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

31 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, Panagiotelis & Hyndman (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, Panagiotelis & Hyndman (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 Section 2, 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

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. Since some readers may be unfamiliar with the nuances of manifolds, we first discuss kernel density estimation for data in Euclidean space, then illustrate in Section 2.1 how

this generalizes to the estimator of Pelletier (2005), when the data lie on some known manifold. In Section 2.2, we describe the Learn Metric 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 M as 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 $m \gg d$, however this need not be the case. For instance, $\boldsymbol{p}_i$ 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 $\|\cdot\|$ be the usual Euclidean norm and following Pelletier (2005), we make these assumptions about the kernel function $K : \mathbb{R}_+ \to \mathbb{R}$,

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

Note that these conditions are different from (and in some cases stricter than) 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 $\boldsymbol{y}_i \in \mathbb{R}^d$ with $i = 1, \ldots, N$ 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.

Kernel density estimators of this form can and have been applied directly on the output embedding $\boldsymbol{y}$, and we will consider this approach as a benchmark in Section 3. As a non-linear transformation, any dimension reduction algorithm 'distorts' the density. To make this clear consider the simpler case computing the density after a change of variables $\psi : \mathbb{R}^d \to \mathbb{R}^d$, which involves a Jacobian term. A similar notion applies to a manifold embedding so that the density of the output vectors $\boldsymbol{y}_i$ differs from the density on the manifold itself. Furthermore, standard kernel density estimates applied directly on the output embedding will be sensitive to the choice of dimension reduction algorithm since each different algorithm will distort the density in its own. This motivates a kernel desnity estimate that corrects for the distortion induced by ψ.

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 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 the inclusion of 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 on the manifold, $\boldsymbol{q} \in M$. Loosely, $\boldsymbol{v}$ 'points' in the direction of the geodesic between $\boldsymbol{p}$ and $\boldsymbol{q}$ and travelling along this geodesic at uniform speed $\|\boldsymbol{v}\|$ takes place in one unit of time. Now, consider a chart φ mapping points in the neighborhood 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 the defined tangent vectors, the Riemannian metric and the geometric quantities are invariant to any specific choice of coordinates, they can still be expressed in terms of local coordinate 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 distortions. 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*. Two parameters, $c = 0.25$ and $\sqrt{\varepsilon} = 0.4$, are set as suggested in the *megaman* library.

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 case, 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

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

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 $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$. Second, the estimator in Equation (1) is valid only when the coordinate mapping is the logarithmic map around $\boldsymbol{y}_i$, and 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 correspond to observed data, 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 summarized in Figure 1. The last two steps in Figure 1 are our main contributions, 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, Panagiotelis & Hyndman (2021), the changes are also highlighted in blue. With this anomaly detection process, outliers based on lowest densities could be detected more accurately regardless of the distortion in manifold learning.

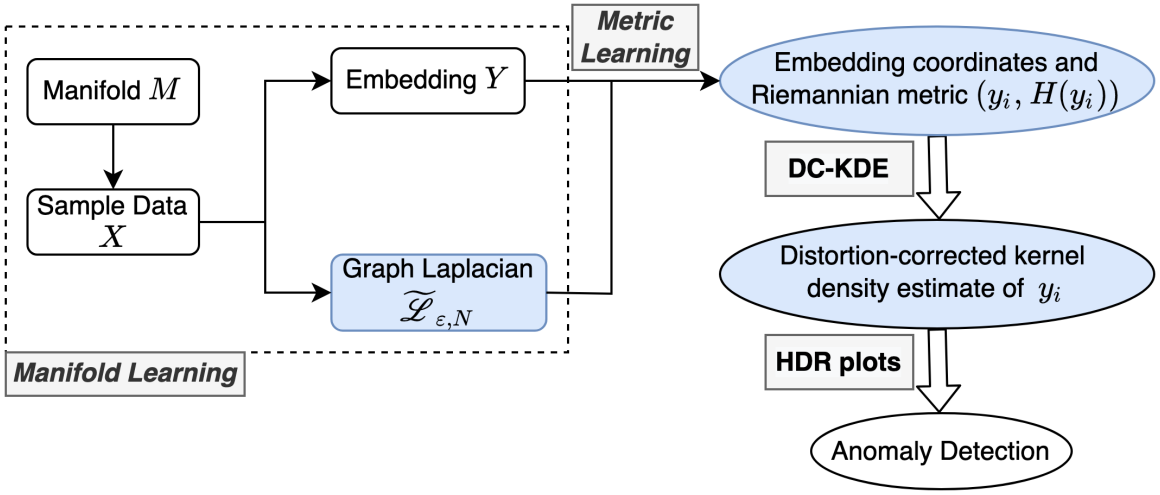

Figure 1: *The proposed schematic for anomaly detection with distortion corrected kernel density estimates.*

3 Simulations

In this section, we examine two scenarios for both low and high dimensions to test our proposed distortion corrected KDE. For visualization purposes, Section 3.1 presents an example of a two-dimensional manifold embedded in 3-dimensional ambient Euclidean space. As a high-dimensional example, the second simulation in Section 3.2 is based on a 4-dimensional manifold embedded in a 100-dimensional ambient space. To estimate the density, we use the dimension reduction algorithms ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP. In general, we aim to highlight two advantages of our proposed distortion corrected KDE compared to KDE applied directly to the output coordinates. First, the density estimates are closer to the ground truth when distortion correction is used, and as a consequence, distortion correction is more adept at detecting anomalies. Second, we show how density estimation and anomaly detection are more robust to a different choice of dimension reduction method when distortion correction is used.

3.1 Twin peaks example

The simulation setup for the twin peaks example is to first generate vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ 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 following

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

The three-dimensional twin peaks mapping is shown in Figure 3. The colors in both Figure 2 and Figure 3 indicate the true density of the data via the twin peaks mapping, with lower density points in darker colors scattered in the outer as well as center areas.

