Learning Locally Adaptive Metrics that Enhance Structural Representation with LAMINAR

Christian Kleiber, **William H. Oliver** and **Tobias Buck**
Interdisciplinary Center for Scientific Computing, University of Heidelberg
Im Neuenheimer Feld 205, D-69120 Heidelberg, Germany

Abstract

We present `LAMINAR`, a novel unsupervised machine learning pipeline designed to enhance the representation of structure within data via producing a more-informative distance metric. Analysis methods in the physical sciences often rely on standard metrics to define geometric relationships in data, which may fail to capture the underlying structure of complex data sets. `LAMINAR` addresses this by using a continuous-normalising-flow and inverse-transform-sampling to define a Riemannian manifold in the data space without the need for the user to specify a metric over the data a-priori. The result is a locally-adaptive-metric that produces structurally-informative density-based distances. We demonstrate the utility of `LAMINAR` by comparing its output to the Euclidean metric for structured data sets.

1 Motivation and related work

Much of the analysis performed within the physical sciences depends heavily upon the geometric properties of the data used. Typically it is assumed that the Euclidean metric is best to describe this geometry. However, in many settings more information can be gleaned from the distribution of the data itself. This process is often referred to as metric learning and generally aims to define a Riemannian metric in the data space such that the structure of the data set is respected and better preserved within downstream analyses. Compared to a global metric, this would suggest that the distances resulting from a more-informative metric should be smaller between data points that exist within the same mode of the data and larger for points belonging to separate modes. It then follows that a density-based metric could be more-informative.

The idea of density-based metrics is not new. Bousquet et al. [2] posit that a density-based metric can be used to acquire more-informative distances for semi-supervised learning. This is extended by Sajama and Orlitsky [9], who find geodesic shortest-paths in a kernel density weighted nearest neighbour graph. The work has since continued with density approximations [1], proofs of convergences [4, 7], and practical implementations [6, 8, 13]. A recent work [12] has shown that although proofs of convergences exist for these implementations, in practice these methods do not generally converge to a ground-truth density-based geodesic. This same work also suggests a new normalising-flow- and score-model-based method of finding a density-based metric, which is shown to converge in practice.

The irony of these approaches and of the nature of defining a more-informative metric is that a meaningful metric is typically already needed to define the meaning of *local* and therefore density. We are only aware of one such class of approaches [10, 11] that does not assume a metric a-priori – which use a scale-invariant entropy-based method to define a meaningful locally-adaptive-metric for the purpose of enhancing structural representation in cosmological simulations. In this work, we seek to improve upon these implementations by utilising inverse-transform-sampling and continuous-normalising-flows to develop an unsupervised locally-adaptive-metric (LAM) algorithm that enhances structural representation and that may be applied within the broader physical sciences.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 LAMINAR

Our approach finds **L**ocally **A**daptive **M**etrics using **I**nvertible **N**etworks on **A**nisotropic **R**egions (`LAMINAR`) and calculates structurally-representative distances between any two points from an input data set. To do this, `LAMINAR` first transports the d-dimensional input data to a uniform distribution within the volume of the d-dimensional unit sphere. This transformation behaves like a cumulative distribution function (i.e. a pseudo-cdf) in the sense that it is uniform and that locality is preserved. Therefore, by then defining a Euclidean k-nearest-neighbour graph in the pseudo-cdf, `LAMINAR` attains an inverse-transform sampling that connects each point to a structurally-representative neighbourhood of the data space. `LAMINAR` then computes neighbour-neighbour distances (edge-weights) with a LAM (where the metric tensor varies according to the Jacobian of the data $\mapsto$ pseudo-cdf transformation) and longer-range distances as the sum of edge-weights belonging to shortest-paths within the graph. This process is unsupervised, density-based, and does not require a metric to be defined over the data space a-priori. The following subsections contain additional details of the implementation.

