramamurthy, Vinay Venkataraman, Ankita Shukla, and Pavan Turaga. Perturbation robust representations of topological persistence diagrams. In Vittorio Ferrari, Cristian Sminchisescu, Martial Hebert, and Yair Weiss (eds.), *Computer Vision – ECCV 2018 - 15th European Conference, 2018, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 638–659. Springer Verlag, 2018. ISBN 9783030012335. doi: 10.1007/978-3-030-01234-2_38. Funding Information: Acknowledgments. This work was supported in part by ARO grant W911NF-17-1-0293 and NSF CAREER award 1452163. Publisher Copyright: © Springer Nature Switzerland AG 2018.; 15th European Conference on Computer Vision, ECCV 2018 ; Conference date: 08-09-2018 Through 14-09-2018.

Victoria Stodden, David Bailey, and Jonathan Borwein. Setting the Default to Reproducible, 2013. URL `http://stodden.net/icerm{_}report.pdf`.

Raghav Subbarao and Peter Meer. Nonlinear mean shift over riemannian manifolds, 2009.

Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. JavaPlex: A research software package for persistent (co)homology. In Han Hong and Chee Yap (eds.), *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, pp. 129–136, 2014. Software available at `http://appliedtopology.github.io/javaplex/`.

Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Anibal Medina-Mardones, Alberto Dassatti, and Kathryn Hess. giotto-tda: A topological data analysis toolkit for machine learning and data exploration, 2020.

The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.4.1 edition, 2021. URL `https://gudhi.inria.fr/doc/3.4.1/`.

Julien Tierny, Guillaume Favelier, Joshua A. Levine, Charles Gueunet, and Michael Michaux. The Topology ToolKit. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):832–842, January 2018. ISSN 1077-2626. doi: 10.1109/TVCG.2017.2743938.

Brenda Torres, Jose Henrique Oliveira, Ann Thomas Tate, Poonam Rath, Katherine Cumnock, and David Schneider. Tracking resilience to infections by mapping disease space. *PLOS Biology*, 14 (4):1–19, 04 2016. doi: 10.1371/journal.pbio.1002436. URL https://doi.org/10.1371/journal.pbio.1002436.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A Python Toolbox for Manifold Optimization using Automatic Differentiation. *arXiv preprint arXiv:1603.03236*, 17:1–4, 2016. ISSN 15337928. URL http://arxiv.org/abs/1603.03236.

Natia Tsomaia. Peptide therapeutics: targeting the undruggable space. *European journal of medicinal chemistry*, 94:459–470, 04 2015. ISSN 0223-5234. doi: 10.1016/j.ejmech.2015.01.014. URL https://doi.org/10.1016/j.ejmech.2015.01.014.

Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya Tolstikhin. Predicting neural network accuracy from weights, 2021.

Remco Veltkamp, Geert-Jan Giezeman, Hannah Bast, Thomas Baumbach, Takahiko Furuya, Joachim Giesen, Afzal Godil, Zhouhui Lian, Ryutarou Ohbuchi, and Waqar Saleem. SHREC'10 Track: Large Scale Retrieval. In Mohamed Daoudi and Tobias Schreck (eds.), *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2010. ISBN 978-3-905674-22-4. doi: 10.2312/3DOR/3DOR10/063-069.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Zeyu Wen, Jiahua He, Huanyu Tao, and Sheng-You Huang. PepBDB: a comprehensive structural database of biological peptide–protein interactions. *Bioinformatics*, 35(1):175–177, 07 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty579. URL https://doi.org/10.1093/bioinformatics/bty579.

Kieran Wynn. PyQuaternions: A fully featured, pythonic library for representing and using quaternions, 2014. URL https://github.com/KieranWynn/pyquaternion.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL http://arxiv.org/abs/1708.07747. cite arxiv:1708.07747Comment: Dataset is freely available at https://github.com/zalandoresearch/fashion-mnist Benchmark is available at http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/.

Chun-Hao Yang, Hani Doss, and Baba C. Vemuri. An empirical bayes approach to shrinkage estimation on the manifold of symmetric positive-definite matrices, 2020.

Qi Zhao, Ze Ye, Chao Chen, and Yusu Wang. Persistence enhanced graph neural network. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2896–2906. PMLR, 8 2020. URL http://proceedings.mlr.press/v108/zhao20d.html.

Wei Zheng, Le Yang, Robert Genco, Jean Wactawski-Wende, Michael Buck, and Yijun Sun. SENSE: Siamese neural network for sequence embedding and alignment-free comparison. *Bioinformatics*, 2019.