Distortion-corrected kernel density estimate on Riemannian manifolds

Fan Cheng
Monash University
Email: Fan.Cheng@monash.edu

Anastasios Panagiotelis
The University of Sydney
Email: Anastasios.Panagiotelis@sydney.edu.au

Rob J Hyndman
Monash University
Email: Rob.Hyndman@monash.edu

25 October 2022

Distortion-corrected kernel density estimate on Riemannian manifolds

Abstract

Manifold learning can be used to obtain a low-dimensional representation of the underlying Riemannian manifold given the high-dimensional data. However, kernel density estimates of the low-dimensional embedding with a fixed bandwidth fail to account for the way manifold learning algorithms distort the geometry of the Riemannian manifold. We propose a novel distortion-corrected kernel density estimator (DC-KDE) for any manifold learning embedding by introducing the estimated Riemannian metric of each point to fix the distortion in the line and volume elements. The geometric information of the manifold guarantees a more accurate density estimation of the true manifold, which subsequently could be used for anomaly detection. To compare our proposed estimator with a fixed-bandwidth kernel density estimator, we run two simulations with a 2-D data from Gaussian mixture model mapped into a 3-D twin peaks shape and a 5-D semi-hypersphere mapped in a 100-D space. We demonstrate that the proposed DC-KDE could improve the density estimates given a good manifold learning embedding and has higher rank correlations with the true manifold density. A shiny app in R is also developed for various simulation scenarios. The proposed method is applied to density estimation in statistical manifolds of electricity usage with the Irish smart meter data. This demonstrates the estimator's capability to fix the distortion of the manifold geometry and a new approach to anomaly detection in high-dimensional data.

Keywords: manifold learning, variable bandwidth, Riemannian metric, geodesic distance, Gaussian kernels

1 Introduction

Multivariate kernel density estimation has gained lots of attention in exploratory data analysis. It is a non-parametric technique to estimate the data density based on weighted kernels centered at the data which usually belongs to a subset of $\mathbb{R}^d$. Applications of kernel density estimation [KDE; Parzen (1962); Chen (2017)] include finding hot spots of traffic network in the GIS environment (Xie & Yan 2008; Okabe, Satoh & Sugihara 2009), automatic detection in visual surveillance systems (Elgammal et al. 2002), wind power density detection (Jeon & Taylor 2012), prime prediction via Twitter messages (Gerber 2014), and so on. However, when samples are assumed to be drawn from a Riemannian manifold embedded in a high-dimensional space of much more than $\mathbb{R}^d$, kernel density estimation has to be generalized to a non-Euclidean space and further approximation methods have to be adapted. Pelletier (2005) propose a kernel density estimator based on the Riemannian geodesic distance of the manifold but it is only applicable when the underlying manifold is known. Manifold learning algorithms could be applied to reduce the dimension and get an approximation of the manifold, but different manifold learning algorithms could induce different distortions of the same manifold. Therefore, we propose a distortion-corrected kernel density estimator for Riemannian manifolds embedded in more than $\mathbb{R}^d$. Our estimator could be applied to any reasonable manifold learning embedding from the high-dimensional sample data and fix the distortions at each point with estimated Riemannian geodesic distance and volume density function. This estimator could be further applied for unsupervised tasks such as anomaly detection where the outliers are the lowest density points.

For a given kernel function, kernel density estimation is flexible to learn the shape of the underlying density of the data controlled by the bandwidth and the selection of bandwidth is crucial in KDE (Jones 1990; Terrell & Scott 1992). Many bandwidth selection methods have been proposed in the literature, including the rule-of-thumb, cross-validation (Jones & R. F. Kappenman 1992; Sain, Baggerly & Scott 1994) and plug-in methods (See Heidenreich, Schindler & Sperlich 2013; Scott 2015, for details). For univariate kernel density estimation, the bandwidth selection problem has been thoroughly investigated (See Jones, Marron & Sheather 1992; Cao, Cuevas & González Manteiga 1994; Jones, Marron & Sheather 1996; Wand & Jones 1994, for reviews). The generalization to multivariate case could mostly be found in Duong & Hazelton (2003), Duong (2004), and Chacón & Duong (2010). In this paper, we focus on the multivariate kernel density estimation.

Note that a fixed bandwidth matrix $\boldsymbol{H}$ is a global smoothing parameter for all data points. However, when the local data structure is not universal for all sample data, which is true in most applications, an adaptive bandwidth matrix that is varying rather than fixed at each data point is needed. The bandwidth is varied depending on either the location of the sample points [sample smoothing

estimator; Terrell & Scott (1992)] or that of the estimated points [balloon estimator; Terrell & Scott (1992)]. In this paper, the densities are estimated at the sample points themselves, so we only need to consider the case where the bandwidth changes for each sample point and will refer to this as the *variable/adaptive kernel density estimation* [VKDE; Section 6.6 of Scott (2015)] unless otherwise stated. However, these kernel density estimators are based on random samples in the Euclidean space.

For samples points lying on a manifold with the differentiable structure called the Riemannian manifold, Pelletier (2005) generalize the kernel density estimator based on the kernel weights from the geodesic distance between the estimated points and the sample points. The idea of the estimator is to use a strictly positive function of the geodesic distance on the manifold and then normalize it with the volume density function of the Riemannian manifold for curvature (Henry & Rodriguez 2009). However, in many application scenarios, we tend to find that the sample points are not drawn directly from the manifolds because they are embedded in a much higher-dimensional space. Therefore, the kernel density estimator from Pelletier (2005) is not applicable because the geodesic distance and the volume density function are unknown. This is when we introduce manifold learning to reduce the input data dimension. For these high-dimensional data set, various manifold learning algorithms including ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP (see details of these algorithms in Cheng, Hyndman & Panagiotelis (2021)), could be applied to get a low-dimensional embedding, which are used as approximations of the underlying manifold.

In manifold learning, the underlying idea is that the data lies on a low-dimensional smooth manifold that is embedded in a high-dimensional space. One of the fundamental objectives of manifold learning is to explore the geometry of the dataset, including the distances between points and volumes of regions of data. These intrinsic geometric attributes of the data, such as distances, angles, and areas, however, can be distorted in the low-dimensional embedding, leading to failure in recovering the geometry of the manifold (Goldberg et al. 2008). To tackle this problem and measure the distortion incurred in manifold learning, Perrault-Joncas & Meila (2013) propose the Learn Metric algorithm to augment any existing embedding output with geometric information in the Riemannian metric of the manifold itself. By applying this algorithm, the outputs of different manifold learning methods can be unified and compared under the same framework, which would highly benefit in improving the effectiveness of the embedding. The Riemannian metric using the method of Perrault-Joncas & Meila (2013) gives some idea of the distortion of an embedding. Mapping the points through a non-linear function "stretches" some regions of space and "shrinks" others. The Riemannian gives us an idea of the direction and angle of this stretching at each point, which is informative for learning the manifold.

By exploiting the connection between the estimated Riemannian metric and the Riemannian geodesic distance as well as the volume density function for curvature, we propose the main contribution of the paper, which is the variable distortion-corrected kernel density estimator (DC-KDE) for manifold learning embedding. Starting from the high-dimensional sample data, we apply manifold learning algorithms to get the low-dimensional embedding in the same dimensional space as the underlying manifold together with the estimated Riemannian matrix at each embedding point. Then the DC-KDE is used to estimate the density of the manifold and distortions induced by manifold learning methods are fixed with the estimated geometric information. Our distortion-corrected estimator is novel in filling the gap between the high-dimensional sample space and the density of the unknown manifold. These density estimates are useful in many other areas, including classification, clustering and anomaly detection. Similar to Cheng, Hyndman & Panagiotelis (2021), the highest density region plots(Hyndman 1996) could be generated using the kernel density estimates for outlier visualization, which brings a novel anomaly detection method for Riemannian manifolds.

The rest of the paper is organized as follows. In **??**, we present our distortion-corrected kernel density estimator for Riemannian manifolds. We start by introducing the multivariate kernel density estimate with adaptive bandwidth and the kernel density estimator for Riemannian manifolds. Then we provide justification for the use of Riemannian metric to correct the distortions in manifold learning embedding and further apply the proposed estimator for anomaly detection. Section 3 is composed of two simulations with the proposed anomaly detection algorithm; the first deals with 2-dimensional data from gaussian mixture model mapped into a 3-D twin peaks structure and the second with a 5-D semi-hypersphere data mapped in a 100-D space. Different manifold learning algorithms are applied to the high-dimensional data to get the low-dimensional embedding which are then used to estimate densities and detect anomalies. Section 4 contains the application to visualize and identify anomalous households in the Irish smart meter dataset. Conclusions and discussions are presented in Section 5. Readers interested in the notions of Riemannian geometry mentioned in this paper could use Appendix A as a reference.

2 Distortion Corrected Kernel density estimate on Riemannian manifolds - Tas

In this section we introduce our method for kernel density estimation on manifolds that uses an embedding from a dimension reduction algorithm while correcting for the distortion induced by this embedding. Some readers familiar with kernel density estimation may not be as familiar with the nuances of manifolds, therefore, before introducing our own estimator, we first discuss kernel

density estimation for data in Euclidean space, then illustrate in Section 2.1 how this generalises to the estimator of Pelletier (2005), when the data lie on some known manifold. In Section 2.2 we describe the Metric Learning algorithm of Perrault-Joncas & Meila (2013), which augments an embedding derived from a dimension reduction algorithm with an estimate of the Riemannian metric expressed in local coordinates. By combining elements from the work of Pelletier (2005) and Perrault-Joncas & Meila (2013) we derive our own novel distortion corrected kernel density estimate in Section 2.3. To keep this section as succinct as possible, we do not define concepts such as manifolds, charts, geodesic distance etc., but provide this information for readers unfamiliar with differential geometry in Appendix A.

In the following we denote as M the d-dimensional manifold from which our data are sampled. Points on this manifold are denoted $\boldsymbol{p}$ in general with $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ denoting the observed sample. Often $\boldsymbol{p}_i$ will be high-dimensional vectors such that $\boldsymbol{p}_i \in \mathbb{R}^m$ with $m >> d$. However this need not be that case, for instance the $\boldsymbol{p}_i$ may be probability distributions on a statistical manifold. The methods we propose for estimating the density at each $\boldsymbol{p}_i$ only require some sense of distance between the 'input' points $d(\boldsymbol{p}_i, \boldsymbol{p}_j)$, such that we can apply dimension reduction algorithms to obtain an 'output' embedding $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ where $\boldsymbol{y}_i \in \mathbb{R}^d$. We will denote this embedding as $\boldsymbol{y}_i = \psi(\boldsymbol{p}_i)$ Finally, we denote by λ the Lebesgue measure of $\mathbb{R}^d$, letting $||.||$ be the usual Euclidean norm and following Pelletier (2005) we make these assumptions about the kernel function $K : \mathbb{R}_+ \to \mathbb{R}$

$$(i) \int_{\mathbb{R}^d} K(\|\boldsymbol{y}\|)\mathrm{d}\lambda(\boldsymbol{y}) = 1; (ii) \int_{\mathbb{R}^d} \boldsymbol{y} K(\|\boldsymbol{y}\|)\mathrm{d}\lambda(\boldsymbol{y}) = 0; (iii) \int_{\mathbb{R}^d} \|\boldsymbol{y}\|^2 K(\|\boldsymbol{y}\|)\mathrm{d}\lambda(\boldsymbol{y}) < \infty;$$
$$(iv)\ \mathrm{supp}\, K = [0;1]; (v)\ \sup K(||\boldsymbol{y}||) = K(0).$$

Note that these conditions are different (and in some cases stricter) that those normally used for kernel density estimation. For instance condition (iv) requires the support of the kernel to be bounded. The reasons for this will become clearer when we discuss the manifold setting in more detail. Also, for illustration purposes, in this section we pay particular attention to the uniform kernel for which $K(z)$ equals one if $0 \leq z \leq 1$ and zero otherwise. In our empirical section more general kernel functions can be, and are, employed.

For data $\boldsymbol{y}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$ and assuming a bandwidth matrix $r\boldsymbol{I}$ where r is a global bandwidth, then the usual kernel density estimator at a point $\boldsymbol{y}$ is given by

$$\hat{f}(\boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r^d} K\left(\frac{||\boldsymbol{y} - \boldsymbol{y}_i||}{r}\right).$$

The intuition behind this estimator is very clear for a uniform kernel. The density at a point $\boldsymbol{y}$ is equal to the proportion of sample points that lie within a ball of radius r centered at $\boldsymbol{y}$, times a term that ensures the density integrates to 1. In general, the bandwidth matrix need not be proportional to the identity matrix. However, the intuition remains the same, only that the ball of radius r centered at $\boldsymbol{y}$ is found with respect to Mahalanobis distance rather than the usual Euclidean distance. For more on kernel density estimation in the Euclidean case see Scott (2015) and references therein.

While one could in principle use a standard kernel density on the output from a dimension reduction algorithm, it must be noted that the density of the output vectors $\boldsymbol{y}_i$ is different to the density on the manifold itself. As a non-linear transformation, any dimension reduction algorithm 'distorts' the density. Furthermore any analysis based on density estimates of the output embedding will be dependent on the choice of dimension reduction algorithm since each different algorithm will distort the density in a different way.

2.1 Kernel Density estimation on manifolds

For kernel density estimation on a known manifold, Pelletier (2005) propose the following estimator,

$$\hat{f}(\boldsymbol{p}) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{r^d \theta_{\boldsymbol{p}_i}(\boldsymbol{p})} K\left(\frac{d_g(\boldsymbol{p},\boldsymbol{p}_i)}{r}\right), \tag{1}$$

where $d_g(\boldsymbol{p},\boldsymbol{p}_i)$ denotes the geodesic distance between two points on the manifold $\boldsymbol{p}$ and $\boldsymbol{p}_i$ and $\theta_{\boldsymbol{p}_i}(\boldsymbol{p})$ is known as the volume density function. The intuition behind the term $K\left(\frac{d_g(\boldsymbol{p},\boldsymbol{p}_i)}{r}\right)$ is relatively clear. For example, for a uniform kernel the estimator at point $\boldsymbol{p}$ will still be depend on the proportion of sample points within a ball of radius r centered at $\boldsymbol{p}$. However in this case, the geodesic distance on the manifold is used, rather than Euclidean or Mahalanobis distance. An additional technical assumption is that r is less than the injectivity radius of the manifold. A definition of the injectivity radius is given by Chavel (2006) and also provided in the appendix. For our purposes, it is sufficient to note that this assumption precludes the possibility that the radius of a ball around $\boldsymbol{p}$ is so large that some points 'fall inside' the ball more than once. For example on a sphere, a ball with radius greater than half the circumference of a great circle will wrap back around the sphere. This phenomenon also explains why the kernel function must be bounded for density estimation on manifolds.

The inclusion of the volume density function $\theta_{\boldsymbol{p}_i}(\boldsymbol{p})$ is perhaps not as immediately clear, therefore before providing formal details we will briefly discuss the intuition behind including this term. We have already highlighted that when using a uniform kernel, the kernel density estimate at a point $\boldsymbol{p}$ directly depends on the proportion of sample points within a ball of radius r around $\boldsymbol{p}$. However the volume of this ball must also be taken into account. In Euclidean space with the usual Lebesgue

measure, a radius r ball will always have the same volume regardless of its center. The same does not hold for manifolds and the volume density function ensures that the density estimate integrates to one.