Transforming the input data The core of `LAMINAR` consists of a continuous planar flow as introduced in [3]. Briefly summarised, a (discrete) planar flow is described by

$$\mathbf{z}(t+1) = \mathbf{z}(t) + uh(w^{\mathrm{T}}\mathbf{z}(t) + b), \;\; \log\left(p(\mathbf{z}(t+1))\right) = \log\left(p(\mathbf{z}(t))\right) - \log\left|1 + u^{\mathrm{T}}\frac{\partial h}{\partial \mathbf{z}}\right|. \quad (1)$$

Note that u, $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are adjustable parameters, h an activation function, t describes an arbitrary time scale, and $z(t)$ is a random variable distributed according to $p(z(t))$. Using the parallels of this ResNet structure to the Euler discretisation of continuous transformations, and the instantaneous change of variables, the continuous planar flow is then given by

$$\frac{d\mathbf{z}(t)}{dt} = uh(w^{\mathrm{T}}\mathbf{z}(t) + b), \;\; \frac{\partial \log\left(p(\mathbf{z}(t))\right)}{\partial t} = -u^{\mathrm{T}}\frac{\partial h}{\partial \mathbf{z}(t)}. \quad (2)$$

Here, h is an activation function adhering to Lipschitz continuity and u, w and b are time-dependent, learnable parameters, given as an output of a hypernetwork (a fully connected MLP with a single hidden layer and only time as an input). This combined ODE can be solved using the given initial distribution, $p(\mathbf{z}(0))$, and since the result, $p(\mathbf{z}(t))$, can be evaluated as having been drawn from a probability distribution – as such the training is guided by the log-likelihood loss. Once `LAMINAR` has finished the training, the ODE will describe a continuous transformation from any d-dimensional initial distribution to a d-dimensional standard-normal distribution. `LAMINAR` then further transforms this multivariate Gaussian into a uniform distribution within the d-dimensional unit sphere, by adjusting the radius of each point according to

$$\mathbf{r}_{\text{sphere}} = \frac{\mathbf{r}_{\text{gaussian}}}{|\mathbf{r}_{\text{gaussian}}|} \cdot F(\mathbf{r}_{\text{gaussian}}), \text{ with } F(\mathbf{r}) = \left(1 - \frac{\Gamma(\frac{d}{2}, \frac{\mathbf{r}^2}{2})}{\Gamma(\frac{d}{2})}\right)^{\frac{1}{d}}, \quad (3)$$

where $F(\cdot)$ is the CDF of the multivariate Gaussian and $\Gamma(\cdot,\cdot)$ is the upper incomplete Gamma function.

Calculating distances Once `LAMINAR` has transformed the data, the k-nearest-neighbours of each point, $N_k(\mathbf{x}_i)$, are found from the pseudo-cdf using a KDTree and a Euclidean-metric. These neighbours can be understood as defining connections in an adjacency matrix in the original data space, whose weights are Mahalanobis distances (analogous to Sharma and Johnston [10]):

$$s^2(\mathbf{x}_i, \mathbf{x}_j) = |\boldsymbol{\Sigma}(\mathbf{x}_i, \mathbf{x}_j)|^{1/d} \cdot (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \cdot \boldsymbol{\Sigma}(\mathbf{x}_i, \mathbf{x}_j)^{-1} \cdot (\mathbf{x}_i - \mathbf{x}_j), \quad (4)$$

where $\boldsymbol{\Sigma}(\mathbf{x}_i, \mathbf{x}_j) = 0.5\,[\boldsymbol{\Sigma}(\mathbf{x}_i)) + \boldsymbol{\Sigma}(\mathbf{x}_j)]$ is the mean metric tensor of the two queried points e.g. neighbours in the adjacency matrix. Here, $\boldsymbol{\Sigma}$ is defined using the Jacobian of the full transformation such that

$$\boldsymbol{\Sigma} = (\mathbf{J}_{\text{total}}^{\mathrm{T}} \cdot \mathbf{J}_{\text{total}})^{-1}, \text{ with } \mathbf{J}_{\text{total}} = \mathbf{J}_{\text{to sphere}} \cdot \mathbf{J}_{\text{flow}}. \quad (5)$$

For completeness, $\mathbf{J}_{\text{flow}}$ is calculated using PyTorch's autograd function, while $\mathbf{J}_{\text{to sphere}}$ is calculated analytically using Eq. 3 – and both are evaluated at the position of each point, $\mathbf{x}_i$. `LAMINAR` can then calculate the distance between any two points of the input data, d_{ij}, via Dijkstras's algorithm [5] – which searches for the shortest-path within the graph defined by the aforementioned adjacency matrix filled with the new local distances as in Eq. 4. For clarity; given any two points, e.g. $\mathbf{x}_i$ and $\mathbf{x}_j$, if $\mathbf{x}_i \in N_k(\mathbf{x}_j)$ or $\mathbf{x}_j \in N_k(\mathbf{x}_i)$, then $d_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$ from Eq. 4.