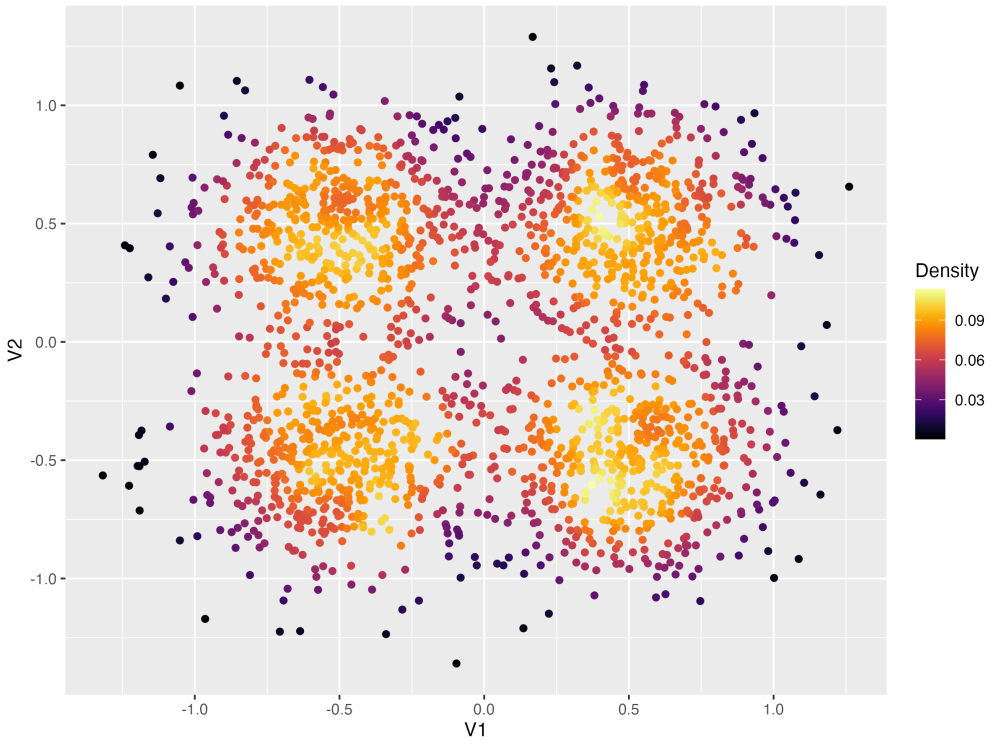

Figure 2: *Underlying data for the Gaussian mixture model of four kenels with means* $(0.25, 0.25), (0.25, 0.75), (0.75, 0.25), (0.75, 0.75)$ *and the same variance-covariance matrix* $diag(0.016, 0.016)$. *The colors indicate the true density of the data when they are mapped via the twin peaks function. Lower density points in darker colors are scattered both in the outer and center areas.*

It is important to note that the *true density* on the manifold is not simply a Gaussian mixture, since the mapping in Equation (3) distorts the distribution. To recover the true distribution requires the correct Jacobian term for the pushforward from $\boldsymbol{v}$ to the volume form of twin peaks manifold. By treating the $\boldsymbol{v}$ as an 'output' embedding from input points $\boldsymbol{x}$ that lie on the true manifold and applying the Learn Metric algorithm, we can obtain $\boldsymbol{\Gamma}_i$ for $i = 1, \ldots, n$ where $\boldsymbol{\Gamma}_i$ is the Riemannian of the coordinate system given by $\boldsymbol{v}$. This notation is distinct from $\boldsymbol{H}_i$ which is the output of the Learn Metric algorithm for a coordinate system obtained via a dimension reduction algorithm. The true density on the manifold can be obtained as $f(\boldsymbol{p}_i) = f(\boldsymbol{v}_i)|\boldsymbol{\Gamma}_i|^{1/2}$, where $f(\boldsymbol{v}_i)$ is a the density of

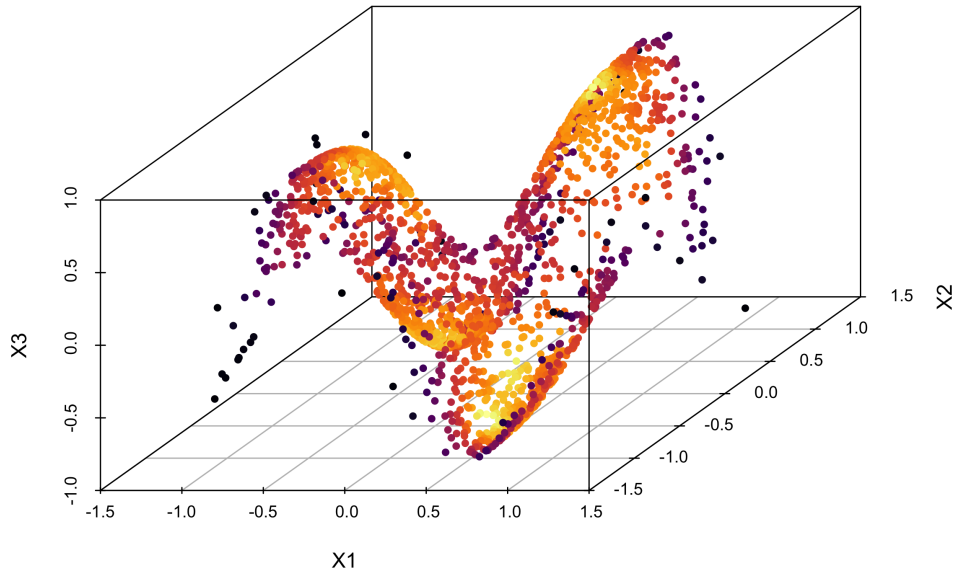

Figure 3: *Scatterplot of the 3-d twin peaks data with the same colors indicating the true density as in Figure 2.*

a four component mixture of normals. Knowledge of $\boldsymbol{v}$ and $\boldsymbol{\Gamma_i}$ will not be used when estimating the density but only to establish a 'ground truth' for densities on the manifold. Figure 2 shows the simulated $\boldsymbol{v}$ with color indicating the true density of data on the manifold. Anomalies are defined as points with the lowest densities shown in darker colors and with 'typical' points having the highest density shown in yellow. The anomalies are found around the edges of the plot, but there are is also a low density region between the means of the four mixture components. The objective is to determine whether we can correctly identify these features without any knowledge of the true density or the $\boldsymbol{v}$.

Figure 4 summarizes the results. Each row of panels corresponds to a different dimension reduction technique, while the left, center and right columns correspond to density estimates for the ground truth density, distortion corrected KDE and KDE respectively. We set the bandwidth parameter in the DC-KDE (2) as $r = 0.5$. Colors show the different estimated density at each point with anomalies shown in black with blue indexing, and higher density points shown in yellow. For many methods, the salient features of the ground truth distribution are clear regardless of whether distortion correction is applied, for example for ISOMAP, all three plots, identify a similar set of outliers and four high density regions. On the other hand for LLE, the left panel shows that dimension reduction pulls outliers on the manifold in towards the centre. The distortion corrected KDE can account for this, while KDE without distortion correction on the other hand does not