More formally, the volume density function can be explained as follows. Consider the exponential map around $\boldsymbol{p}$, given by $exp_{\boldsymbol{p}}(\boldsymbol{q})$, mapping vectors in the tangent space $\boldsymbol{v} \in T_{\boldsymbol{p}}M$ to points the manifold $\boldsymbol{q} \in M$. Loosely, $\boldsymbol{v}$ 'points' in the direction of the geodesic between $\boldsymbol{p}$ and $\boldsymbol{q}$ and travel along this geodesic at uniform speed $||\boldsymbol{v}||$ takes place in one unit of time. Now, consider a chart φ mapping points in the neighbourhood of $\boldsymbol{p}$, via the inverse of the exponential map, to these $\boldsymbol{v}$ vectors, expressed in some local coordinate system. The volume density function is the square root of the determinant of the Riemannian metric expressed in this coordinate system. For more on the volume density function see Brigant & Puechmorel (2019).

2.2 Riemannian metric estimation

To be able to apply the estimator of Pelletier (2005) to the case where the manifold is not known, but where coordinates $\boldsymbol{y}_i$ for $i = 1, \ldots, n$ are obtained from a dimension reduction algorithm, requires an estimate of the Riemannian metric in the coordinate system. Formally, the Riemannian metric g is a symmetric and positive definite tensor field which defines an inner product $\langle , \rangle_g$ on the tangent space $T_{\boldsymbol{p}}M$ for every point $\boldsymbol{p} \in M$. The inner product between two tangent vectors $u, v \in T_{\boldsymbol{p}}M$, given by $\langle u, v \rangle_g$, can be used to define geometric quantities. For example angles on a manifold are given by $\cos\theta = \frac{\langle u,v \rangle_g}{|u||v|}$, while distances and volumes on manifolds are also defined with reference to the Riemannian metric. While tangent vectors, the Riemannian metric and the geometric quantities they define are invariant to any specific choice of coordinates, they can still be expressed in terms of local coordinates systems. This is precisely the situation when data on a manifold are mapped to d-dimensional Euclidean vectors $\boldsymbol{y}_1, \boldsymbol{y}_2 \ldots, \boldsymbol{y}_n$ via a dimension reduction algorithm. After this mapping, angles, distances and volumes in this Euclidean 'output space' are not the same as on the manifold since dimension reduction algorithms introduce distortion. To alleviate this issue Perrault-Joncas & Meila (2013) propose a method to augment $\boldsymbol{y}_1, \boldsymbol{y}_2 \ldots, \boldsymbol{y}_n$ with $d \times d$ positive definite matrices $\boldsymbol{H}_1, \boldsymbol{H}_2 \ldots, \boldsymbol{H}_n$ at each data point. These matrices estimate the Riemannian metric in local coordinates defined by the dimension reduction algorithm, for example the angle between $\boldsymbol{p}_j$ and $\boldsymbol{p}_k$ at $\boldsymbol{p}_i$ depends (up to a first order approximation) on the inner product $(\boldsymbol{y}_j - \boldsymbol{y}_i)' \boldsymbol{H}_i^{-1} (\boldsymbol{y}_k - \boldsymbol{y}_i)$ rather than the usual Euclidean inner product $(\boldsymbol{y}_j - \boldsymbol{y}_i)'(\boldsymbol{y}_k - \boldsymbol{y}_i)$.

While full details are provided in Perrault-Joncas & Meila (2013) , we briefly describe the Learn Metric algorithm here. There are four main steps in the algorithm. First a weighted neighborhood graph is constructed, with edges between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ when $\boldsymbol{p}_i$ is a K-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. Second,

the discrete Laplacian on this graph $\hat{\mathcal{L}}_{\varepsilon,n}$ is estimated (Zhou & Belkin 2011), where ε is the radius parameter for the nearest neighbor graph. Third, a dimension reduction method is applied to obtain the output embedding $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$. Fourth, the Riemannian metric at each point is estimated by exploiting the connection between the Riemannian metric and the Laplace Beltrami operator (to which the graph Laplacian at step 2 is a discrete estimator). Full details on these four steps are provided in Algorithm 1. 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*.

As pointed out by Perrault-Joncas & Meila (2013), dimension reduction can be carried out such that the dimension of the output vectors is larger than the intrinsic manifold dimension d. In this cae the ranks of the matrices $\boldsymbol{H}_i$ are equal to d. Using a larger embedding dimension is justified since it is in general not possible to embed a manifold of dimension d globally into d-dimensional Euclidean space. In our simulated examples we abstract from this issue by constructing examples that can be globally embedded into d dimesional Euclidean space. In practice, to determine the dimension of the manifold, the *two-nearest neighbor estimator (TWO-NN estimator)* (Facco et al. 2017; Denti et al. 2021) can be used. The *R* library *intRinsic* (Denti 2021) implements this algorithm and is used in all examples involving real data where the intrinsic dimension is unknown.

2.3 Distortion corrected KDE

With all fundamentals introduced we can now give our novel Distortion Corrected KDE (DC-KDE) as

$$\hat{f}(\boldsymbol{p}_j) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{r^d}\left(\frac{|\det \boldsymbol{H}_j|}{|\det \boldsymbol{H}_i|}\right)^{1/2} K\left(\frac{||\boldsymbol{H}_i^{-1/2}(\boldsymbol{y}_j - \boldsymbol{y}_i)||}{r}\right).$$

The estimator has a similar structure to Equation (1) with some key differences. To understand these differences it is first critical to appreciate that the coordinates $\boldsymbol{H}_i^{-1/2}(\boldsymbol{y}_j - \boldsymbol{y}_i)$ give an embedding that is approximately isometric in a small neighborhood around the i^{th} observed point (this insight is discussed at length is section 6.2 of Perrault-Joncas & Meila (2013)). This is crucial for two reasons. First, this implies that the term $||\boldsymbol{H}_i^{-1/2}(\boldsymbol{y}_j - \boldsymbol{y}_i)||$ approximates the geodesic distance between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. Second, the estimator in Equation (1) is valid only when the coordinate mapping is the logarithmic map around $\boldsymbol{p}_i$, it is this mapping that is approximated by $\boldsymbol{H}_i^{-1/2}(\boldsymbol{y}_j - \boldsymbol{y}_i)$. For this reason there is a ratio of two determinants to ensure the density integrates to one, the first is a consequence of the mapping from the manifold to the coordinate system (from a dimension reduction algorithm) while the second is the transformation $\boldsymbol{H}_i^{-1/2}(\boldsymbol{y}_j - \boldsymbol{y}_i)$ which ensures that the embedding approximates the logarithmic map. Also worth noting is the resemblance between the

Algorithm 1: Learn metric algorithm in Perrault-Joncas & Meila 2013

Input : high-dimensional data $\boldsymbol{x}_i \in \mathbb{R}^s$ for all $i = 1, \ldots, N$
Output : low-dimensional data $\boldsymbol{y}_i \in \mathbb{R}^d$ and its Riemannian metric $\boldsymbol{H}(\boldsymbol{y}_i)$ for all $i = 1, \ldots, N$
parameter : embedding dimension d, bandwidth parameter $\sqrt{\varepsilon}$, manifold learning algorithm
optimization parameter: manifold learning parameters EMBED

1: Construct a weighted neighborhood graph $\mathcal{G}_{w,\varepsilon}$ with weight matrix $\boldsymbol{W}$ where $w_{i,j} = \exp(-\frac{1}{\varepsilon}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$ for data points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^s$;
2: Calculate the $N \times N$ geometric graph Laplacian $\widetilde{\mathcal{L}}_{\varepsilon,N}$ by

$$\widetilde{\mathcal{L}}_{\varepsilon,N} = 1/(c\varepsilon)(\widetilde{D}^{-1}\widetilde{W} - I_N),$$

where $\widetilde{D} = diag\widetilde{W}\mathbf{1}$, $\widetilde{W} = D^{-1}WD^{-1}$, and $D = diagW\mathbf{1}$;
3: Embed all data point $\boldsymbol{X} \in \mathbb{R}^s$ to embedding coordinates $\boldsymbol{Y} = (\boldsymbol{y}^1, \ldots, \boldsymbol{y}^d)'$ by any existing manifold learning algorithm EMBED;
4: Obtain the matrix $\tilde{\boldsymbol{H}}$ of all data point by applying the graph Laplacian $\widetilde{\mathcal{L}}_{\sqrt{\varepsilon},N}$ to the embedding coordinates matrix $\boldsymbol{Y}$ with each element vector in $\tilde{\boldsymbol{H}}$ being

$$\tilde{\boldsymbol{H}}^{ij} = \frac{1}{2}\left[\tilde{\mathcal{L}}_{\varepsilon,N}\left(\boldsymbol{y}^i \cdot \boldsymbol{y}^j\right) - \boldsymbol{y}_i \cdot \left(\tilde{\mathcal{L}}_{\varepsilon,n}\boldsymbol{y}^j\right) - \boldsymbol{y}^j \cdot \left(\tilde{\mathcal{L}}_{\varepsilon,n}\boldsymbol{y}^i\right)\right],$$

where $i, j = 1, \ldots, d$ and the $\cdot$ calculation is the elementwise product between two vectors;
5: Calculate the Riemannian metric $\boldsymbol{H}$ as the rank d pseudo inverse of $\tilde{\boldsymbol{H}}$ with

$$\boldsymbol{H} = \boldsymbol{U}diag1/(\Lambda[1:d])\boldsymbol{U}',$$

where $[\boldsymbol{U}, \Lambda]$ is the eigendecomposition of matrix $\tilde{\boldsymbol{h}}(x)$, and U is the matrix of column eigenvectors ordered by the eigenvalues Λ in descending order.

estimator and multivariate variable bandwidth estimation (Breiman, Meisel & Purcell 1977; Jones 1990; Terrell & Scott 1992).

One limitation of the kernel density estimator is that the density can be estimated only at points where data have been observed since the estimator requires the Riemannian $\boldsymbol{H}_j$. To estimate the density at points that do not belong to the sampled points, any smoothed average of nearest neighbors can be used instead. We note that the particular downstream task that we are interested in is anomaly detection for which only the density estimates at observed sample points are required, since anomalies are identified as the points with lowest density. The entire workflow is summarised in Figure 1. The last two steps in Figure 1 are the main contributions of Algorithm **??**, generating distortion-corrected KDE with adaptive Riemannian metric $\boldsymbol{H}_i$ at each point and computing the highest density region plots based on the density estimates for anomaly detection. Compared to the anomaly detection with a general kernel density estimator in Cheng, Hyndman & Panagiotelis (2021), the changes are also highlighted in blue. With this anomaly detection algorithm, outliers

based on lowest densities could be detected more accurately regardless of the distortion in manifold learning.

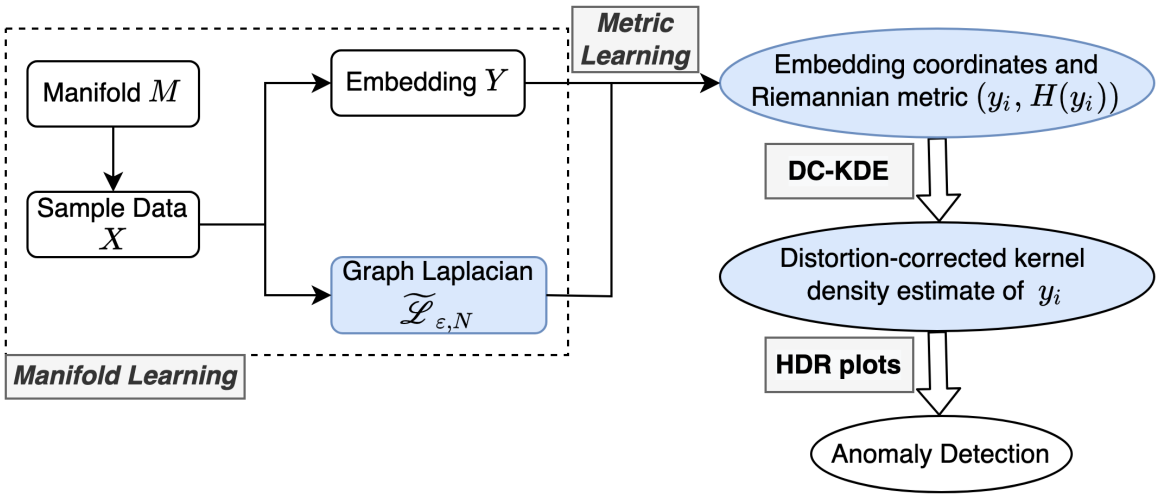

Figure 1: *The proposed schematic for anomaly detection with distortion-corrected kernel density estimates.*

3 Simulations

In this section, we examine two scenarios for both low and high dimensions to test our proposed algorithm. For visualization purposes, Section 3.1 presents an example of a two-dimensional manifold embedded in 3-dimensional Euclidean space. General stuff here...

We first simulate the data of size $N = 2,000$ from a mixture of four Gaussian kernels with the same covariance but different means, each consisting of 500 points. Different mapping functions are then applied to the 2-D meta data to be mapped in a 3-D feature space, which gives the higher-dimensional input for different manifold learning algorithms, including ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP. The embedded dimension is set as $d = 2$, the same as the meta data dimension. This enables us to compare the manifold learning embedding with the true meta data. We could now apply Algorithm **??** to get the density estimates of all data points and further detect anomalies. As a high-dimensional example, the second simulation in Section 3.2 is based on a 5-D meta data of size $N = 2,000$ embedded in a 100-D space and the corresponding embedding dimension is $d = 5$.

3.1 Twin peaks example

The simulation setup for the twin peaks example is to first generate vector $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ for $N = 2000$ from a 2-dimensional Gaussian mixture model. The mixture has four components with different means $\boldsymbol{\mu_1} = (0.25, 0.25)', \boldsymbol{\mu_2} = (0.25, 0.75)', \boldsymbol{\mu_3} = (0.75, 0.25)', \boldsymbol{\mu_4} = (0.75, 0.75)'$ and the same variance-covariance matrix $\boldsymbol{\Sigma}_i = diag(0.016, 0.016), i = 1, 2, 3, 4$. The mixture proportions are

equally set as $\pi_i = 0.25, i = 1,2,3,4$. The two dimensional data in Figure 2 is mapped to a 'twin peaks' surface via the mapping given by

$$\begin{aligned} x_1 &= v_1, \\ x_2 &= v_2, \\ x_3 &= \sin(\pi v_1)\tanh(3v_2). \end{aligned}$$

This is shown on the right panel of **??**. We also considered the 'Swiss Roll' mapping shown on the left panel of **??**, the results for this manifold are summarised in Appendix Appendix C.