3 Visualisation of the LAMINAR metric

It is difficult to quantitatively assess the `LAMINAR`'s performance since this requires measuring the degree of structural meaningfulness produced in the resultant metric. Nevertheless, we can assess its ability to match a set of *ground-truth* metrics determined analytically from simple transformations of a uniform circle (where the Euclidean metric is structurally meaningful). Although, we don't necessarily expect `LAMINAR` to match these metrics perfectly as there are an infinity of ways that each data point could be transformed while still producing the same final distribution (consider applying transformation $T(\mathbf{x})$ vs. applying $T(\mathbf{Rx})$ where $\mathbf{R}$ is some rotation). Hence with this analysis we assess whether LAMINAR does what we have claimed it to do and begin to demystify its behaviour.

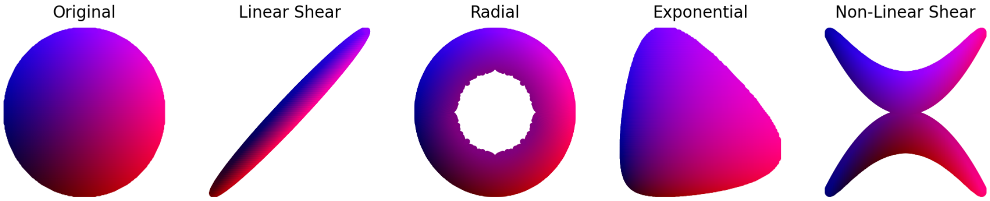

Figure 1: The original (uniform) distribution and the transformations applied to it.

Fig. 1 depicts four transformations while for each, Fig. 2 shows the ground-truth metric, the metric predicted by `LAMINAR`, and the point-wise difference between these metrics (measured with the Wasserstein distance between multivariate normal distributions whose covariance matrices are the metric tensors). The metrics are visualised according to the colour scheme described with Fig. 3.

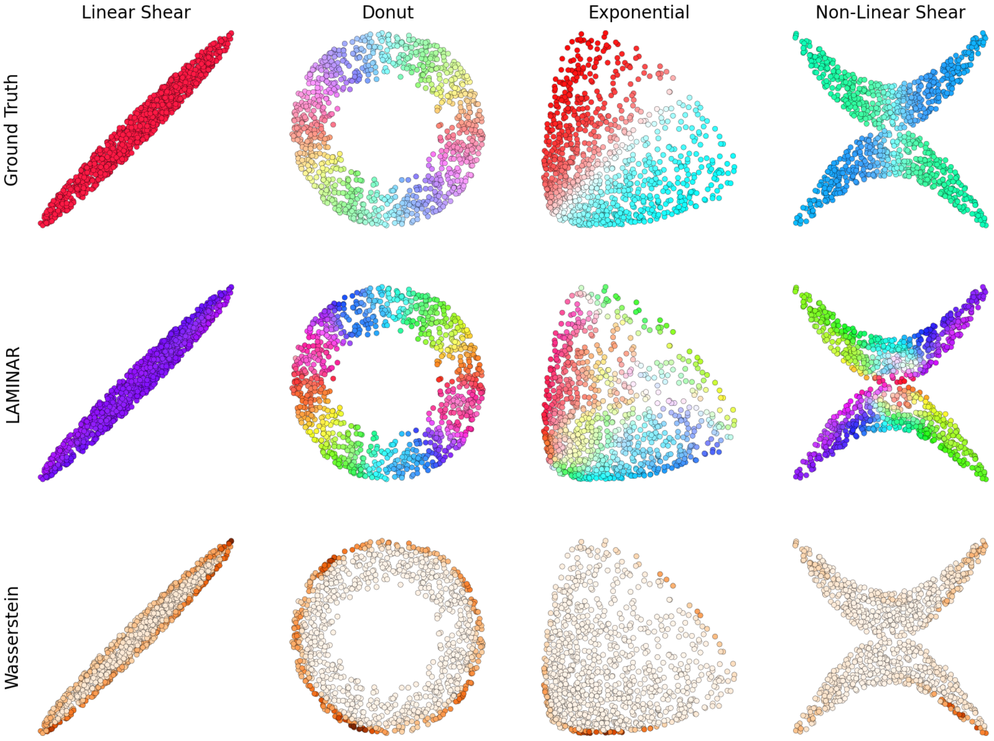

Figure 2: Comparison of *ground-truth* and `LAMINAR` metric tensors produced using data in Fig. 1.