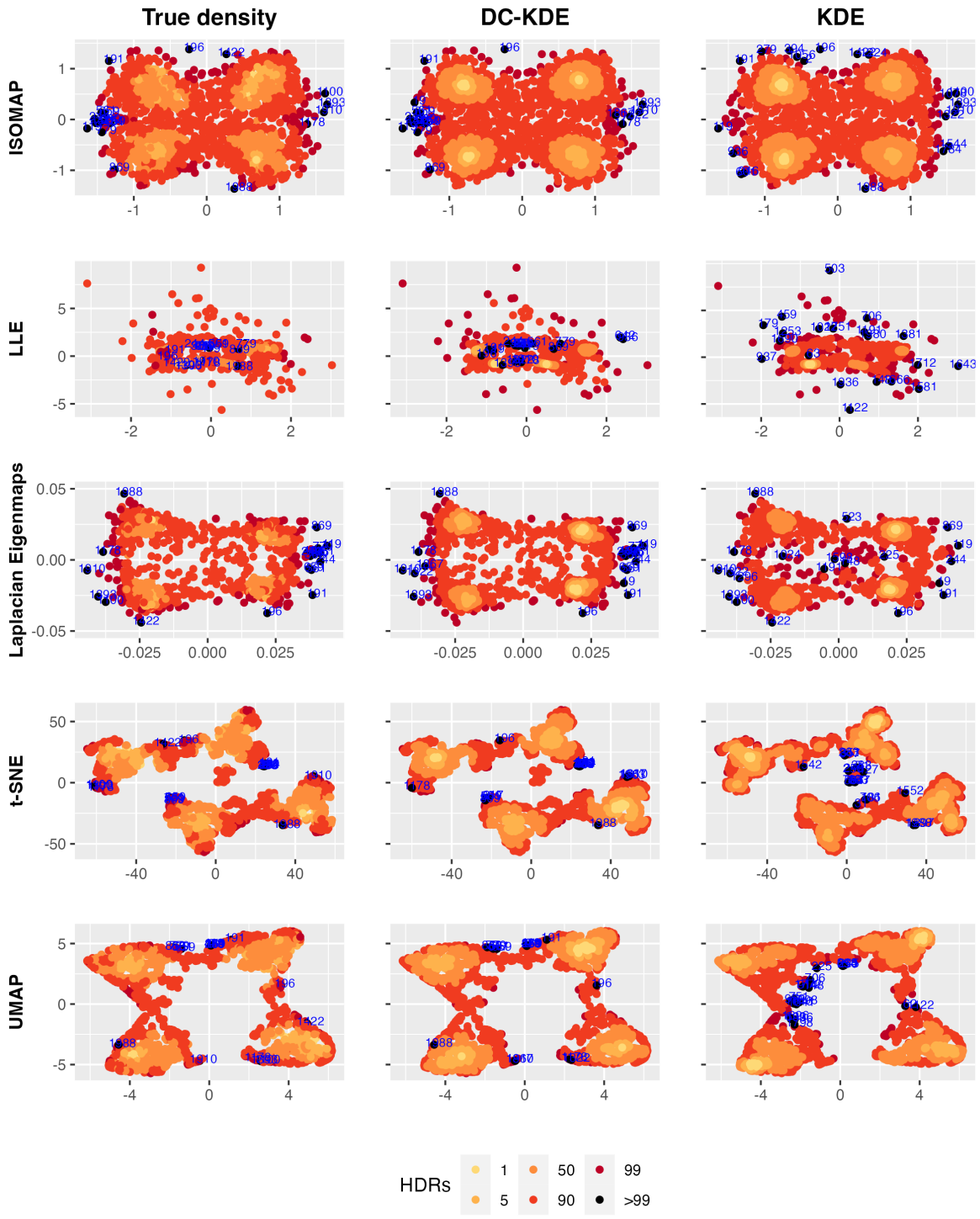

Figure 4: *Highest density region plots of five manifold learning embeddings of the twin peaks data in each row. The top 20 outliers, hightlighted in black and indexed in blue text, are found by the true manifold density (left panel), DC-KDE (middle panel) and KDE (right panel). DC-KDE finds more true outliers than KDE in all five rows.*

Table 1: *Correlation between true density ranking and estimated density ranking for different manifold learning embeddings of the twin peak data. Distortion corrected KDE outperfoms for all dimension reduction algorithms and gives the higher rank correlation to the output of t-SNE and UMAP.*

	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

correctly identify the anomalies. For t-SNE, the ground truth and distortion corrected KDE identify four regions of high density, while a KDE estimate without distortion correction seems to identify a larger number of modes. This concurs with the common observation that t-SNE tends to output clusters even where such clusters may not be present in the underlying data.

We can gain further insight by comparing the correlation between ranks of true densities and estimated densities from KDE with and without density correction by Table 1. Distortion correction improves the rank correlation for all dimension reduction algorithms. In particular, while applying KDE to the output of t-SNE and UMAP leads to a moderate correlation below 0.5, applying distortion correction improves these rank correlations to values close to 0.8.

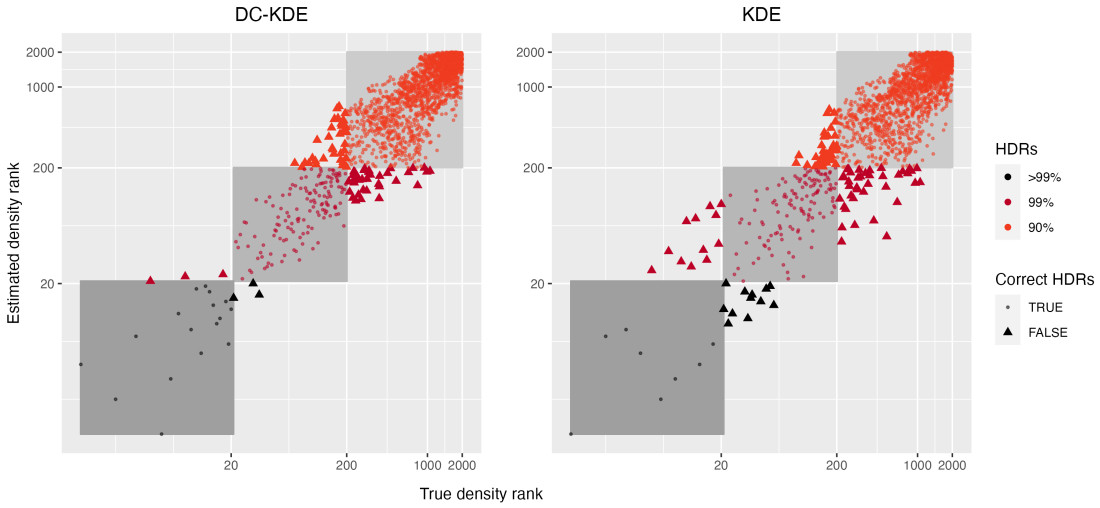

Figure 5: *Scatterplot of log scale ranks of true density and estimated density ranks for DC-KDE (in the left panel) and KDE (in the right panel) based on ISOMAP embedding. The colors indicate different level of highest density regions and the shapes indicate whether the density estimators correctly classify the true anomalies. The shading contains all anomalies that are both truely and correctly identified. KDE without distortion correction gives more misclassified anomalies.*