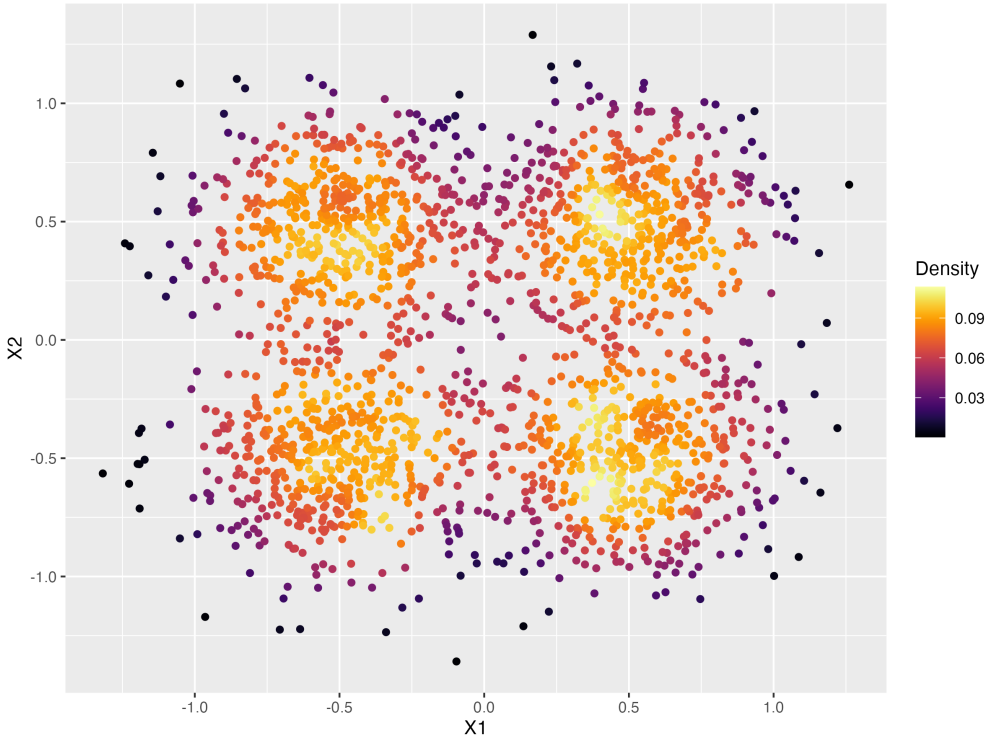

Figure 2: *True density of the Gaussian mixture model of four kenels with means* $(0.25, 0.25), (0.25, 0.75), (0.75, 0.25), (0.75, 0.75)$ *and the same variance-covariance matrix* $diag(0.016, 0.016)$. *The colors indicate the density of the data and lower density points in darker colors are scattered both in the outer and center areas. The shapes indicate the four kernels.*

It is important to note that the *true density* on the manifold is not simply a Gaussian mixture, since the mapping in Section 3.1 distorts the distribution. To recover the true distribution requires the correct change of variables from $\boldsymbol{v}$ to the twin peaks manifold. By treating the $\boldsymbol{z}$ as an 'output' embedding from input points $\boldsymbol{x}$ that lie on the true manifold and applying the learn metric algorithm, the true density on the manifold can be obtained as $p(\boldsymbol{p}_i) = f(\boldsymbol{v}_i)|\Gamma_i|^{1/2}$, where $f(\boldsymbol{v}_i)$ is a the density of a four component mixture of normals and $|\Gamma_i|^{1/2}$ are the outputs of using the learn metric algorithm on $\boldsymbol{v}$. Knowledge of $\boldsymbol{v}$ and $|\Gamma_i|^{1/2}$ will not be used when estimting the density, only to compare the estimated densities to the true densities. Figure 2 shows the simulated $\boldsymbol{v}$ with the colors indicating the true density of data on the manifold. Anomalies are defined as points with lowest densities

Table 1: *Correlation between true density ranking and estimated density ranking for different manifold learning embeddings of the twin peak data. Variable bandwidth KDE outperfoms for LLE and UMAP, and LLE gives the highest rank correlation.*

	ISOMAP	LLE	Laplacian.Eigenmaps	t.SNE	UMAP
DC-KDE	**0.823**	**0.673**	**0.672**	**0.806**	**0.794**
KDE	0.798	0.500	0.606	0.451	0.469

shown in black and 'typical' points are defined as points with the highest densities shown in yellow. Based on the true density plot, the anomalies are found around the edges of the plot, but also in the region between the means of the four mixture components. The objective is to determine whether we can correctly identify these anomalies without any knowledge of the true density or the $\boldsymbol{v}$.

Similar to Figure 19, different manifold learning embeddings are obtained and used to detect outliers with true densities and two bandwidth selection methods shown in Figure 3. In general, the four highest density regions in yellow are identified in almost all manifold learning embeddings except for ISOMAP with fixed bandwidth and t-SNE. For ISOMAP, our proposed variable KDE, compared with the true density, gives the most accurate mixture kernel structure with the lowest estimated densities (darker colored points) in the outside and the center of the embedding, and the kernel means (yellow points) with highest densities are also clearly identified. In contrast, the fixed bandwidth KDE failed to identify the lowest density area in the center. Both variable and fixed bandwidth KDE are quite close with LLE and Laplacian Eigenmaps, but in Laplacian Eigenmaps embedding, the top outliers are indexed in the middle instead of the true outer areas due to the large distortion in the middle. For t-SNE and UMAP, there are four clusters in the embedding and UMAP does a better job in finding the HDRs than t-SNE. Also due to the clusters in the embedding, the outliers found in UMAP are clustered.

We can gain further insight by comparing the correlation between ranks of true densities and estimated densities from variable and fixed bandwidth KDE by Table 1. Again the highest correlations appear from embedding with higher quality, including ISOMAP, Laplacian Eigenmaps, and UMAP. The rank correlations between variable and fixed bandwidth are equivalent to the third decimal place in Laplacian Eigemaps and UMAP. As for t-SNE, the four clusters in the embedding are less separated than in UMAP and our proposed method has misidentified the kernel cores, leading to a lower rank correlation in variable bandwidth. Since the embeddings from twin peak data generally capture the rectangular structure in the meta data than those from the swiss roll data, the rank correlations are much higher in Table 1 than in Table 4, with the lowest correlation being 0.451. This again suggests that the accuracy of outlier detection is highly related to the quality of manifold learning embedding.

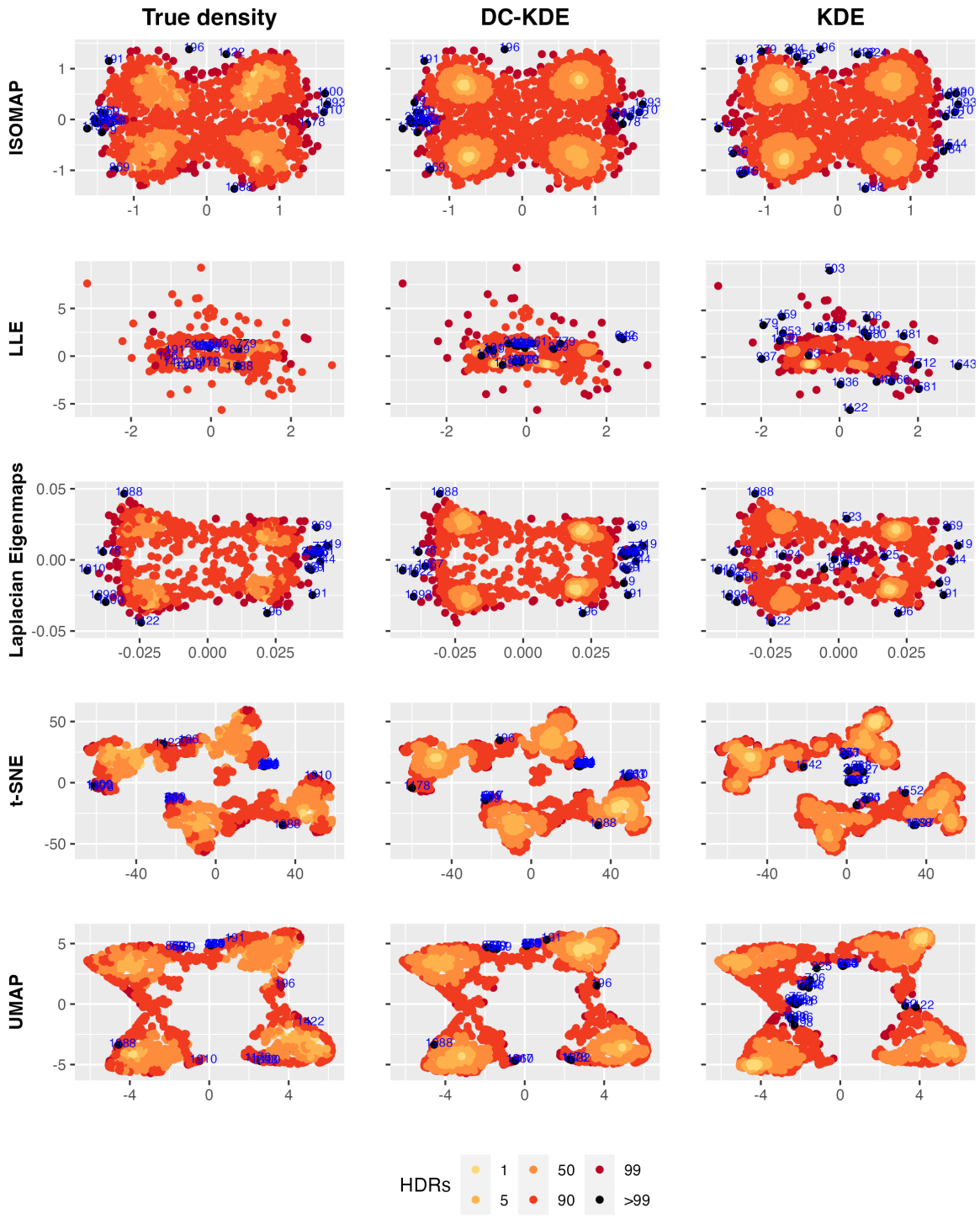

Figure 3: *Highest density region plots of five manifold learning embeddings of the twin peaks data in each row. The top 20 outliers, hightlighted in black and indexed in blue text, are found by the true manifold density (left panel), DC-KDE (middle panel) and KDE (right panel). DC-KDE finds more true outliers than KDE in all five rows.*

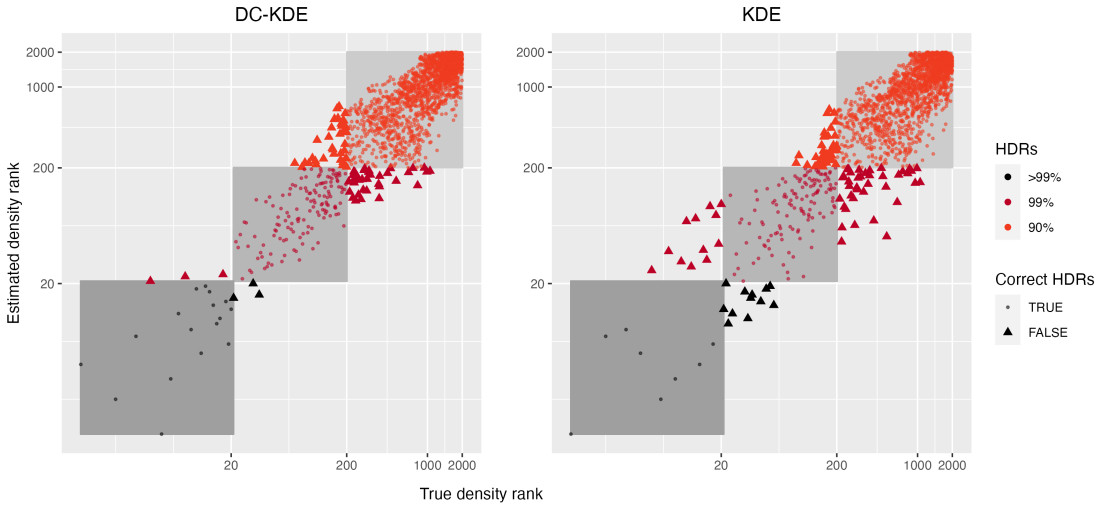

Figure 4: *Scatterplot of true density and estimated density ranks of ISOMAP embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.*

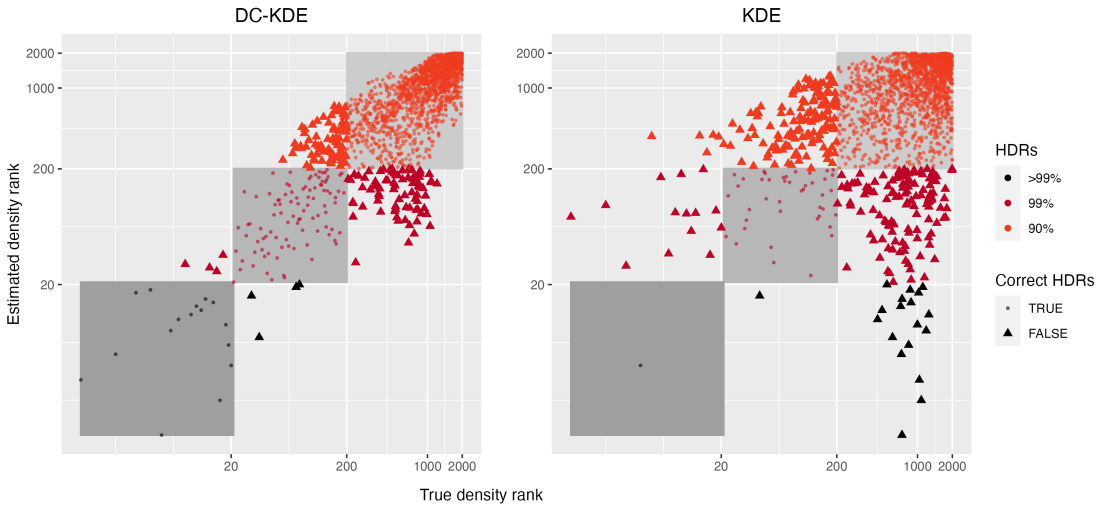

Figure 5: *Scatterplot of true density and estimated density ranks of t-SNE embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.*

In Figure 4, we plot the estimated density against the true density of the ISOMAP embedding for KDE with variable and fixed bandwidth, with colors and shapes showing the four kernels in the meta data. The linear positive relationship between the true densities and variable KDEs on the left handside is stronger than that of the fixed bandwidth KDEs. Combined with the top-right subplot in Figure 3, we could tell that most points are underestimated near the true kernel cores, which also suggests that the fixed bandwidth tries to smooth across all the data points and fails to fix the local distortions in the manifold learning process like the proposed pointwise variable bandwidth.