In Fig. 2 we see that while there is some discrepancy due to noise as well as due to edge effects (related to the boundary of the pseudo-cdf), `LAMINAR` is able to learn the correct metric having only seen the transformed data set. It is worth noting that in the case of the linear shear transformation, the metric predicted by `LAMINAR` is seemingly a better representation of the global structure than that provided by the *ground-truth* – the *ground-truth* here results from the data experiencing a y-dependent translation while the `LAMINAR` metric corresponds to the global Mahalanobis distance one can recover by calculating this data set's covariance matrix. Still, these results suggest that `LAMINAR` would likely benefit from using an optimal-transport-based flow.

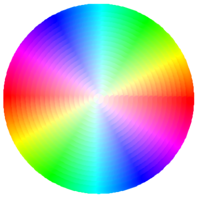

Figure 3: A reference colour wheel for the visualisation of the metric tensor in Fig. 2. To assign a colour to a data point, we first create an ellipse by transforming a circle with that point's metric tensor. The colour (angle) assigned is given by the orientation of this ellipse, i.e. red (blue) if its major axis aligns horizontally (vertically). The saturation (radius) of this colour is determined by the ratio between the lengths of the major and minor axes – so that a more spherical ellipse is lighter in colour. Visualising the metric in this way shows us the direction in, and degree to, which the distance function increases most (from each data point).

4 Direct comparison between the Euclidean metric and LAMINAR

We now conduct a qualitative check to see how `LAMINAR` compares to the Euclidean metric globally. Fig. 4 shows a few toy data sets coloured according to their distance from a query point, where brighter (darker) colours indicate smaller (larger) distances. The shortest-paths found by `LAMINAR` prefer dense regions, adapting to the data, while naturally those from the Euclidean metric are data-independent. The contours show an approximation of how the distance may look for points beyond the data set – estimated as the average distance-value of the $k = 25$ nearest neighbours.

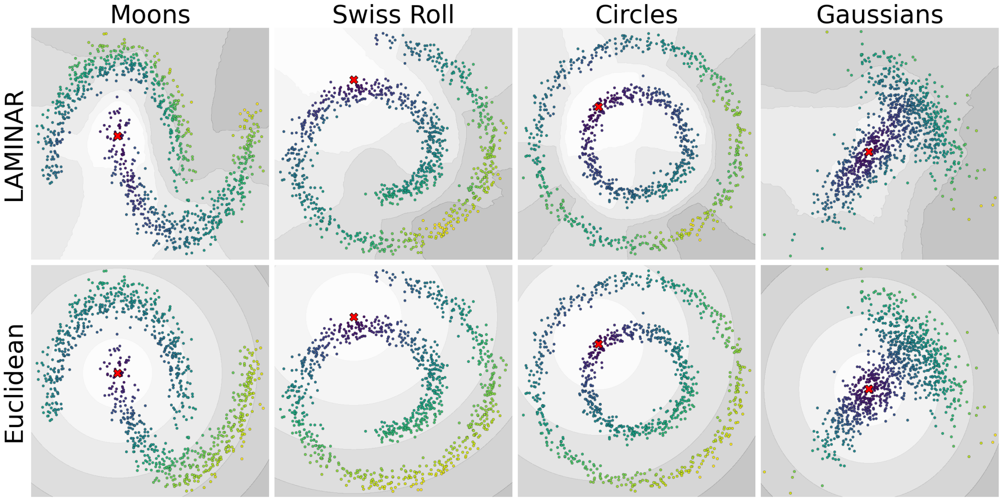

Figure 4: The distance distribution from a query point (red cross) on four toy data sets found with the `LAMINAR` (top) and Euclidean (bottom) metrics. The points are coloured according to the viridis colour map. The grey-scale contours show an estimate for the out-of-distribution distances.