In Figure 5, we plot ranks of the estimated density against the true density for the ISOMAP embedding with left panels showing results for distortion correction and the right panel showing results without distortion correction. Data are presented on a log scale to highlight anomalies. The bottom left shaded region contains all points that are truly anomalies and are identified as such (true positives), where an anomaly is defined as a point not falling within a 99% highest density

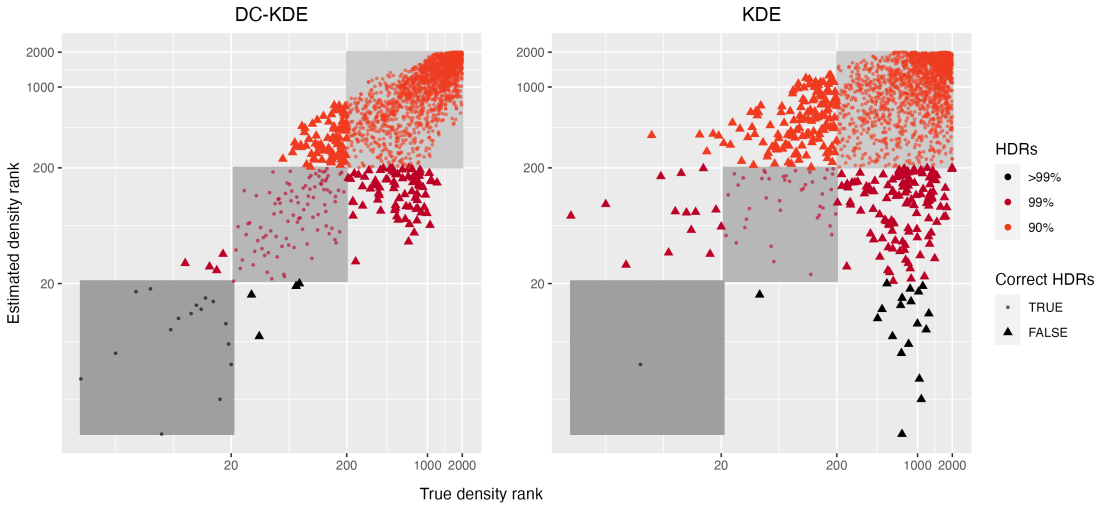

Figure 6: *Scatterplot of log scale ranks of true density and estimated density ranks for DC-KDE (in the left panel) and KDE (in the right panel) based on t-SNE embedding. KDE without distortion correction gives many more misclassified anomalies.*

region. The middle shaded region contain anomalies that are true positives in the sense of not lying in a 90% HDR. Points lying outside squares (shown as triangles) are incorrectly classified. For example, the three red triangles in the middle left of the left panel are truly anomalies since they lie outside the 99% HDR, but are not classified as such (although they are true positives if a 90% HDR is used). Overall, the right panel contains many more misclassified anomalies, which shows that failing to apply distortion correction can have a severe impact on anomaly detection. Figure 6 shows the same plot but for t-SNE. The quality of t-SNE is worse than ISOMAP in this example therefore many more anomalies are misclassified. However, the difference between KDE with and without distortion correction is stark. These results highlight the importance of applying distortion correction especially when the quality of dimension reduction may not be high.

Finally, Figure 7 demonstrates the robustness of distortion correction methods to the use of dimension reduction algorithm. Each row of panels compares ranks of the estimated densities based on a dimension reduction algorithm to the estimated density based on ISOMAP. The left column shows results when distortion correction is applied, the right column when it is not applied. It can be seen that the rank correlation between estimates based on different dimension reduction algorithms is much higher when distortion correction is applied. This is critical since conclusions will be more robust to the choice of dimension reduction algorithm.

3.2 Semi-hypersphere example embedded in 100-D space

As a high-dimensional experiment, we generate the underlying data from a 5-dimensional semi-hypersphere, embedded within 100-dimensional ambient space. To start with, we simulate vectors $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N)'$ for $N = 10,000$ points from a 4-dimensional Gaussian mixture model with two