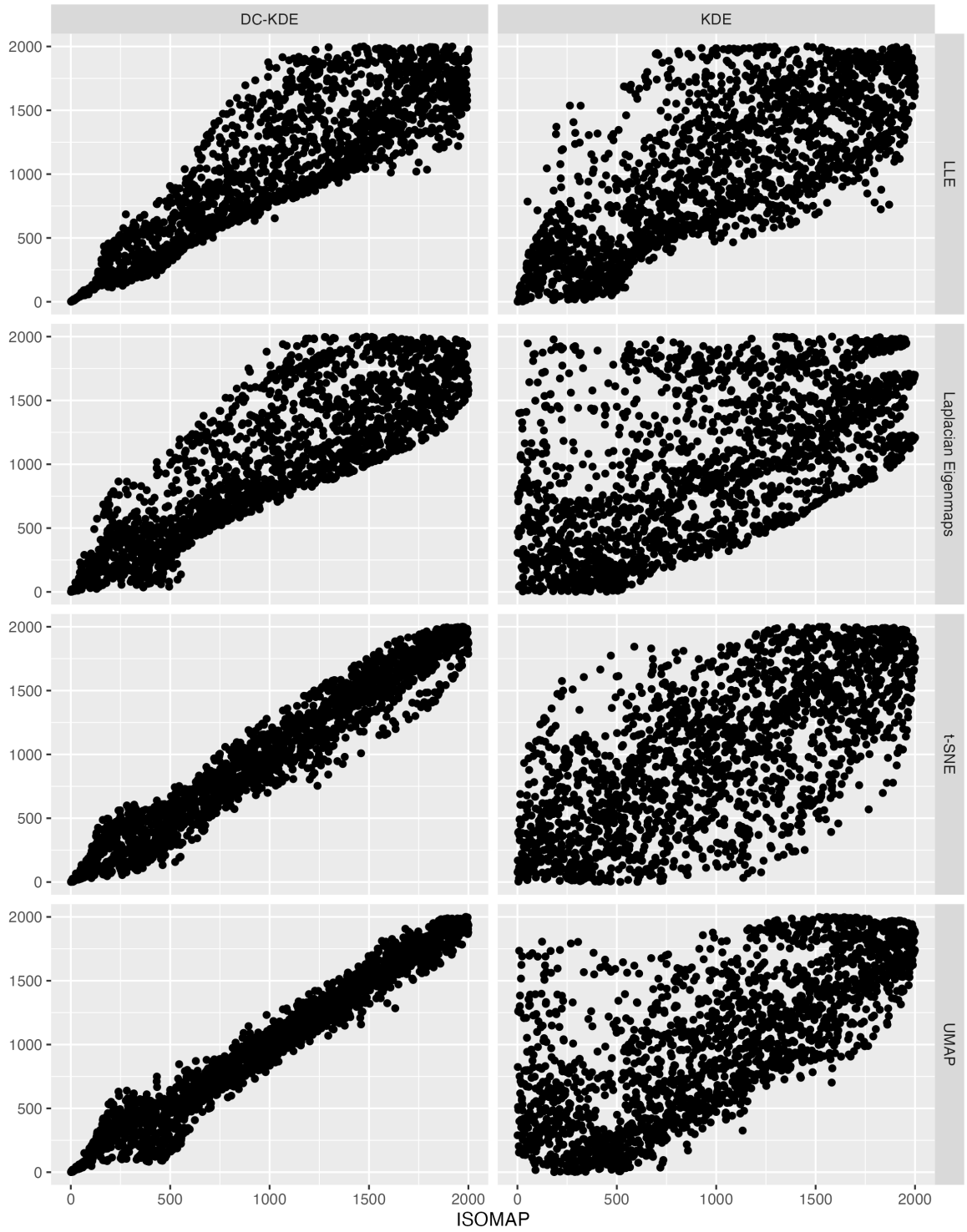

Figure 6: *Comparison of outliers found by ISOMAP and other four manifold learning methods for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless of the manifold learning embedding.*

Finally, **??**fig:tpisomapvs4ml) is used to show the robustness of the density estimates using DC-KDE regardless of the embedding method. By plotting the density ranks of LLE, Laplacian Eigenmaps, t-SNE, and UMAP against ISOMAP, we could tell that the rank correlation from DC-KDE is higher that that from KDE, meaning that the rank of DC-KDE are more consistent with the manifold learning methods. Compared with KDE where the density ranks varies with the embedding methods even for the same manifold, DC-KDE is more robust when the distortions are fixed.

3.2 100-D mapping from a 5-D semi-hypersphere

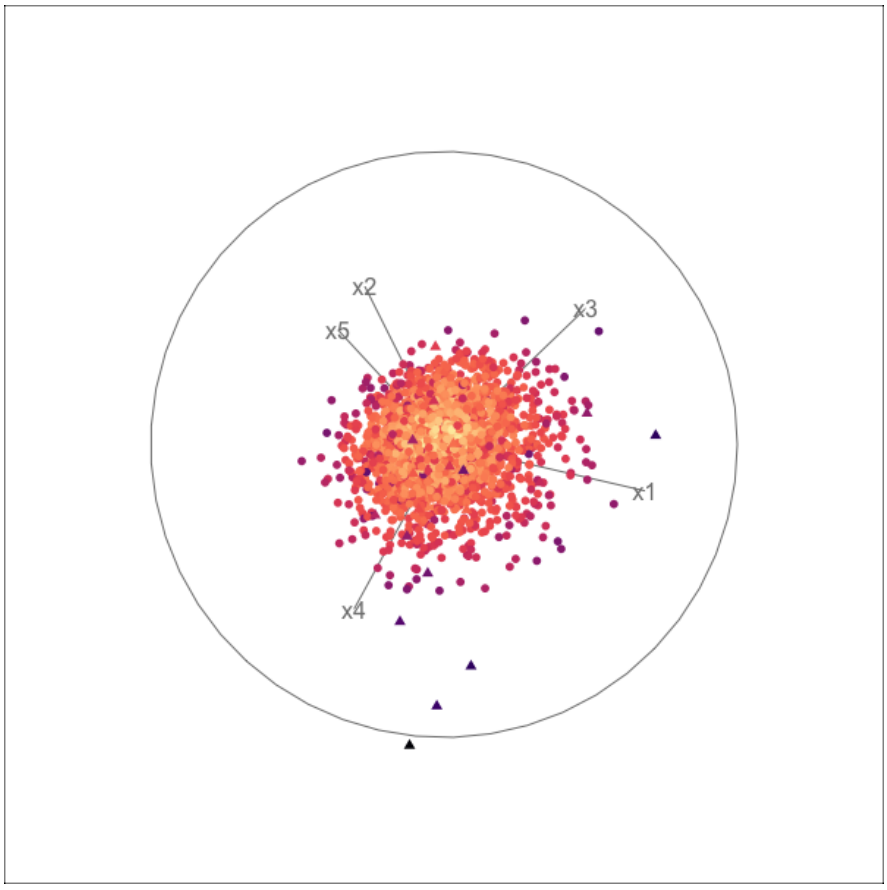

Figure 7: *Scatterplot display of the animation of a 5-D tour path with shapes indexing the Gaussian mixture component and the colors showing the distance to the kernel cores.*

As a high-dimensional experiment, we generate the meta data from a 5-dimensional semi-hypersphere, transform it into a 100-dimensional space, and then embed it in $d = 5$ with manifold learning. First, we simulate $N = 2,000$ points, $(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \boldsymbol{X}_4)'$, from a 4-dimensional Gaussian mixture model with two mixture components, $\mathsf{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathsf{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = (0,0,0,0)'$, $\boldsymbol{\Sigma}_1 = diag(1,1,1,1)$, and $\boldsymbol{\Sigma}_1 = diag(2,2,2,2)$. In order to manually add anomalies to be distant points from the means, the mixture proportions are set as $\pi_1 = 0.99$ and $\pi_2 = 0.01$. The fifth dimension is calculated to satisfy the five-dimensional semi-hypersphere surface equation $X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 = r^2$ where $X_5 > 0$ and r is set as 7. The Gaussian mixture densities could be calculated using Equation (**??**) as the true density of the 5-d meta data. Figure 7 shows a

scatterplot display when animating a 5-D tour path with the R package *tourr* [REFERENCE]. The round and triangular point shapes index the two mixture components $\mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2)$, and the colors shows the distance between the simulated $4-D$ data point from Gaussian mixture model and the kernel means $(0,0,0,0)'$. The more distant from the point to the kernel cores, the lower the true densities, which shows in a darker color in Figure 7. It can be seen that the most distant points are in a triangular shape, meaning that they are simulated from $\mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. The dark colors also indicate that they are the true outliers because of their low densities.

Then we initial the other 95 dimensions in the high-dimensional space as zero columns and further rotate the 100-dimensional data of size N(denote the transpose of the data matrix as $\boldsymbol{X}_0$ with dimension $100 \times N$) to get rid of the zeros so that it could be passed to the manifold learning algorithms. The rotation matrix is derived from the QR decomposition of a 100×100 matrix $\boldsymbol{A}$ with all components randomly generated from a uniform distribution $\mathcal{U}(0,1)$. For any real matrix $\boldsymbol{A}$ of dimension $p \times q$, the QR decomposition could decompose the matrix into the multiplication of two matrix $\boldsymbol{Q}$ and $\boldsymbol{R}$ so that $\boldsymbol{A} = \boldsymbol{QR}$, where the dimension of Q is a matrix with unit norm orthogonal vectors, $\boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I}$, and $\boldsymbol{R}$ is an upper triangular matrix. Matrix $\boldsymbol{Q}$ satisfies $\boldsymbol{X}_0'\boldsymbol{X} = (\boldsymbol{QX}_0)'(\boldsymbol{QX}_0)$, meaning that the pairwise Euclidean distances between data points in $\boldsymbol{X}_0'$ is equivalent to that of $(\boldsymbol{QX}_0)'$. Therefore, we use matrix Q as the rotation matrix for where the rotated data matrix $\boldsymbol{X} = (\boldsymbol{QX}_0)'$ of dimension $N \times 100$ is now the input data for the manifold learning algorithms. Again, we set the embedding dimension to be equal to the meta data dimension $d = 5$.

In Figure 8, the estimated densities are compared with the true density on the x-axis for four manifold learning embeddings, ISOMAP, LLE, Laplacian Eigenmaps, and UMAP. Note that we exclude t-SNE algorithm in this section because it is designed mainly for low-dimensional visualization purposes, and it is only applicable to embedding dimensions within three. Similar to Figure 7, the point shapes show the two mixture component in the meta data, and the colors represent the distance to the kernel means, with distant outliers shown in darker colors. For well-estimated densities, the true outliers with low true densities will also have low estimated densities, which suggests that darker-colored points should appear in the bottom-left corner in Figure 8. This is true for both ISOMAP and LLE, partly true for Laplacian Eigenmaps, but not in UMAP where these outliers have relatively high density estimates. For variable bandwidth KDE, there is a strong positive linear relationship with the true densities for ISOMAP, LLE, and Laplacian Eigenmaps, and the relationship is stronger than the fixed bandwidth. This suggests that our proposed KDE with variable bandwidth is more accurate than the fixed bandwidth in estimating the manifold learning embedding densities. In KDE with fixed bandwidth, the bandwidth is often too large to smooth across all data points, especially when there is severe distortion in the embedding data. By

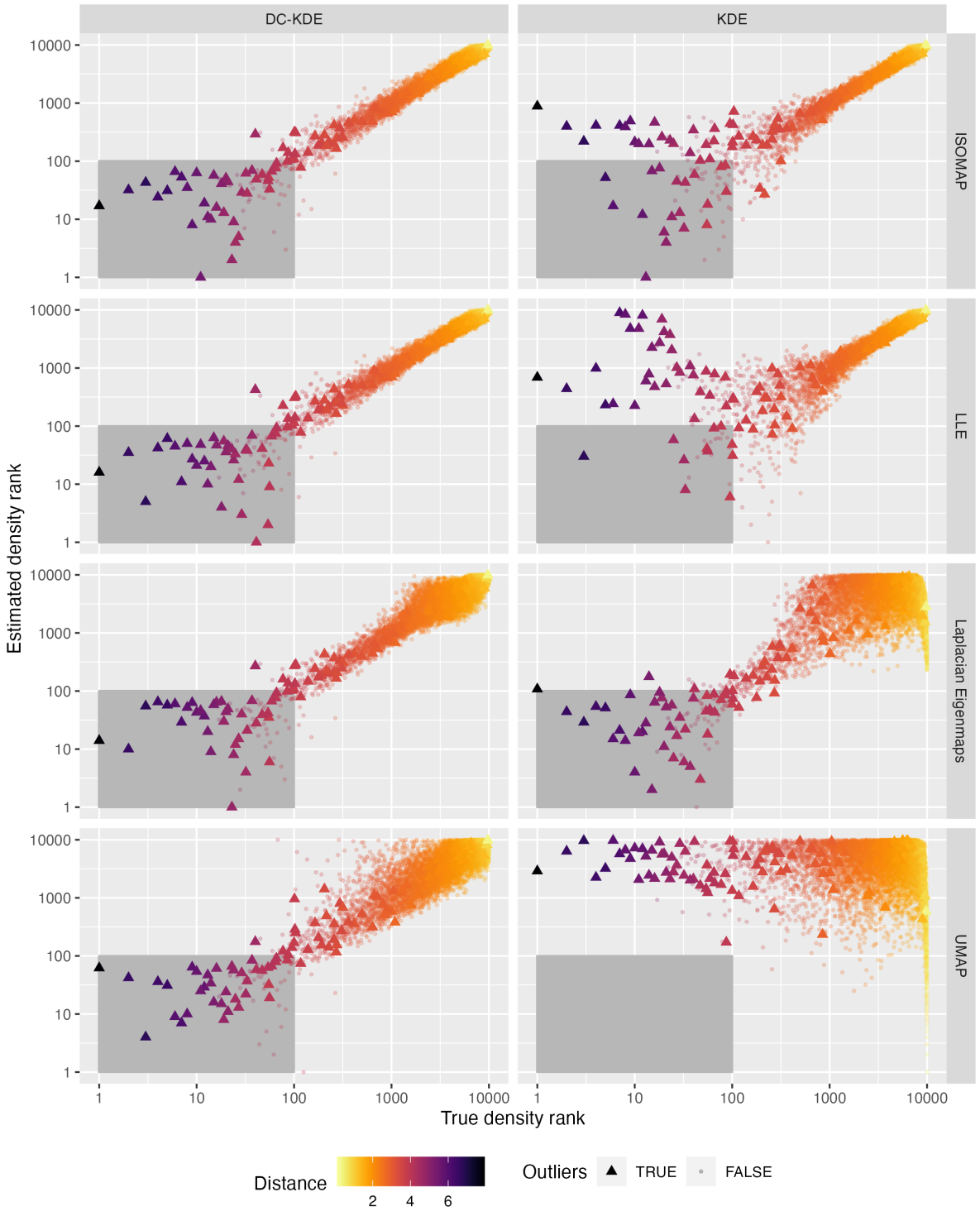

Figure 8: *Rank comparison between the true density and estimated density from both DC-KDE and KDE. Four manifold learning methods are used rowwise. The point shapes indicates whether they are the true outleirs, and the grey shading highlights the top 1% rank region. The colors show the distance to the center of the semisphere, with darker points being distant from the center.*

Table 2: *Correlation between true density and estimated density for four manifold learning embeddings.*

	ISOMAP	LLE	Laplacian.Eigenmaps	UMAP
DC-KDE	0.968	0.970	**0.8674**	**0.782**
KDE	**0.976**	**0.971**	0.0328	-0.181

Table 3: *Percentage comparison of correct highest density regions in density estimation of four manifold learning embeddings.*

	ISOMAP		LLE		Laplacian Eigenmaps		UMAP	
	DC-KDE	KDE	DC-KDE	KDE	DC-KDE	KDE	DC-KDE	KDE
>99	**0.830**	0.490	**0.830**	0.240	**0.840**	0.840	**0.770**	0.000
99	**0.818**	0.685	**0.805**	0.478	**0.815**	0.595	**0.632**	0.032
90	0.919	**0.926**	**0.921**	0.900	**0.798**	0.440	**0.779**	0.428
50	0.825	**0.851**	0.831	**0.848**	**0.659**	0.512	**0.623**	0.448
5	0.508	**0.556**	0.514	**0.562**	**0.431**	0.000	**0.220**	0.000
1	0.080	**0.190**	0.090	**0.230**	**0.000**	0.000	**0.000**	0.000

introducing the pointwise variable Riemannian metric in kernel density estimation, it is reasonable to believe that it could fix the distortion introduced by these three manifold learning algorithms.