Fig. 5 illustrates a more direct comparison of the Euclidean and `LAMINAR` metrics. For each metric, we take the logarithm of the distances shown in Fig. 4 and standardise each by subtracting the mean and dividing by the standard deviation – these values for each metric are then subtracted from one another. The resultant quantity represents a ratio of distances, and while the exact value is not meaningful, it portrays how the `LAMINAR` metric behaves relative to the Euclidean one. Again, we see that moving along the modes of the data is preferred (discouraged) by the `LAMINAR` (Euclidean) metric, while moving perpendicular to the data is treated oppositely. As such, it is clear that `LAMINAR` is able to learn and emphasise the structure implicit to the data sets.

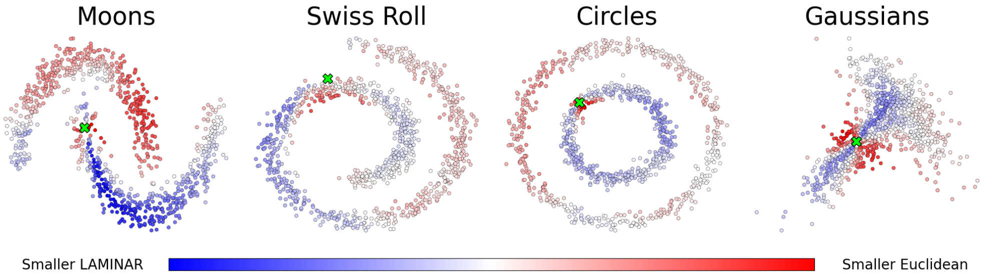

Figure 5: A comparison of distances from query points (marked with a green cross) produced via the `LAMINAR` and Euclidean metrics (as shown in Fig 4). Here blue points mark those with smaller `LAMINAR` distances compared to their Euclidean counterparts, and *vice versa* for red points.

5 Usage in downstream analysis

To demonstrate the advantage of LAMINAR in downstream analyses, an example is provided by using LAMINAR-calculated distances in the k-medoids algorithm for clustering. These results are compared to the same for Euclidean distances, i.e. the standard metric used in such algorithms. k-medoids is chosen as it allows clustering without deviating from the data points (in contrast to k-means) – since this early version of LAMINAR is not able to calculate distances between out-of-distribution points and is instead limited to distances between existing data points. Additionally, k-medoids is a very simple algorithm, so the choice of metric has a large impact on its performance. In contrast, it is possible that other more complicated clustering algorithms may not see as much improvement due to a focus on optimizations using the standard metric. These results are shown in Fig. 6.

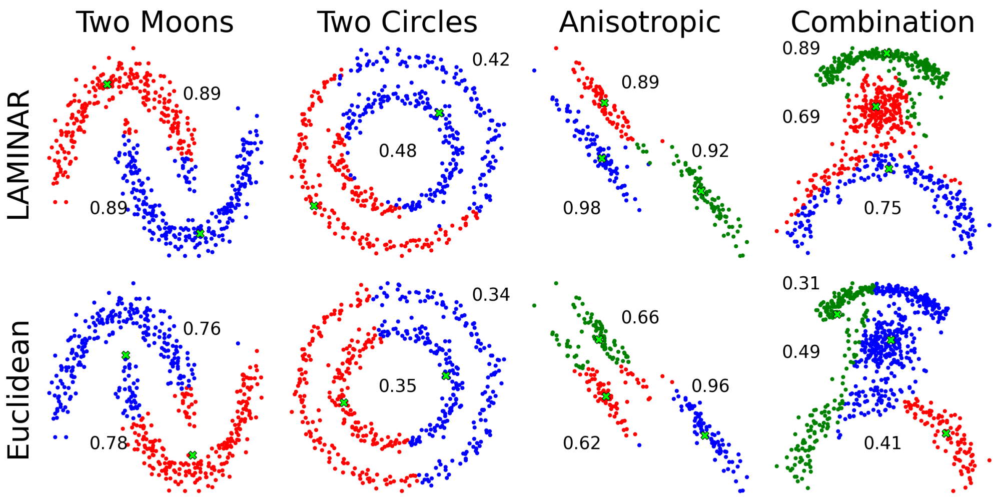

Figure 6: Comparison of the LAMINAR (top) and Euclidean (bottom) metrics in the downstream clustering tool, k-medoids. Values shown next to the clusters are Jaccard similarities between the ground-truth clusters and the best-fitting predicted cluster (colours).

