# Response to Reviewer 1's

# *"Report on 'Distortion corrected kernel density estimation on Riemannian manifolds" by F.Cheng, R.J Hyndman & A.Panagiotelis'*

We would like to extend our appreciation to the reviewer for their detailed and insightful feedback on our manuscript titled "Distortion corrected kernel density estimator on Riemannian manifolds." Your comments have been invaluable in enhancing the clarity and depth of our work. We have carefully addressed each point raised and provide detailed responses below highlighted in blue text.

## Review

*However, I believe that the quality of the paper could be improved by comparing in more detail the performances with other density estimation methods that do not rely on a first step being the estimation of an embedding. Indeed, while embedding and then correcting for the distortion is a good idea, if one is only interested in estimating the density then the additional embedding step may potentially introduce unnecessary bias/instability. I have detailed my points below.*

1. *In page 2 lines* $20-27$ *it is said that several methods for density estimation on manifolds are limited to the case of known manifolds. While this is true for most of the examples, this is not true for [2. In this example the manifold needs not to be known for the method to be tractable. However it is assumed that the observations are living in a small tubular neighbourhood around the manifold, not exactly on it. Furthermore, at this point in the paper it would be sensible to cite the works of Berenfeld & Hoffman [1] (which is actually mentioned later on) and Divol [3]. These works show that several types of KDEs are actually minimax optimal in the context of density estimation on manifolds. The goal of the present being slightly different (estimating the density of the sample after embedding), the comparison with the aforementioned works would*

*be relevant and would help the reader to understand the contribution of the present paper.*

We have revised the manuscript to clarify the distinction between our approach and those that do not require a known manifold. Additionally, we have included citations to the works of Berenfeld & Hoffmann, 2021 and Divol, 2022 to provide context on minimax optimality in density estimation on manifolds. In the introduction we now state:
"Recently, Berenfeld & Hoffmann, 2021, Berenfeld, Rosa & Rousseau, 2022 and Divol, 2022 demonstrated the minimax optimality of certain types of KDE on manifolds. In contrast, our objective in this paper is to propose a new KDE for data that lie on some unknown manifold found using embedding techniques."

2. *Following the previous remark, when the data is embedded in a Euclidean space (such as in the simulated examples: twin peaks and the semihypersphere), it is possible (even if the manifold is unknown) to perform density estimation directly on the data (and not after estimating an embedding) with, for instance, KDEs [1, 3] or special Dirichlet process mixtures [2] while preserving optimal statistical convergence properties (at least theoretically). Would your (empirical) results stay competitive in this case?*

While it is possible to perform KDE on the data without using an embedding, this is challenging for the high dimensional examples we consider. The popular `ks` package [Duong, 2007] in R recommends that it not be used for dimensions greater than 6 due to issues with numerical instability. The `weird` package [Hyndman, 2024] in R also implements multivariate kernel density estimation, however its function to find the bandwidth matrix was prohibitively slow even for a dimension of 5. The example we consider is 100-dimensional. In this case, we have coded a KDE by hand with a bandwidth proportional to an identity matrix, i.e. $\mathbf{H} = h^2\mathbf{I}$. The scale factor $h$ is found using Silverman's rule adapted for multivariate data : $h = \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} n^{-\frac{1}{d+4}}$. In our semi-hypersphere example, $d = 100$ and $n = 10,000$. Note that due to the curse of dimensionality the KDE on 100-dimensional data is somewhat flat with the densities of all points close to zero. This can be shown by observing the distribution of the estimated densities evaluated at each observation. This bottom panel (yellow violin plot) of Figure 1 shows the results for a 100-dimensional KDE without dimension reduction. In contrast to the true densities and the KDEs on emeddings, the 100-dimensional KDE estimate does not detect the high proportion of points with relatively lower densities.

In high-dimensional spaces, the data sparsity leads to exponentially slower convergence rates and requires an impractically large sample size to achieve reliable density estimates. Computationally, bandwidth selection algorithms are extremely demanding, both in terms of processing time and memory requirements. All these factors combined make traditional KDE methods unsuitable for 100-dimensional data, and dimension reduction is essential as proposed here.