4 Application

4.1 Irish smart meter dataset

In this application, we use the smart meter data from the *CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010* in Ireland (Commission for Energy Regulation (CER) 2012) between 14 July 2009 and 31 December 2010. The CER dataset[1] records the half-hourly electricity consumption of individual residential and commercial properties, not including energy for cooling or heating systems. We selected the 3,639 residential data with no missing values during the data collection period for a total of 535 days.

For the electricity consumption data of residential individuals, it would be worthwhile to explore the distribution of electricity demand rather than the raw consumption data to study the usage patterns of different households or different periods or the week (Hyndman, Liu & Pinson 2018). Cheng, Hyndman & Panagiotelis (2021) propose two non-Euclidean distance estimators to enable manifold learning algorithms in statistical manifolds with each observation as a distribution. Cheng, Hyndman & Panagiotelis (2021) use the same smart meter data for identifying outliers with kernel bandwidth estimation but fail to consider the distortion and information loss in the 2-dimensional embeddings given that the input data dimension is much higher. By introducing the Riemannian

[1] accessed via the Irish Social Science Data Archive - www.ucd.ie/issda.

metric as the bandwidth matrix, we could take into account the distortion in the 2-D embedding and further improve the accuracy of the density estimation.

In this section, we first calculated the same empirical distributions of the 336 half-hourly periods of the week for each household, and apply the total variation distance estimator proposed in Cheng, Hyndman & Panagiotelis (2021) in the statistical manifold learning to get the 2-D embedding of all households. Algorithm **??** is then used to obtain density estimates with the pointwise variable Riemannian metric as the bandwidth matrix and detect outliers. The data processing steps have been clearly stated in the application section of Cheng, Hyndman & Panagiotelis (2021) and they are skipped here. Unlike the simulations in Section 3, we know nothing about the true density of the electricity distributions for all periods of the week and all households, so it is impossible to compare the estimated densities with the true meta data density as in Figure 8. However, we could generate all the density estimates with the existing KDE method with fixed bandwidth, which is an optimal method for density estimation, and compare the densities from our proposed method with them.

Figure 9 shows the electricity usage data of three households for both density estimation methods respectively, with the top one being the most typical household with the highest density and the bottom two being the top two outliers with the lowest densities. The typical households in the top row are close except that there are a few spikes for the one with fixed bandwidth. As to the anomalies, variable bandwidth tends to capture the unusual electricity demand volume when the usage is very low or high. It could also capture the unusual usage pattern when there are sudden spikes in ID 3243 or very high base electricity usage for the middle time periods in ID 2838. In contrast, fixed bandwidth KDE is more sensitive to spikes even when the spikes happen in a certain time window in 7049, or when the usage has an obvious time-of-week pattern with a few low electricity usage periods.

Further insights could be gained by comparing the quantile region plots of electricity demand against the time of the week for the same typical or anomalous households in Figure 10 and Figure 11. Again the distribution of both typical households in the top panel has shown a repeated period-of-the-week usage pattern, with higher usage during mealtime on all seven days of the week and slightly higher usage for weekends. However, this repeated pattern in a week window is clearly for the typical household ID 1321. As for the distributions for outliers, the middle row outliers from variable bandwidth have spikes only on Tuesday and Sunday noons, while the fixed bandwidth has an increasing electricity demand across the day of the week. The bottom row outliers both have a repeated time usage pattern, but the electricity usage amount is higher with the highest median above 3kWh. These findings show the difference in finding typical and anomalous households with different bandwidth selections.

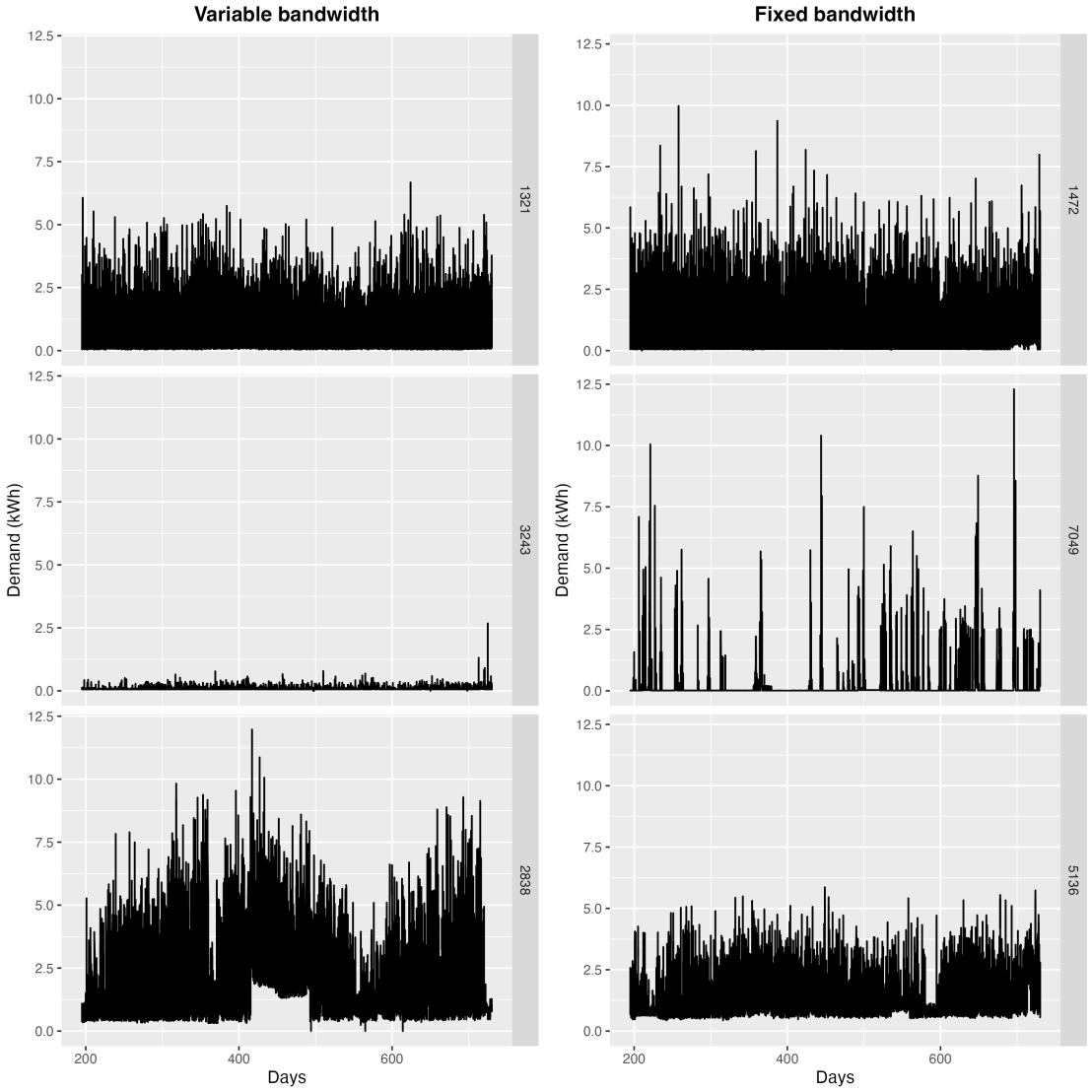

Figure 9: *Electricity usage plots of all 535 days for the most typical household and two anomalies in rows and two bandwidth selection methods in columns.*

5 Conclusion

In this paper, we propose a new method to estimate the density of manifold learning embedding and further identify outliers based on the densities. The Riemannian metric measure the direction and angle of the distortion when mapping data points through a non-linear function in manifold learning algorithms. By introducing the Riemannian metric as the pointwise variable bandwidth matrix in kernel density estimation, the local distortion in the low-dimensional embedding could be used to estimate densities, leading to a more accurate description of the data distributions. We compare our proposed method with fixed bandwidth KDE by two simulation settings, 2-D meta data mapped as a 3-D swiss roll or twin peaks data and 5-D semi-hypersphere mapped in 100-D space, and show that variable bandwidth could improve the density estimation given a good manifold learning embedding.

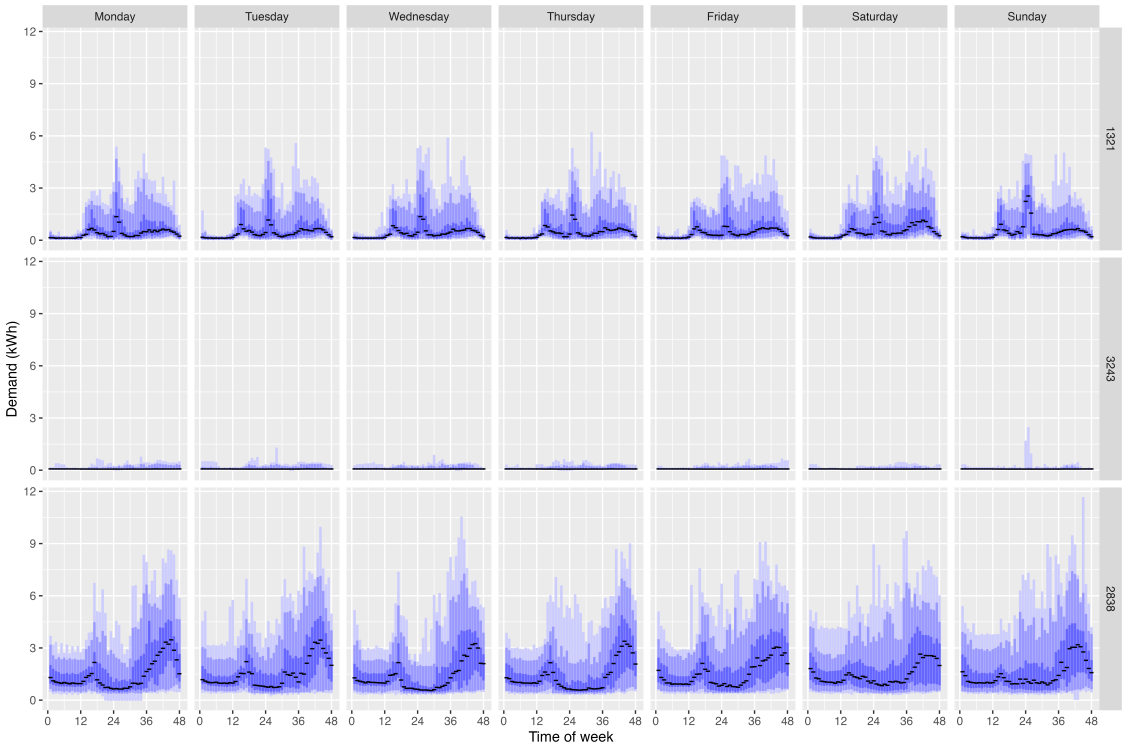

Figure 10: *Two smart-meter demand examples, ID 1003 and ID 1539, from the Irish smart meter data set.*

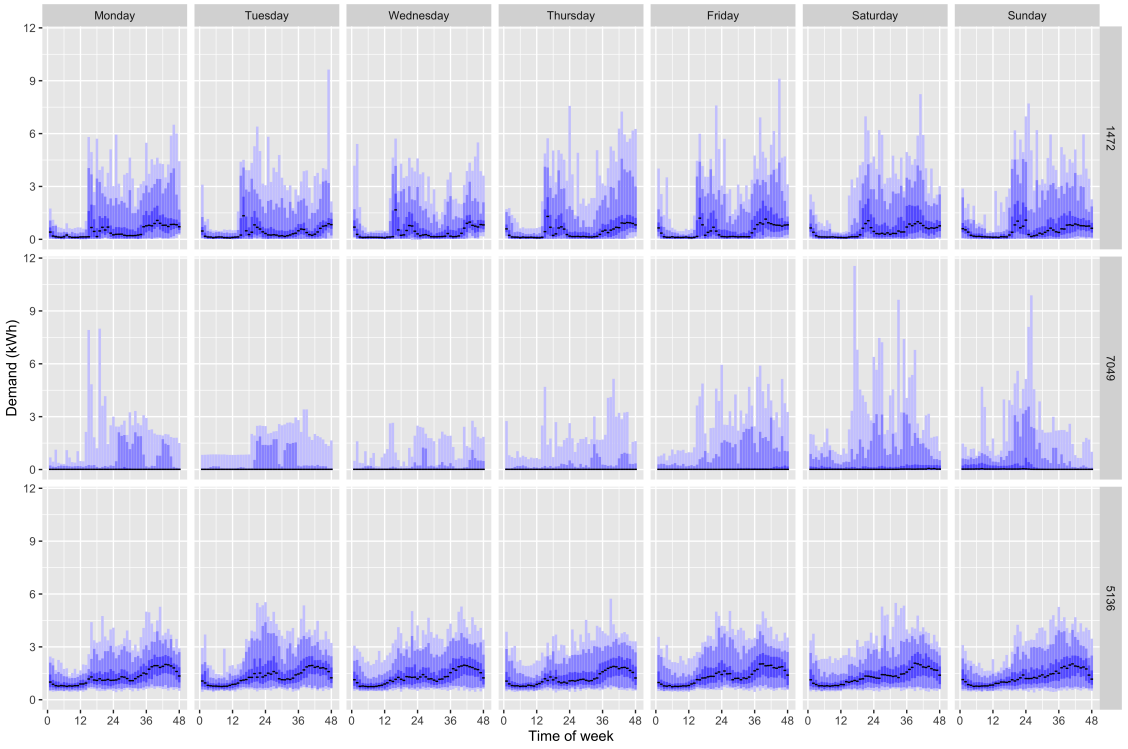

Figure 11: *Two smart-meter demand examples, ID 1003 and ID 1539, from the Irish smart meter data set.*

As an empirical example, we explore the distributions of different households and time periods of the week in the Irish smart meter data. Five manifold learning algorithms, including ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP, are applied to get the 2-D embeddings, and KDE with both variable and fixed bandwidth are used to get the density estimates. We compare both density estimates by looking at the distributions of the most typical households with the highest densities and the most anomalous households with the lowest densities. Both methods could identify the typical households with certain usage patterns, while the outliers are anomalous in different ways.