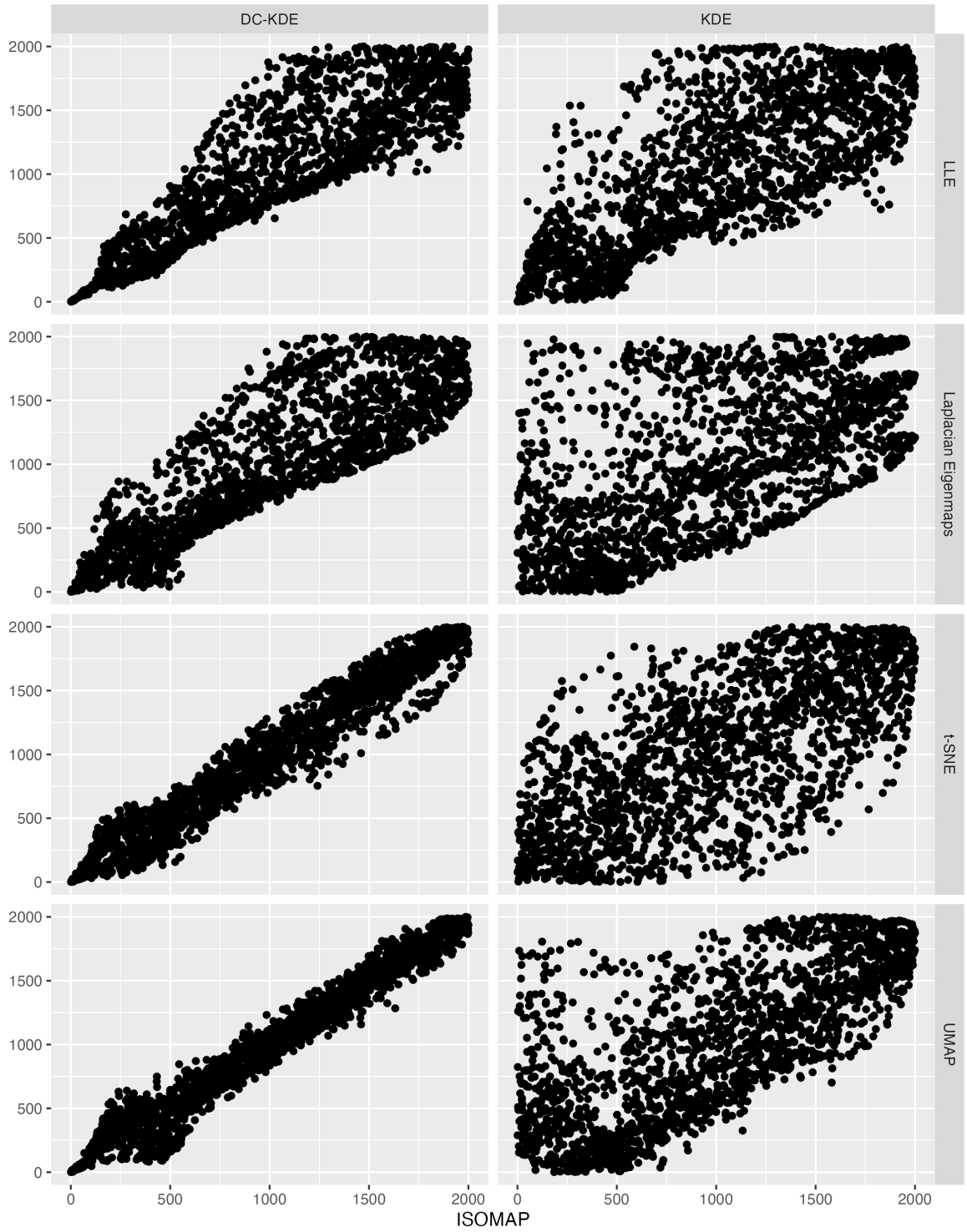

Figure 7: *Comparison of ranks of the estimated densities based on ISOMAP and four other dimension reduction algorithms in each row. Distortion corrected KDE (on the left panel) and KDE (on the right panel) are compared and DC-KDE shows the robustness to the use of different dimension reduction methods.*

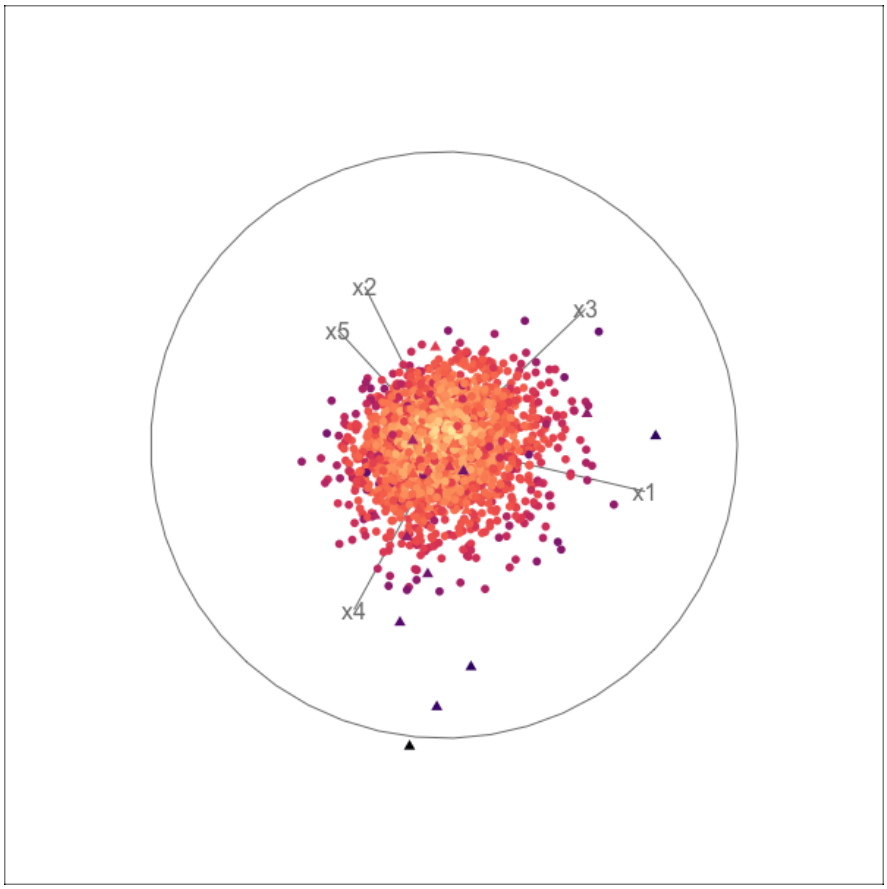

Figure 8: *Scatterplot display of the animation of a 5-D tour path with shapes indexing two Gaussian mixture components and the colors showing the distance to the kernel cores. Distant points in darker colors could be seen as anomalies.*

mixture components, $\mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, with the same means $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = (0,0,0,0)'$ and different variance-covariance matrices $\boldsymbol{\Sigma}_1 = diag(1,1,1,1)$ and $\boldsymbol{\Sigma}_2 = diag(2,2,2,2)$. The mixture proportions are set as $\pi_1 = 0.99$ and $\pi_2 = 0.01$. With this design, the observations from the second component tend to be outlying anomalies. The data are mapped to a hemisphere via the equation $v_1^2 + v_2^2 + v_3^2 + v_4^2 + v_5^2 = r^2$ where $v_5 > 0$ and r is set as 8. Figure 8 shows scatterplot which is a single frame from a 5-D tour path[1] animation using the R package *tourr* (Wickham et al. 2011). The round and triangular point shapes indicate the two mixture components $\mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2)$, and the colors indicate the distance between each point and the centre of the distribution. It can be seen that the most distant points are in darker colors and triangular shapes, meaning that the most anomalous observations are generated from the second mixture component.

To embed the 5-D hyper-semisphere into 100-D space, we append 95 zero columns to $\boldsymbol{v}_i$ so that $\boldsymbol{v}_i = (v_1, \ldots, v_5, 0, \ldots, 0), i = 1, \ldots, N$. Next, we rotate the 100-dimensional vectors $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N)'$ by multiplying by a randomly generated rotation matrix. To generate the rotation matrix we first simulate elements from a uniform $(0,1)$ distribution, stack them into a 100×100 matrix $\boldsymbol{A}$ and then

[1] See the animation of the 5-D grand tour at https://github.com/ffancheng/kderm/blob/master/paper/figures/tourr_5d_animation.gif.

take the R matrix from the QR decomposition of $\boldsymbol{A}$. Rotating the vectors results in input vectors that are no longer sparse. Nonetheless, the intrinsic dimension of this is still $d = 4$.

Following similar steps to Section 3.1, we estimate the densities of the 4-D manifold and compare them with the ground truth density. The bandwidth parameter for the DC-KDE estimator is set as $r = 1$ for this example. In Table 2, the rank correlations between true densities and estimated densites from DC-KDE and KDE are presented. For ISOMAP and LLE, KDE without distortion correction has a slightly higher rank correlation with the true density than for KDE with distortion correction, but these differences are negligible and both density estimators have a very high correlation of more than 0.96. However, for Laplacian Eigenmaps and UMAP there more distortion is induced throughdimension reduction, the rank correlation between estimated and true densities is close to 0 when distortion correction is not applied. In the same settings, the corresponding rank correlations are relatively high at 0.87 and 0.78 for Laplacian Eigenmaps and UMAP respectively,when distortion correction is applied.