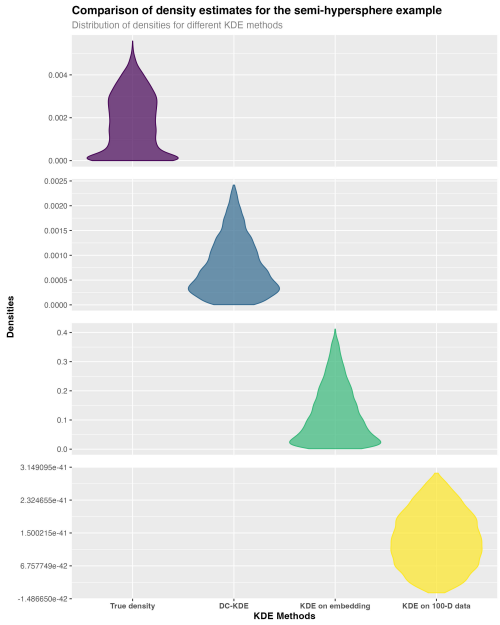


Figure 1: Comparison of KDE methods for high-dimensional data.

3. *During all of the experiments (both with simulated and real data), the choices of the parameters $\varepsilon, r$ are not really discussed thoroughly, even though it is known that tuning them is usually quite hard. The problem for $r$ is even not related to the manifold learning problem as it also appears in classical KDE on flat Euclidean spaces. I believe it would be beneficial to give some guidance on this matter or at least some observations.*

We acknowledge the importance of parameter tuning and have added more details on this. Regarding the choice of $\varepsilon$ we use as stated at the end of Section 2.2 the default choice of 0.4. Experimenting with alternative values did not lead to appreciably better results that the default choice, and we now state this clearly in the paper. Regarding the scale $r$ we similarly experimented with different values. Since there is only a single parameter this could be tuned with reference to the downstream task. For example, for anomaly detection, $r$ could be chosen to maximise the coherence between selecting anomalies both in and out of sample using cross validation. We have revised the manuscript to state this in final paragraph of the conclusion.

## Miscellaneous

1. *page 2 line 32: it should be made clear at this point that the dimension $d$ is either considered known or needs to be estimated separately (i.e., the proposed method is not directly dimension agnostic).*

We have clarified that the dimension $d$ is assumed to be known as a priori or estimated separately, as our method is not dimension agnostic and $d$ is a crucial parameter. In the second paragraph of the paper we state (with the part in bold now added):

"These algorithms, which include ISOMAP, LLE, t-SNE, and UMAP among others, can be thought of as a mapping $\psi : M \to \mathbb{R}^d$ that take points on the manifold $\boldsymbol{p}$ to $d$-dimensional vectors $\boldsymbol{y}$, **where the dimension $d$ is either assumed to be known a priori or needs to be estimated separately**."

2. *Equation (1): a reference to a relevant textbook/source for the formula would be useful for the inexperienced reader.*

A book reference, Chapter 3 of "Riemannian Geometry" by Do Carmo, 1992, has been added to guide readers to a relevant source for Equation (1).

"By a change of variables, this would imply the following density for $\boldsymbol{y}$,

$$Pr(\boldsymbol{y} \in \psi(\mathcal{A})) = \int_{\psi(\mathcal{A})} (f \circ \psi^{-1})(\boldsymbol{y}) |\det \boldsymbol{H}(\boldsymbol{y})|^{1/2} d\boldsymbol{y}, \tag{1}$$

where $\psi(\mathcal{A})$ is the image of $\mathcal{A}$ under $\psi$ and $\boldsymbol{H}(\boldsymbol{y})$ is the Riemannian metric expressed in local coordinates given by the mapping $\psi$ (Do Carmo, 1992)."

3. *page 3 line 24: I think the author meant to write "Riemannian metric."*

We have corrected the missing "metric" in the specified line.

"Critical to our estimator, is obtaining an estimate of $|\boldsymbol{H}(\boldsymbol{y})|^{1/2}$ for this purpose we use the Learn Metric algorithm of Perrault-Joncas & Meila, 2013 which augments any dimension reduction algorithm with an estimate of the Riemannian **metric** at each data point."