There are several open questions to be explored. The first involves the selection of tuning parameters for the manifold algorithm so that a maximal level of embedding quality is achieved, where embedding quality is measured using one of the metrics discussed in the online supplementary material of Cheng, Hyndman & Panagiotelis (2021). The scaling of the Riemannian metric to get the closest range of the true densities is also worth exploring. The scale of distortion in each point in manifold learning could vary a lot. If we could smooth across all data points, eg. multiply the Riemannian matrix with the ratio between its determinant and the sum of all Riemannian matrix determinants, the global density estimates could be potentially smoothed. The density estimates on the edges of the whole data structure could be improved because most outer area points tend to be detected as outliers. The choice of manifold learning algorithms also has a large impact on the embedding accuracy, which we have seen will affect the density estimation and outlier detection. However, the outperformance of VKDE with Riemannian bandwidth than the fixed bandwidth has been shown in the higher dimensional simulation data and the electricity usage data, which are more related to real-life data sets.

Acknowledgment

This research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH HPC Cluster. The first author acknowledges the financial support of the Monash Graduate Scholarship (MGS) and the Monash International Tuition Scholarship (MITS) at the Monash University.

A Appendix: Notions about Riemannian geometry

In this appendix, we present some notions about the Riemannian geometry used in this paper.

A.1 Differentiable manifolds

In topology, a *homeomorphism* is a bijective map between two topological spaces that is continuous in both directions. A *Hausdorff space* is a topological space where any two distinct points can be

separated by disjoint neighborhoods. And a d-dimensional (topological) *manifold* M is a connected Hausdorff space $(M, \mathcal{T}_M)$ where the neighborhood U for each point p is homeomorphic to an open subset V of the Euclidean space $\mathbb{R}^d$. Such a homeomorphism $\varphi : U \to V$ together with U gives a (coordinate) *chart*, denoted as (U, φ), with the corresponding local coordinates $(x^1(p), \ldots, x^d(p)) := \varphi(p)$. Further, a collection of charts $\{U_\alpha, \varphi_\alpha\}$ ranging over the manifold M is called an *atlas*, denoted as $\mathcal{A}$.

The manifold M is a *differentiable manifold* if there exists an atlas of M, $\{U_\alpha, \varphi_\alpha\}$, such that the *transition maps* between any two charts,

$$\varphi_\beta \circ \varphi_\alpha^{-1} : \varphi_\alpha(U_\alpha \cap U_\beta) \to \varphi_\beta(U_\alpha \cap U_\beta),$$

are differentiable of class C^∞ (smooth). Let φ be an injective map: $E \to \varphi(E)$. Then φ is an *embedding* of E into M if and only if $\varphi : E \to \varphi(E)$ is a homeomorphism, and $\varphi(E)$ is called an embedded submanifold of M with the subspace topology.

A.2 Tangent vector and tangent space

The tangent vector at point p can be intuitively viewed as the velocity of a curve passing through point p or as the directional derivatives at p. Here we define the tangent vector via the velocity of curves.

For any point $p \in M$, let $\gamma_1 : (-\epsilon_1, \epsilon_1) \to M$ and $\gamma_2 : (-\epsilon_2, \epsilon_2) \to M$ be two smooth curves passing through p, i.e. $\gamma_1(0) = \gamma_2(0) = p$. We say γ_1 and γ_2 are *equivalent* if and only if there exists a chart (U, φ) at p such that

$$(\varphi \circ \gamma_1)'(0) = (\varphi \circ \gamma_2)'(0).$$

A *tangent vector* to a manifold M at point p, denoted as v_p, is any equivalent class of the differentiable curves initialized at p. The set of all tangent vectors at p defines the *tangent space* of M at p, denoted as T_pM. The tangent space is a vector space of dimension d, equal to the dimension of M, and it does not depend on the chart φ locally at p. The collection of all tangent spaces defines the *tangent bundle*, $TM = \cup_{p \in M} T_pM$.

Tangent vectors can also be seen as the directional derivatives at p. For a given coordinate chart $\varphi = (x^1, \ldots, x^d)$, the tangent vectors defining partial derivatives are denoted as $\frac{\partial}{\partial x^1}(p), \ldots, \frac{\partial}{\partial x^d}(p)$, which defines a *basis* of the tangent space. The tangent space T_pM also admits a dual space $T_p^\star M$ called the *cotangent space* with the corresponding *cotangent vectors* $z_p : T_p^\star M \to \mathbb{R}^d$ and a basis denoted as $dx^1(p), \ldots, dx^d(p)$.

A.3 Riemannian metric and geodesic distance

A Riemannian metric g_p defined on the tangent space T_pM at each point p is a local inner product $T_pM \times T_pM \to \mathbb{R}$, where g_p is a $d \times d$ symmetric positive definite matrix and varies smoothly at p. Generally, we omit the subscript p and refer to g as the Riemannian metric. The inner product between two vectors $u, v \in T_pM$ is written as $\langle u, v \rangle_g = g_{ij} u^i v^j$ using the Einstein summation convention where implicit summation over all indices, $\sum_{i,j} g_{ij} u^i v^j$, is assumed. A differentiable manifold M endowed with the Riemannian metric g on each tangent space T_pM is called a *Riemannian manifold* (M, g).

The Riemannian metric g can be used to define the norm of a vector u, $\|u\| = \sqrt{\langle u, v \rangle_g}$, and the angle between two vectors u and v, $\cos\theta = \frac{\langle u,v \rangle_g}{\|u\|\|v\|}$, which are the geometric quantities induced by g. It could also be used to define the line element $dl^2 = g_{ij} dx^i dx^j$ and the volume element $dV_g = \sqrt{\det(g)} dx^1 \dots dx^d$, where $(x^1, \dots, x^d)$ are the local coordinates of the chart (U, φ). For a curve $\gamma : I \to M$, the length of the curve is

$$l(\gamma) = \sqrt{\int_0^1 \|\gamma'(t)\|_g^2 dt} = \sqrt{\int_0^1 g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt} dt},$$

where $\gamma(I) \subset U$. The volume of $W \subset U$ is defined as

$$Vol(W) = \int_W \sqrt{\det(g)} dx^1 \cdots dx^d,$$

which is also called the *Riemannian measure* on M.

The *geodesics* of M are the smooth curves that locally joins the points along the shortest path on the manifold. Intuitively, geodesics are the *straightest possible curves* in a Riemannian manifold (Section 7.2.3 of Nakahara 2018). A curve $\gamma : I \to M$ is a geodesic if for all indices i, j, k, the second-order ordinary differential equation is satisfied,

$$\frac{d^2 x^i}{dt^2} + \Gamma^i_{jk} \frac{dx^j}{dt} \frac{dx^k}{dt} = 0,$$

where $\{x^i\}$ are the coordinates of the curve γ and Γ^i_{jk} is the *Christoffel symbol* defined by

$$\Gamma^i_{jk} = \frac{1}{2} \sum_l g^{il} \left(\frac{\partial g_{il}}{\partial x^k} + \frac{\partial g_{kl}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^l} \right).$$

The geodesics have a constant speed with norm $\|\gamma'(t)\|$, and they are the local minimizers of the arc length functional $l : \gamma \to \sqrt{\int_0^1 \|\gamma'(t)\|_g^2 dt}$ when the curves are defined over the interval $[0, 1]$. The *geodesic distance* d_g is the length of the shortest geodesic between two points on the manifold.

For a point $p \in M$, when the geodesic distance starting at p is not minimized, we call such set of points the *cut locus* of p, and the distance to the cut locus is the *injectivity radius* at $p \in M$. Therefore, the injectivity radius of the Riemannian manifold (M, g), $inj_g M$, is the infimum of the injectivity radii over all points on the manifold.

A.4 Exponential map and logarithmic map

Denote $B(p, r) \subset M$ as an open ball centered at point p with raidus r. Then $B(0_p, r) = exp_p^{-1}(B(p, r))$ is an open neighborhood of 0_p in the tangent space at p, $T_p M$, where exp_p is the *exponential map* at point p. The exponential map maps a tangent vector $u \in B(0_p, r)$ to the endpoint of the geodesic $\gamma : I \to M$ satisfying $\gamma(0) = p, \gamma'(0) = u$, and $\gamma(1) = exp_p(u)$. It is a differentiable bijective map of differentiable inverse (i.e. *diffeomorphism*). Intuitively, the exponential map moves point p to an endpoint at speed u after covering the length of $\|u\|$ along the geodesic in one time unit.

The inverse of the exponential map is called the *logarithm map*, denoted as $\log_p(q) := \exp_p^{-1}(q)$, which gives the tangent vector to get from point p to q in one unit time. Also define the *geodesic ball* centered at p of radius $r > 0$ as the image by the exponential map of $B(0_p, r) \subset T_p M$ with $r < inj_g M$. Then we could interpolate a geodesic γ between two points p and q with the exponential map and the logarithmic map, $\gamma(t) = \exp_p(t \log_p(q))$, and the geodesic distance is given by $d_g(p, q) = \| \log_p(q) \|_g$.

A.5 Pushforward and pullback metric

Pushforward and pullback are two notions corresponding to the notions of tangent and cotangent vectors. Let $\phi : M \to E$ be a smooth map between the Riemannian manifold (M, g) to another smooth manifold E. Then the differential of ϕ at point p is a linear map $d\phi_p : T_p M \to T_{\phi(p)} E$, which pushes the tangent vector $u \in T_p M$ at point p forward to the tangent vector $\phi_* u \in T_{\phi(p)} E$ at the mapping point $\phi(p)$. The image of the tangent vector $u \in T_p M$ under the differential $d\phi_p$, denoted as $d\phi_p u$ is called the pushforward of u by the map ϕ. Then pushforward metric $h = \varphi_* g$ of the Riemannian metric g along φ is given by the inner product

$$\langle \phi_* u, \phi_* v \rangle_{\varphi * g} = \langle d\phi_p \phi_* u, d\phi_p \phi_* v \rangle_g.$$

The tangent vectors $\phi_* u$ are equivalent to the velocity vector of a curve $\gamma : I \to M$ passing through point p at time zero with a constant speed $\gamma'(0) = u$,

$$d\phi_p(\gamma'(0)) = (\phi \circ \gamma)'(0).$$

Similarly, the pullback maps the cotangent vectors $z_{f(p)}$ at $f(p) \in E$ to cotangent vectors at $p \in M$ acting on tangent vectors $u \in T_pM$. The linear map is called the pullback by ϕ and is often denoted as ϕ^*.

B Appendix: Rank comparison plots for twin peaks mapping

This appendix contains the comparison plots for the density rank between DC-KDE and KDE using different manifold learning algorithms, similar to Figure 4, Figure 5, and Figure 6. By comparing these plots, it could be concluded that DC-KDE could categorize the density ranks into highest density regions more accurately than KDE. By correcting the distortion in different manifold learning embeddings, DC-KDE is more robust in identifying the lowest density regions, which are usually used to detect anomalies.

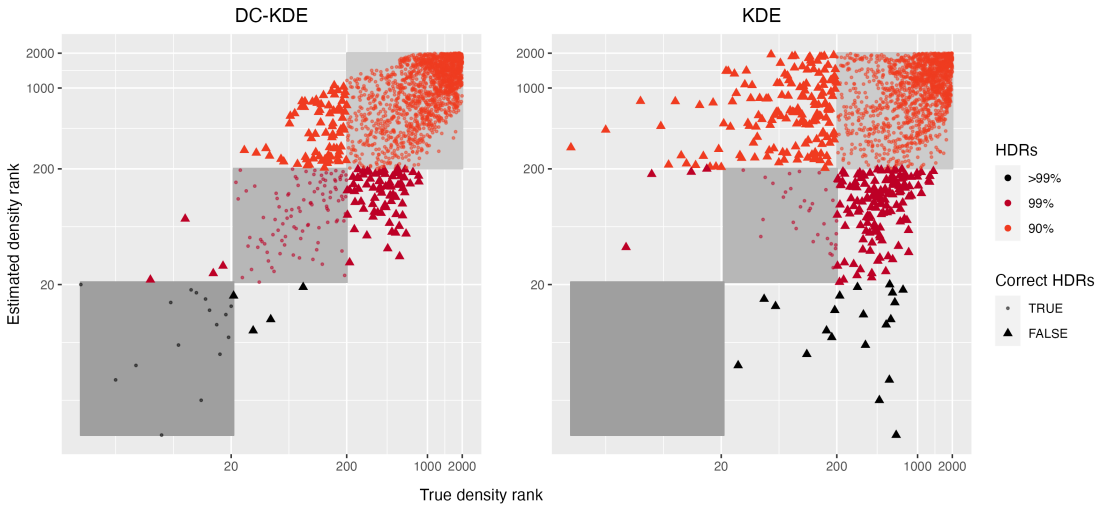

Figure 12: *Scatterplot of true density and estimated density ranks of LLE embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.*

C Appendix: Simulation with swiss roll mapping

In this appendix, we demonstrate the simulation results for the data in Section 3.1 with the swiss roll mapping.

One of the most famous examples in manifold learning is the swiss roll data, with the mapping function in (2). The two-dimensional meta data $(\boldsymbol{X}_1, \boldsymbol{X}_2)'$ is transformed into the three-dimensional data $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})'$, shown in the left plot of **??**. The four colors in the mappings represent the four

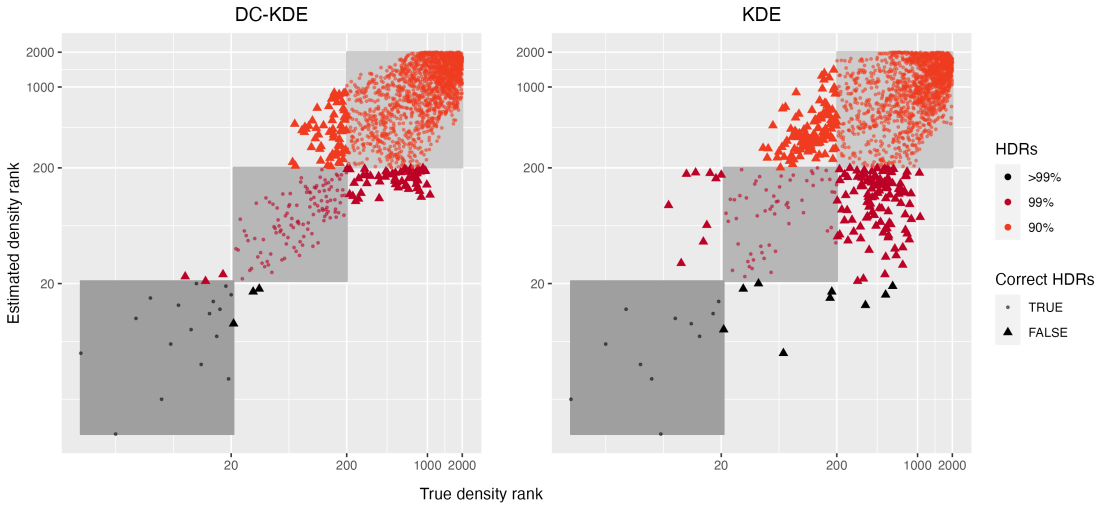

Figure 13: *Scatterplot of true density and estimated density ranks of Laplaxian Eigenmaps embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.*