Figure Figure 9 is constructed in a similar fashion to Figure 5 to show the effect of distortion correction on the detection of anomalies. As in Figure 8, the point shapes indicate which of the two mixture Gaussian mixture components an observation was generated from and the colors indicate the distance from the centre of the distribution with outliers shown in darker colors. The ranks of the estimated densities are shown against ranks of true densities (on the log scale) with panels on the left showing distortion corrected KDE and panels on the right showing KDE without distortion correction. The panels from top to bottom show four dimension reduction algorithms: 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 is only applicable to an embedding dimension less than or equal to three.

Comparing the left and right panel for each row, we notice that KDE with distortion correction, has fewer misclassified observations, and therefore outperforms KDE. For UMAP, almost all ground truth anomalies (points outside a 99% HDR) are not correctly detected using KDE, while almost all anomalies are correctly detected after correcting for distortion. For DC-KDE in the left panel there are several triangles in a horizontal line, this arises,due to many points having an estimated density of close to zero (values less than 10^{-7}) which leads to tied ranks. These low-density points are all detected as anomalies for all dimension reduction algorithms, which again shows the robustness of DC-KDE.

Table 3 further illustrates the percentage of correctly classified anomalies in the $> 99\%$ and 99% highest density regions. Compared with KDE, the percentages are always higher when distortion correction methods are used. The percentages with distortion correction are very close for ISOMAP,

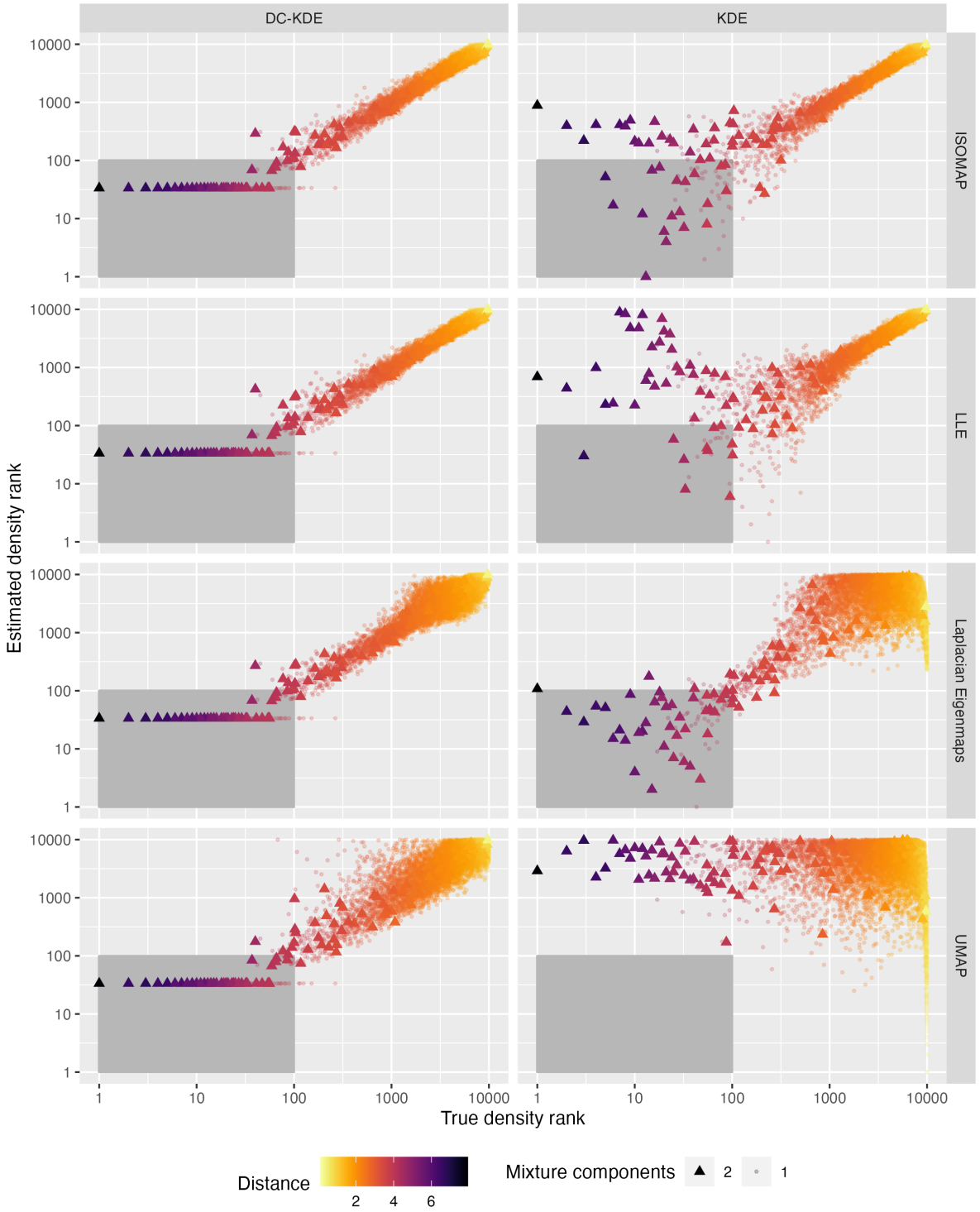

Figure 9: *Rank comparison between the true density and estimated density from both DC-KDE and KDE. Four manifold learning methods are used rowwise. The point shapes indicates whether they are the true outleirs, and the grey shading highlights the top 1% rank region. The colors show the distance to the 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% HDR	**0.830**	0.490	**0.830**	0.240	**0.840**	0.840	**0.770**	0.000
99% HDR	**0.818**	0.685	**0.805**	0.478	**0.815**	0.595	**0.632**	0.032

LLE and Laplacian Eigenmaps with a value of around 83% while slightly lower for UMAP at around 77%. This could be due to the severe distortion usually induced by the UMAP algorithm.