4. *page 3 line 38: it is not clear how a 2 or 3-dimensional is constructed and is actually confusing in the context: is $d = 2$ or 3? probably not but clarifications would be useful here.*

Clarifications have been added to explain the construction of 2 or 3-dimensional spaces in order to visualize the output of dimension reduction, where $d = 2$ for a 2-dimensional scatterplot and $d = 3$ for a 3-dimensional scatterplot. We now state:

"In terms of data visualization, our approach allows a two or three-dimensional scatterplot of $\boldsymbol{y}$ to be augmented by coloring points according to the magnitude of the density estimate, where the dimension of $\boldsymbol{y}$, $d$, is set as 2 or 3 in the dimension reduction algorithms."

5. *page 7 lines* $36-42$*: this is a bit confusing, at first the graph is described as being a k-NN graph but later a radius parameter is mentioned (which is usually a parameter used to define random geometric graphs, not k-NN ones). It would be beneficial to clarify this point/the construction of the graph.*

Thank you for pointing this out. Either a K-nearest neighbor (KNN) or fixed radius method could be used. In the Learn Metric algorithm, the weighted neighborhood graph is constructed by using a fixed-radius nearest neighbor search instead of a KNN approach. The fixed-radius method considers all points within a specified radius of $\sqrt{\epsilon}$. This radius parameter allows the graph to adapt to varying densities in the data, capturing local structure more effectively. By using a fixed-radius approach, the algorithm ensures that the graph reflects the underlying geometry of the data manifold, which is crucial for the analysis. We now clarify this:

"First, a weighted neighborhood graph is constructed, with edges between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ when $\boldsymbol{p}_i$ is a fixed-radius nearest neighbor of $\boldsymbol{p}_j$ or vice versa, and edge weights depending on the distance between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ on the manifold. The fixed-radius method considers all points within a specified radius of $\sqrt{\varepsilon}$, allowing the graph to adapt to varying densities in the data and to capture local structure more effectively. Second, the discrete Laplacian on this graph $\hat{\mathcal{L}}_{\varepsilon,n}$ is estimated (Zhou & Belkin, 2011), where $\sqrt{\varepsilon}$ is the radius parameter in the previous step and a constant value $c = 0.25$ is used for the use of heat kernel in the weighted neighborhood graph."

6. *page 9 /algorithm 1: at step 2 it looks like you are using the symmetric normalized graph Laplacian. Shouldn't it be* $\tilde{W} = D^{-1/2}WD^{-1/2}$ *and not* $D^{-1}WD^{-1}$*? Also, the constant* $c$ *in the definition of* $\tilde{\mathcal{L}}_{n,\varepsilon}$ *is not defined, does the value matter?*

Here we simply restate the algorithm as defined in Algorithm 1 of Perrault-Joncas & Meila, 2013 who define $\tilde{W} = D^{-\lambda}WD^{-\lambda}$, and then $\lambda = 1$ in Step 2 of their Algorithm 3. We agree that by setting $\lambda = 1/2$, we would be using the symmetric normalized graph Laplacian but for consistency with the original reference, set $\tilde{W} = D^{-1}WD^{-1)}$ and use this in our computation.

Setting $c = 0.25$ is recommended when using heat kernel used in the weighted neighborhood graph $w_{i,j} = \exp(-\frac{1}{\varepsilon}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$. The values for $c$ is stated in Section 2.2, and we have provided an explanation for this choice now and included the value of $c$ in Step 2 of Algorithm 1. We also note that since $c$ and $\varepsilon$ only enter Algorithm 1 as a product, it is sufficient to tune only one of these parameters. This is now also discussed in the paper at the end of Section 2.2.

7. *page 9 /algorithm 1: it looks like the steps* 3&4 *are completely independent. In general, in order to get an estimated embedding one can use techniques based on neighborhood/nearest neighbors graphs (like*

*Laplacian Eigenmaps). If one does that and then uses the same graph to perform step 4 would your technique be "improved"? Have you tried that? A few words on this would be helpful.*

Thanks for your suggestion. Yes, we did use the same nearest neighbor graph from Step 1 to get the estimated embedding. The computational efficiency is largely improved by doing this as the nearest neighbor searching is the most time-consuming step. For implementation, we used approximate nearest neighbor searching methods proposed in another paper Cheng, Panagiotelis & Hyndman, 2021. We have now added further explanation in Section 2.2.
"This algorithm is implemented in a Python library *megaman* (McQueen et al., 2016) although our own results are based on a re-implementation of the algorithm in `R`. For computational efficiency, approximate nearest neighbor searching methods are implemented to construct the neighborhood graph (Cheng, Panagiotelis & Hyndman, 2021, Perrault-Joncas & Meila, 2013), which are then used in both steps 1 and 3."