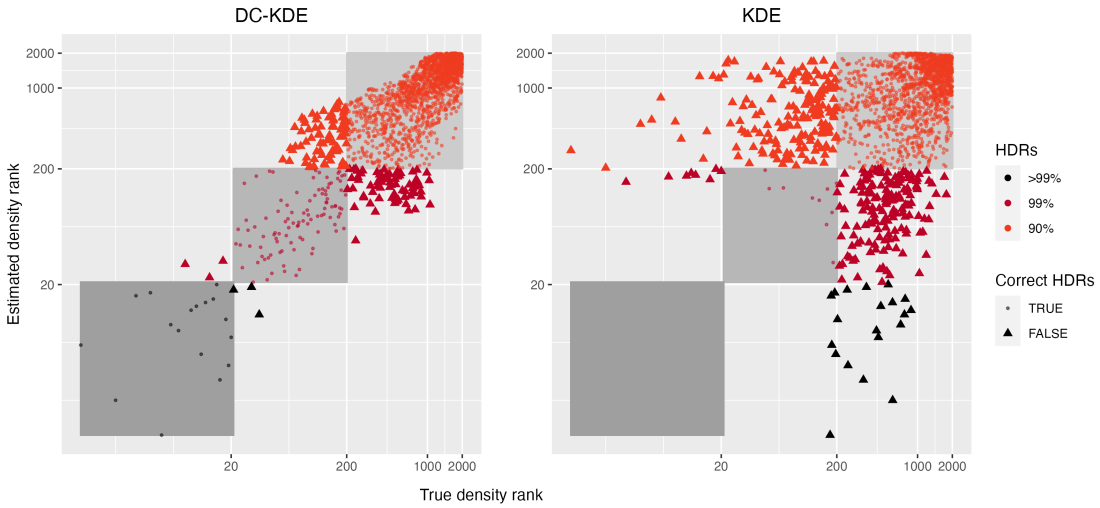

Figure 14: *Scatterplot of true density and estimated density ranks of UMAP embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.*

Gaussian kernels used to generate the meta data $(\boldsymbol{X}_1, \boldsymbol{X}_2)'$.

$$\begin{cases} X = X_1 \cos X_1, \\ Y = X_2, \\ Z = X_1 \sin X_1. \end{cases} \tag{2}$$

Now we are able to apply different manifold learning algorithms to $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})'$ and reduce the dimension back to $d = 2$, and further estimate the density of the 2-D embedding. According to

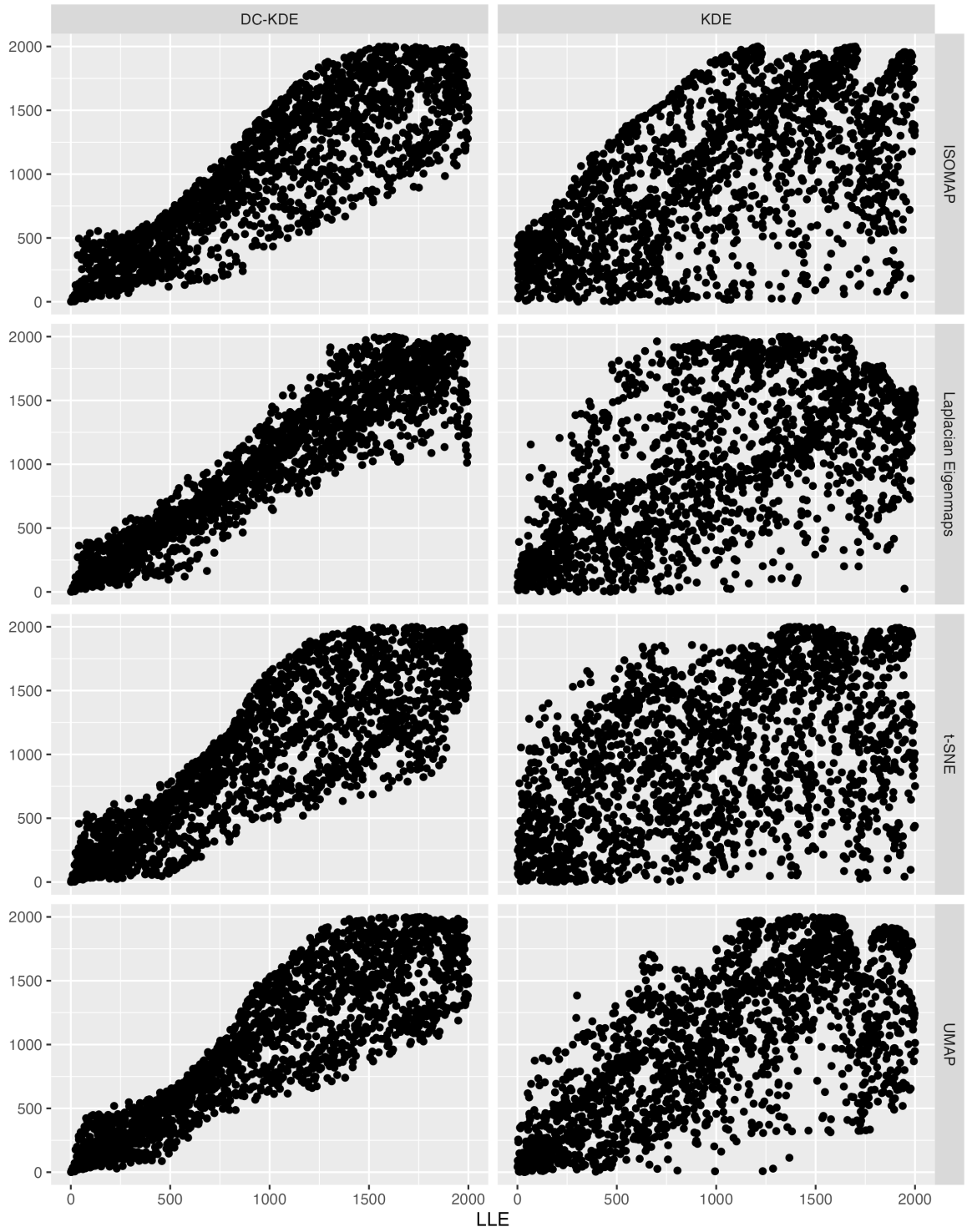

Figure 15: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless of the manifold learning embedding.*

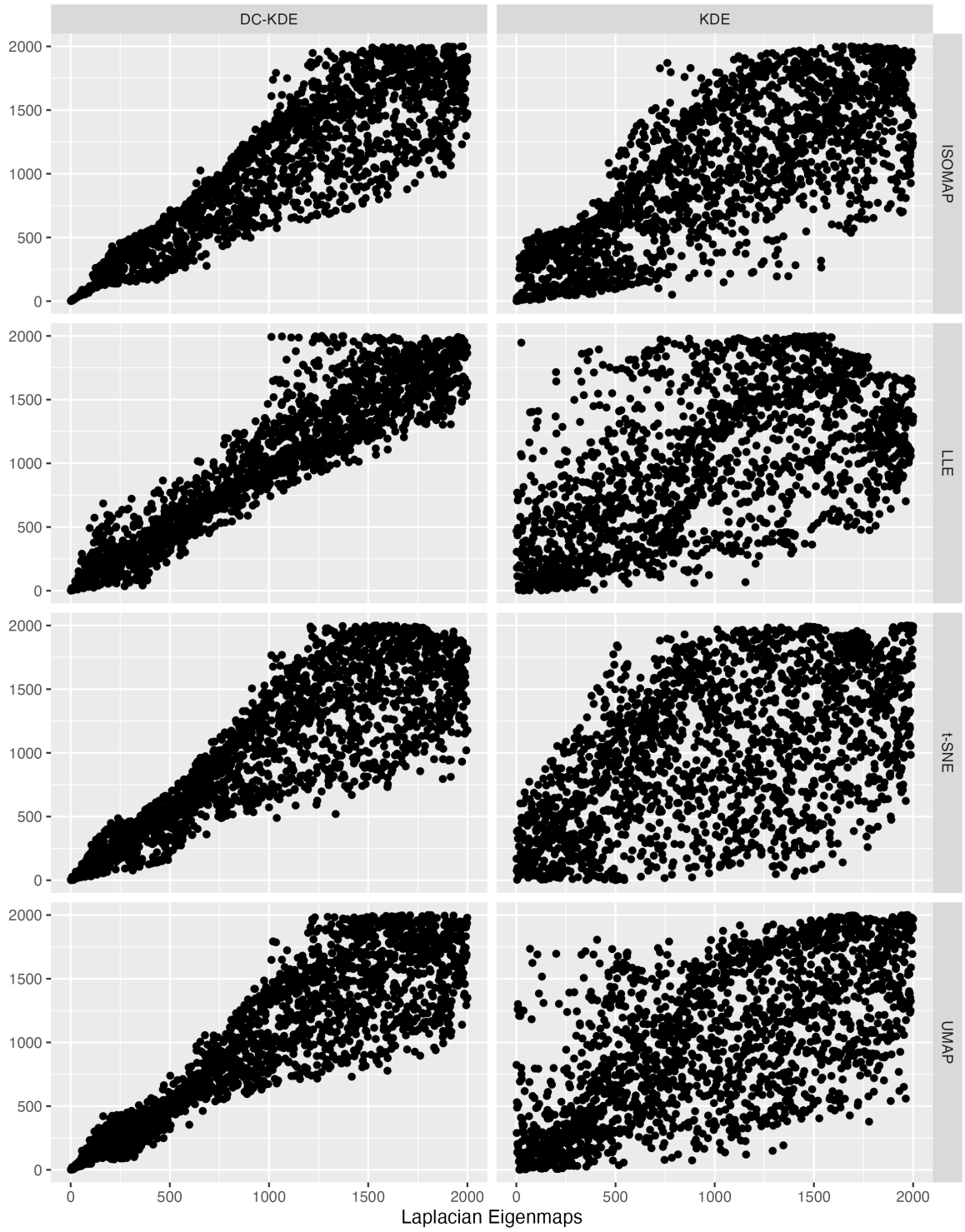

Figure 16: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless of the manifold learning embedding.*

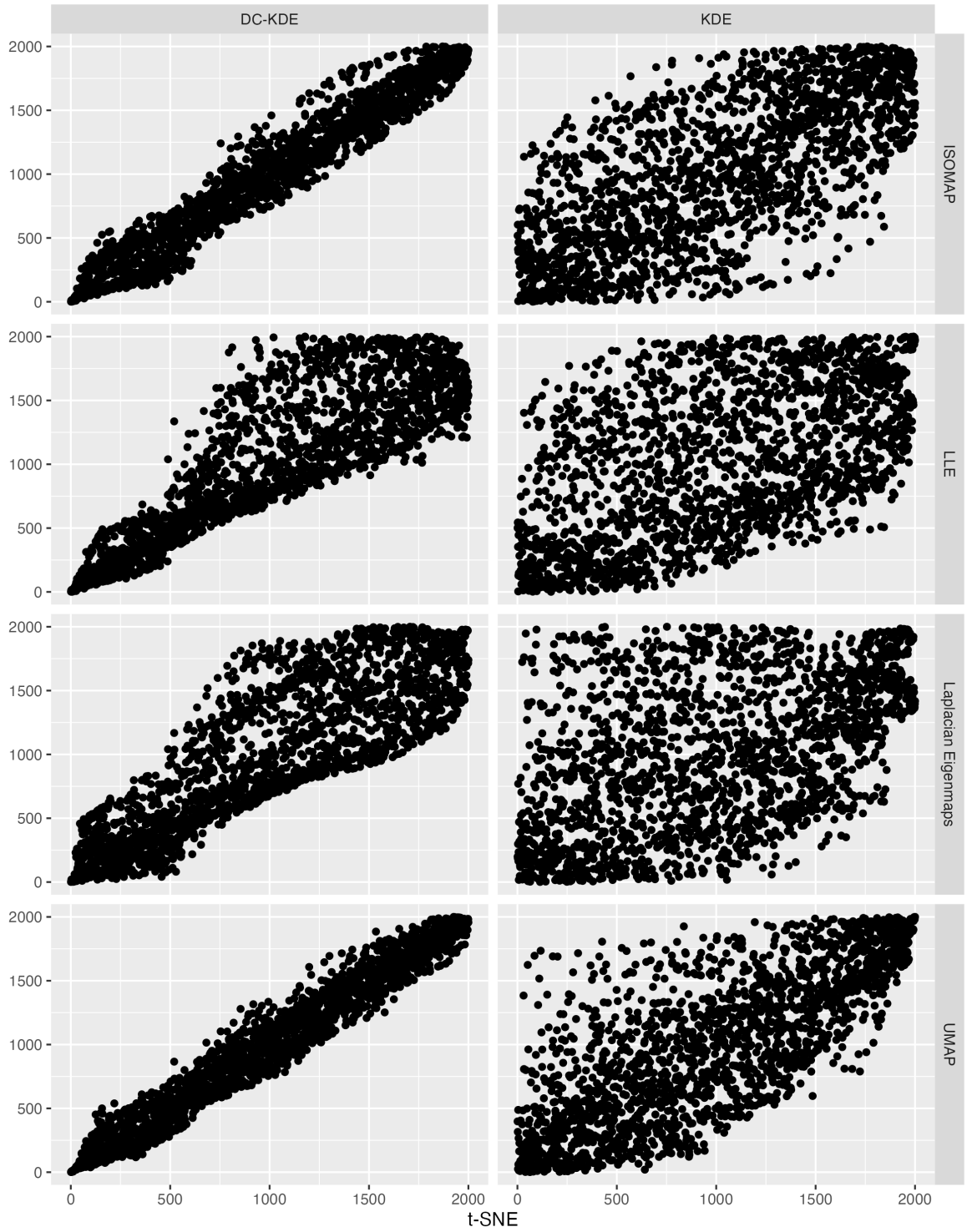

Figure 17: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless of the manifold learning embedding.*