Even though the ground-truth clusters are visibly apparent, they pose significant problems for k-medoids. We assess the effect of the different metrics by calculating the Jaccard index between each of the ground-truth clusters and the best-fitting predicted cluster from each of the metrics. Here, a value of 1 shows perfect agreement and flawless reconstruction of the cluster while 0 shows no overlap of the two clusters. We can see from Fig. 6 that the effect of the LAMINAR metric is to improve cluster extraction for (almost) all cases – in certain cases almost achieving a perfect reconstruction. This demonstrates the usability of LAMINAR in downstream analysis for enhanced performance and results that depend on the structural representation of the data.

6 Conclusions and outlook

We have introduced LAMINAR, a novel unsupervised machine learning pipeline that generates a locally-adaptive-metric to enhance the structural representation of data. LAMINAR is able to compute more-informative distances that preserve the underlying global structure of the data more effectively than the traditional Euclidean metric. Our approach avoids the need for pre-defining a meaningful metric, which is often a limitation in other density-based methods, and provides a robust means of uncovering the geometry inherent in the data itself. Our current implementation, found at https://github.com/CKleiber/LAMINAR, is a proof-of-concept and in a future work we aim to incorporate the techniques developed by Sorrenson et al. [12], employ an optimal-transport-based flow architecture, as well as to investigate whether additional geometric information can be utilized in order to enhance structural representation implicit within a data set.

Broader impact statement

The authors are not aware of any immediate ethical or societal implications of this work. This work purely aims to aid scientific research and proposes to apply normalizing flows to learn a meaningful distance metric that respects the structure implicit within any given point-based data set.

Acknowledgments and Disclosure of Funding

This work is funded by the Carl-Zeiss-Stiftung through the NEXUS program.

References

[1] Avleen S Bijral, Nathan Ratliff, and Nathan Srebro. Semi-supervised learning with density based distances. *arXiv preprint arXiv:1202.3702*, 2012.

[2] Olivier Bousquet, Olivier Chapelle, and Matthias Hein. Measure based regularization. *Advances in Neural Information Processing Systems*, 16, 2003.

[3] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019. URL `https://arxiv.org/abs/1806.07366`.

[4] Timothy Chu, Gary L Miller, and Donald R Sheehy. Exact computation of a manifold metric, via lipschitz embeddings and shortest paths on a graph. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 411–425. SIAM, 2020.

[5] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[6] Pablo Groisman, Matthieu Jonckheere, and Facundo Sapienza. Nonhomogeneous euclidean first-passage percolation and distance learning. *Bernoulli*, 28(1):255–276, 2022.

[7] Sung Jin Hwang, Steven B Damelin, and Alfred O Hero III. Shortest path through random points. 2016.

[8] Anna Little, Daniel McKenzie, and James M. Murphy. Balancing geometry and density: Path distances on high-dimensional data. *SIAM Journal on Mathematics of Data Science*, 4(1):72–99, 2022. doi: 10.1137/20M1386657. URL `https://doi.org/10.1137/20M1386657`.

[9] Sajama and Alon Orlitsky. Estimating and computing density based distance metrics. In *Proceedings of the 22nd international conference on Machine learning*, pages 760–767, 2005.

[10] Sanjib Sharma and Kathryn V. Johnston. A group finding algorithm for multidimensional data sets. *The Astrophysical Journal*, 703(1):1061–1077, sep 2009. doi: 10.1088/0004-637x/703/1/1061. URL `https://doi.org/10.1088/0004-637x/703/1/1061`.

[11] Sanjib Sharma and Matthias Steinmetz. Multidimensional density estimation and phase-space structure of dark matter haloes. *Monthly Notices of the Royal Astronomical Society*, 373(4): 1293–1307, December 2006. doi: 10.1111/j.1365-2966.2006.11043.x.

[12] Peter Sorrenson, Daniel Behrend-Uriarte, Christoph Schnörr, and Ullrich Köthe. Learning distances from data with normalizing flows and score matching. *arXiv preprint arXiv:2407.09297*, 2024.

[13] Nicolás García Trillos, Anna Little, Daniel McKenzie, and James M Murphy. Fermat distances: Metric approximation, spectral convergence, and clustering algorithms. *arXiv preprint arXiv:2307.05750*, 2023.