4 Application

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[2] records the half-hourly electricity consumption of individual residential and commercial properties, but 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 is worthwhile to explore the distribution of electricity demand rather than the raw consumption data, so as to study the usage patterns of different households or different periods or the week (Hyndman, Liu & Pinson 2018). This can be considered as a case of dimension reduction on a statistical manifold, that is a manifold with elements that are probability distributions. Cheng, Panagiotelis & Hyndman (2021) propose estimators of the Total Variation Metric and Hellinger distance between distributions that can be used in a computationally practical manner for dimension reduction on statistical manifolds. Again, we use ISOMAP, LLE, Laplacian Eigenmaps, t-SNE and UMAP for dimension reduction to obtain a 2 dimensional embedding for kernel density estimation and anomaly detection. To this end, we compare the density estimates from KDE with and without distortion correction and show how robust the anomalies are to different dimension reduction algorithms, only when distortion correction is used. We use the highest density regions plot to visualize the density estimates. However, for this real data set with unknown structure, the ground truth densities are unknown and it is not possible to tell which anomalies are the true ones.

Although full details for data processing and dimension reduction are provided in Cheng, Panagiotelis & Hyndman (2021), we briefly describe the process here. For each household, a discrete approximation to the distribution of electricity demand at each one of the 336 half-hourly periods

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

of the week is found. For any pair of households the total variation metric can found between the distributions corresponding to any half hour of the week, and summing over these gives a distance measure between the pair of households, subsequently used for dimension reduction algorithms and the Learn matric algorithm. The bandwidth parameter r is set as 1 in Equation (2).

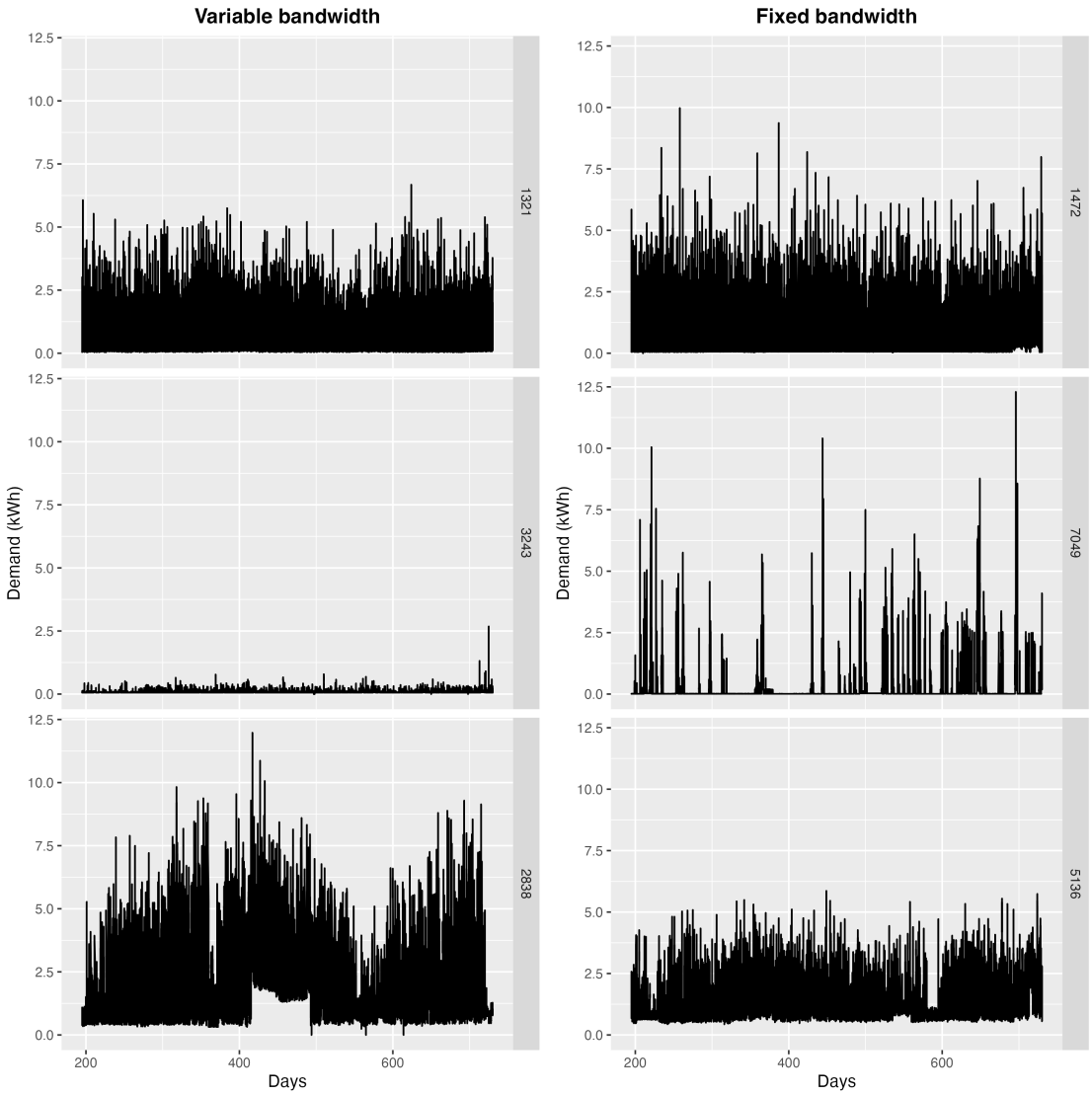

Figure 10: *Electricity usage plots of all 535 days for the most typical household and two anomalies in rows and KDE with distortion correction (left panel) and KDE (right panel) in columns.*

Figure 10 shows the electricity usage data of three households for density estimation both with and without distortion correction 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 similar except that there are a few spikes for household 1472 when using KDE. As for the anomalies, distortion corrected KDE tends to capture the unusual electricity demand when the usage is very low or high in volume. 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, KDE without distortion correction 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.