8. *page 10 line 30: it is stated that for the estimation of the density at an unobserved point one can use "any smoothed average of nearest neighbors." Isn't there the same problem as in the original problem of density estimation at observed points? i.e., isn't there distortion issues as well when doing that?*

The referee is correct to point this out. Indeed to prevent this distortion the average of nearest neighbors should be weighted by a term that depends on the the determinant of the Riemannian at each of the nearest neighbors. We now state explicitly in the paper:
"To account for distortion, this average should be weighted by the determinant of the Riemannian at each of the nearest neighbors."

9. *page 13 line 53: can you point to a reference for the behavior of t-SNE in this context?*

(Cai & Ma, 2022) has been added here for discussion about the clustering behavior of t-SNE. t-SNE is designed to preserve local structure of the high-dimensional data very well, which means it emphasizes similarities between nearby points, which can lead to the formation of apparent clusters in the low-dimensional visualization. The paper mentions an "early exaggeration" stage in t-SNE that allows for easier movement of clusters in the early stages of the algorithm. This can contribute to the formation of spurious clusters.

10. *page 15 line 48: a definition/short reminder on ranks and their definition would be beneficial (possibly in appendix?).*

Thanks for your suggestion. We have included a brief definition of density rank in the Appendix B for

clarity.

"The density rank used in the paper is defined as the relative position or order of an item within a set from the lowest to the highest, based on its density value."

## Conclusion

*We are enthusiastic about the method and believe this is a valuable contribution; the only major missing point being, in my opinion, a more complete comparison with other methods. I hope my remarks will help the authors to improve the quality of the paper.*

We are grateful for the reviewer's enthusiasm and constructive remarks. We have made the suggested improvements and believe they have significantly enhanced the quality of the paper. Thank you once again for your valuable feedback.

## References


Berenfeld, Clément & Marc Hoffmann (2021). "Density estimation on an unknown submanifold". In: *Electronic Journal of Statistics* 15.1, pp. 2179–2223.

Berenfeld, Clément, Paul Rosa & Judith Rousseau (2022). "Estimating a density near an unknown manifold: a Bayesian nonparametric approach". In: *arXiv preprint arXiv:2205.15717*.

Cai, T Tony & Rong Ma (2022). "Theoretical foundations of t-sne for visualizing high-dimensional clustered data". In: *Journal of Machine Learning Research* 23.301, pp. 1–54.

Cheng, Fan, Anastasios Panagiotelis & Rob J Hyndman (Feb. 2021). "Computationally Efficient Learning of Statistical Manifolds". In: arXiv: `2103.11773 [cs.LG]`. URL: `https://ui.adsabs.harvard.edu/abs/2021arXiv210311773C`.

Divol, Vincent (2022). "Measure estimation on manifolds: an optimal transport approach". In: *Probability Theory and Related Fields* 183.1, pp. 581–647.

Do Carmo, Manfredo P. (1992). "Riemannian manifolds". In: *Riemannian geometry*. 2nd ed. Boston: Birkhäuser. Chap. 3, pp. 35–45.

Duong, Tarn (Oct. 2007). "ks: Kernel Density Estimation and Kernel Discriminant Analysis for Multivariate Data in R". In: *Journal of Statistical Software* 21, pp. 1–16.

Hyndman, Rob J (2024). *weird: Functions and Data Sets for "That's Weird: Anomaly Detection Using R" by Rob J Hyndman*. R package version 1.0.2.9000. URL: https://pkg.robjhyndman.com/weird-package/.

McQueen, James et al. (2016). "Megaman: Scalable Manifold Learning in Python". In: *J. Machine Learning Research* 17.148, pp. 1–5.

Perrault-Joncas, Dominique & Marina Meila (May 2013). "Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery". In: arXiv: 1305.7255 [stat.ML].

Zhou, Xueyuan & Mikhail Belkin (2011). "Semi-supervised learning by higher order regularization". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. Proceedings of Machine Learning Research. JMLR Workshop & Conference Proceedings, pp. 892–900.