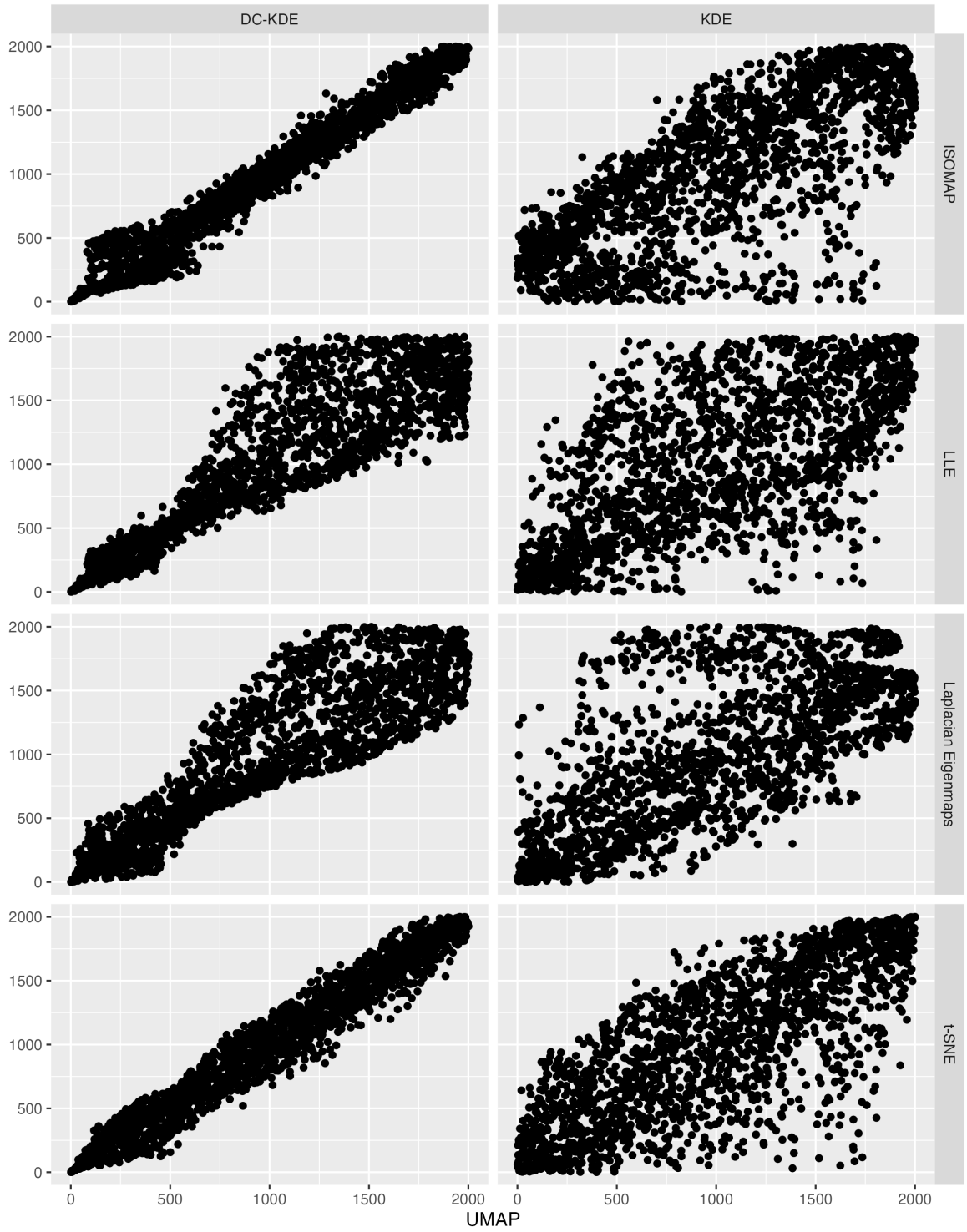

Figure 18: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless of the manifold learning embedding.*

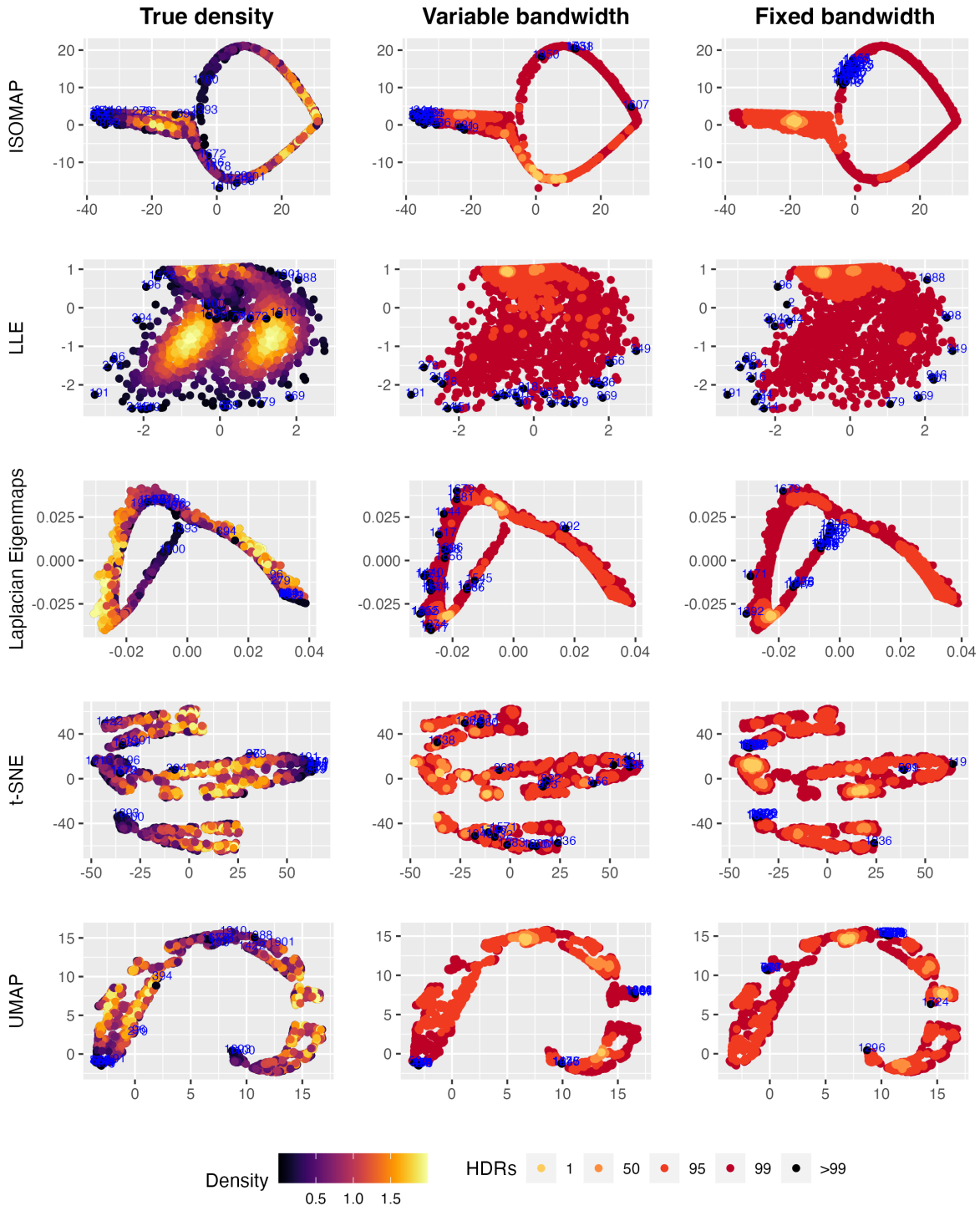

Figure 19: *Highest density region plots of five manifold learning embeddings of the swiss roll data. Colors are indicating densities from left: true densities from the Gaussian mixture model; middle: KDE with Riemannian matrix as variable bandwidth; and right: KDE with fixed bandwidth. Variable KDE performs better in finding kernel structures with ISOMAP, LLE, and Laplacian Eigenmaps, and in locating outliers with ISOMAP and LLE. The t-SNE and UMAP embeddings are highly distorted and the outliers found are clustered.*

Table 4: *Correlation between true density ranking and estimated density ranking for different manifold learning embeddings of the swiss roll data. Variable bandwidth KDE outperfoms for LLE and UMAP, and LLE gives the highest rank correlation.*

	ISOMAP	LLE	Laplacian.Eigenmaps	t.SNE	UMAP
Variable bandwidth	0.0696	**0.400**	-0.2357	0.023	**0.0138**
Fixed bandwidth	**0.2798**	0.351	**0.0141**	**0.367**	-0.0110

the density estimates, we could rank the data points and then identify which observations lie in the highest density region of specified coverage, eg. 1%, 5%, 50%, 99%, >99%. For each of the five manifold learning methods, namely ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP, Figure 19 presents the 2-D embedding plot in the same row, with the colors indicating the densities levels, the left column for true densities from the Gaussian mixture model, the middle column for highest density region plots with densities from our proposed variable KDE method, and the right for similar HDR plots with densities from KDE with fixed bandwidth. The top twenty outliers with the lowest densities are highlighted in black with point indexes in blue. From Figure 2 and the data generating process, we know that there are four highest density regions. However, in all manifold learning embeddings colored with true densities (left column in Figure 19), except for LLE, the number of highest density regions are not the same as the meta data. When comparing the number of HDRs for variable and fixed bandwidth (middle and right column in Figure 19), our proposed method with variable bandwidth outperforms fixed bandwidth for ISOMAP, LLE, and Laplacian Eigenmaps (top three rows in Figure 19). In terms of the top 20 outliers found rowwise, variable bandwidth could find most outliers lying on the left area of the embedding in ISOMAP and UMAP, and both methods in LLE embedding could find the outliers in the outer area, but for the other methods, both variable and fixed bandwidth are not detecting true outliers accurately. For t-SNE and UMAP embedding, the embedding structure is highly distorted and the points are clustered together in a discontinuous way, which is also shown in the clustered outliers.

To further compare the accuracy of the estimated densities for all data points, we calculate the correlation between the rank of true densities and the estimated densities and present in Table 4. It can be seen that the rank correlation of our proposed method with variable bandwidth is higher for LLE and UMAP, although the correlation for UMAP is very close to zero. The highest correlation comes from our method in LLE embedding, which is mainly due to it being closest to the rectangular structure of the meta data shown in Figure 2. For Laplacian Eigenmaps, our method has wrongly estimated the left area with lower densities even though their true densities are very high in yellow, leading to a negative correlation. The negative correlation would occur typically when the highest or lowest true density areas are not well estimated. As for the estimates in highly distorted embedding, including ISOMAP, t-SNE, and UMAP, the rank correlations are quite low. This shows that our

proposed method could improve the kernel density estimate of manifold learning embedding by considering the distortion using the Riemannian metric. However, if the distortion is too severe, eg. ISOMAP, or when the embedding is discontinuous, eg. t-SNE and UMAP, the density estimates are not as reliable for outlier detection.

References

Breiman, L, W Meisel & E Purcell (1977). Variable Kernel Estimates of Multivariate Densities. *Technometrics: a journal of statistics for the physical, chemical, and engineering sciences* **19**(2), 135–144. https://www.tandfonline.com/doi/abs/10.1080/00401706.1977.10489521.

Brigant, A le & S Puechmorel (2019). Approximation of Densities on Riemannian Manifolds. en. *Entropy* **21**(1).

Cao, R, A Cuevas & W González Manteiga (1994). A comparative study of several smoothing methods in density estimation. *Computational statistics & data analysis* **17**(2), 153–176. https://www.sciencedirect.com/science/article/pii/016794739200066Z.

Chacón, JE & T Duong (2010). Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices. *Test* **19**(2), 375–398. https://doi.org/10.1007/s11749-009-0168-4.

Chavel, I (2006). *Riemannian Geometry: A Modern Introduction*. en. Cambridge University Press, p. 108.

Chen, YC (2017). A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology* **1**(1), 161–187.

Cheng, F, RJ Hyndman & A Panagiotelis (2021). Manifold Learning with Approximate Nearest Neighbors. (3/21).

Commission for Energy Regulation (CER) (2012). *CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset]*. SN: 0012-00.

Denti, F (2021). intRinsic: an R package for model-based estimation of the intrinsic dimension of a dataset. arXiv: 2102.11425 [stat.CO].

Denti, F, D Doimo, A Laio & A Mira (2021). Distributional Results for Model-Based Intrinsic Dimension Estimators. arXiv: 2104.13832 [stat.ME].

Duong, T (2004). Bandwidth selectors for multivariate kernel density estimation. https://www.mvstat.net/tduong/research/publications/duong-2005-thesis.pdf.

Duong, T & M Hazelton (2003). Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of nonparametric statistics* **15**(1), 17–30.

Elgammal, A, R Duraiswami, D Harwood & LS Davis (2002). Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE* **90**(7), 1151–1163. http://dx.doi.org/10.1109/JPROC.2002.801448.

Facco, E, M d'Errico, A Rodriguez & A Laio (2017). Estimating the intrinsic dimension of datasets by a minimal neighborhood information. en. *Scientific reports* **7**(1), 12140.

Gerber, MS (2014). Predicting crime using Twitter and kernel density estimation. *Decision support systems* **61**, 115–125. https://www.sciencedirect.com/science/article/pii/S0167923614000268.

Goldberg, Y, A Zakai, D Kushnir & Y Ritov (2008). Manifold Learning: The Price of Normalization. *J. Mach. Learn. Res.* **9**(Aug), 1909–1939.

Heidenreich, NB, A Schindler & S Sperlich (2013). Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *AStA. Advances in Statistical Analysis. A Journal of the German Statistical Society* **97**(4), 403–433. https://doi.org/10.1007/s10182-013-0216-y.

Henry, G & D Rodriguez (2009). Kernel Density Estimation on Riemannian Manifolds: Asymptotic Results. *Journal of mathematical imaging and vision* **34**(3), 235–239. https://doi.org/10.1007/s10851-009-0145-2.

Hyndman, RJ, X Liu & P Pinson (2018). Visualizing big energy data: Solutions for this crucial component of data analysis. *IEEE Power Energ. Mag.*

Hyndman, RJ (1996). Computing and Graphing Highest Density Regions. *Am. Stat.* **50**(2), 120–126.

Jeon, J & JW Taylor (2012). Using Conditional Kernel Density Estimation for Wind Power Density Forecasting. *Journal of the American Statistical Association* **107**(497), 66–79. https://doi.org/10.1080/01621459.2011.643745.

Jones, MC (1990). Variable kernel density estimates and variable kernel density estimates. en. *The Australian journal of statistics* **32**(3), 361–371.

Jones, MC, JS Marron & SJ Sheather (1992). *Progress in data-based bandwidth selection for kernel density estimation*. http://www.springer.com/statistics/journal/180.

Jones, MC, JS Marron & SJ Sheather (1996). A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the American Statistical Association* **91**(433), 401–407. https://www.tandfonline.com/doi/abs/10.1080/01621459.1996.10476701.

Jones, MC & R. F. Kappenman (1992). On a Class of Kernel Density Estimate Bandwidth Selectors. *Scandinavian journal of statistics, theory and applications* **19**(4), 337–349.

McQueen, J, M Meilă, J VanderPlas & Z Zhang (2016). Megaman: Scalable Manifold Learning in Python. *J. Mach. Learn. Res.* **17**(148), 1–5.

Nakahara, M (2018). *Geometry, topology and physics*. taylorfrancis.com. https://www.taylorfrancis.com/books/mono/10.1201/9781315275826/geometry-topology-physics-mikio-nakahara.

Okabe, A, T Satoh & K Sugihara (2009). A kernel density estimation method for networks, its computational method and a GIS-based tool. en. *Geographical Information Systems* **23**(1), 7–32. https://www.tandfonline.com/doi/abs/10.1080/13658810802475491.

Parzen, E (1962). On Estimation of a Probability Density Function and Mode. *Annals of Mathematical Statistics* **33**(3), 1065–1076.

Pelletier, B (2005). Kernel density estimation on Riemannian manifolds. *Statistics & probability letters* **73**(3), 297–304.

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

Sain, SR, KA Baggerly & DW Scott (1994). Cross-Validation of Multivariate Densities. *Journal of the American Statistical Association* **89**(427), 807–817.

Scott, DW (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization*. en. John Wiley & Sons. https://play.google.com/store/books/details?id=pIAZBwAAQBAJ.

Terrell, GR & DW Scott (1992). Variable Kernel Density Estimation. *Annals of statistics* **20**(3), 1236–1265.

Wand, MP & MC Jones (1994). *Kernel Smoothing*. en. CRC Press. https://play.google.com/store/books/details?id=GTOOi5yE008C.

Xie, Z & J Yan (2008). Kernel Density Estimation of traffic accidents in a network space. *Computers, environment and urban systems* **32**(5), 396–406. https://www.sciencedirect.com/science/article/pii/S0198971508000318.

Zhou, X & M 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 and Conference Proceedings, pp.892–900.