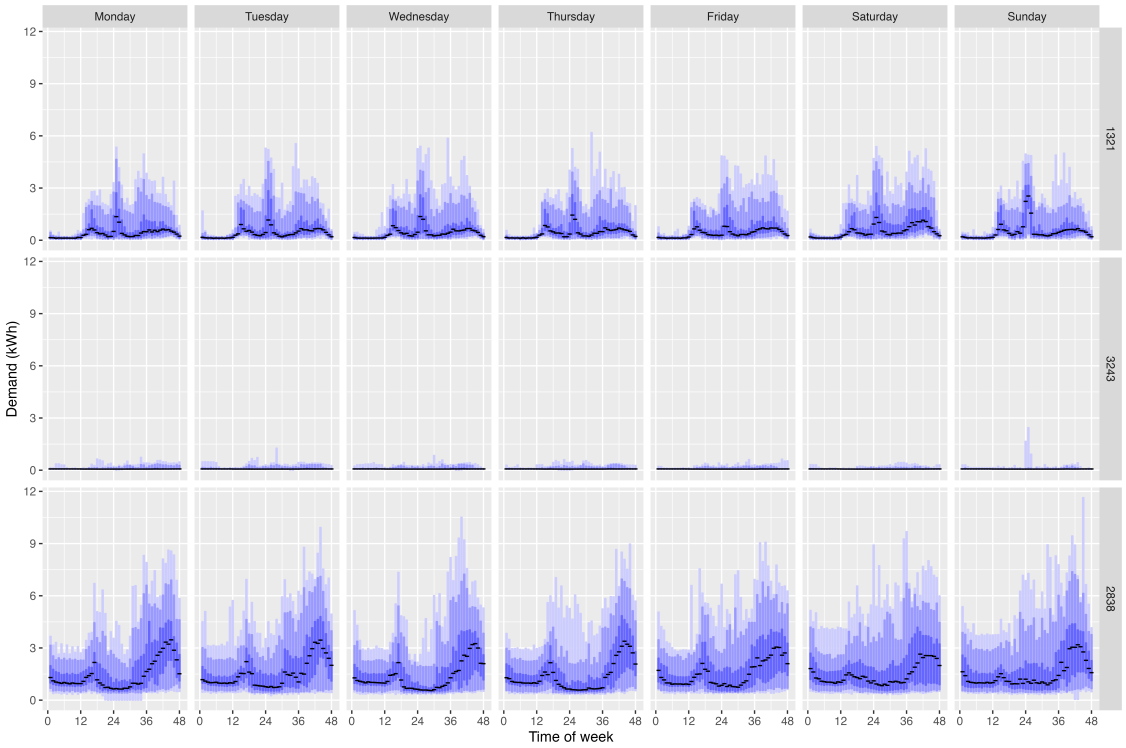

Figure 11: *Quantile region plots of electricity demand against the time of week for one typical household 1321 and two anomalies, 3243 and 2838. The quantile regions displayed in the plot are 99.0%, 95.0%, 75%, and 50%. Household 1321 has a repeated period-of-the-week electricity usage pattern slightly different for weekdays and weekends.*

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 11 and Figure 12. 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. This repeated pattern in a week window is also obvious for the typical household ID 1321 from DC-KDE. 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 kernel density estimation methods.

5 Conclusions

In this paper, we propose a novel distortion correction method to estimate the density of an embedding from manifold learning algorithms and further identify outliers based on the densities.

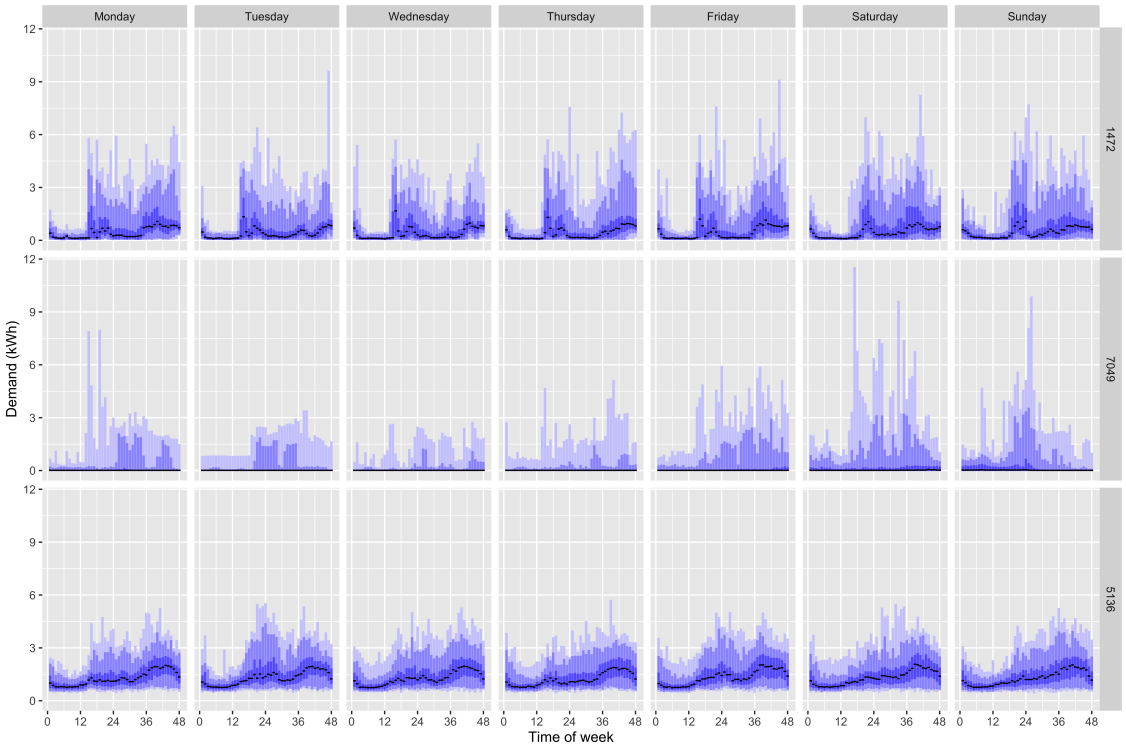

Figure 12: *Quantile region plots of electricity demand against the time of week for one typical household 1472 and two anomalies, 7049 and 5136. The day-of-the-week patterns is much more different for Household 7049 compared with 5136, the latter of which does not have the obvious meal time related usage patterens.*

Compared with KDE, our distortion corrected KDE makes use of geometric information for each data point to correct the distortion induced by the embedding. The Riemannian metric is estimated with the Learn Metric algorithm to approximate the geodesic distance and volume density function locally at each point. We compare our proposed method with KDE by two simulation settings, a 2-D manifold embedded as a 3-D twin peaks shape and a 4-D manifold mapped in a 100-D ambient space, and show that DC-KDE could generate more accurate kernel density estimation is more robust to the choice of dimension redution algorithm.

As an empirical example, we explore the distributions of 3, 639 households and 336 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 embedding, followed by density estimation with KDE and DC-KDE. Without the ground truth density, we compare both density estimates by looking at the distributions of the most typical households with the highest densities and the anomalous households with the top two 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 bandwidth parameters for kernel density estimation, which has been explored in many KDE-related literatures.

The second one is related to the quality of the embedding from dimension reduction methods. Although our distortion correction method is fairly robust to different choices of manifold learning methods, in certain cases, when the data structure is too complex and the distortion is too severe to correct, the quantitive relationship between embedding quality and density estimation accuracy is not immediate clear. The embedding quality could be measured using one of the metrics discussed in the online supplementary material of Cheng, Panagiotelis & Hyndman (2021), but when the ground truth densities are unknown, which is usually the case with real-world data set, it is hard to tell wherther the distortion are corrected in the right way. The density estimates on the edges of the whole data structure could alos be explored because most outer area points tend to be detected as outliers. However, the outperformance of DC-KDE than KDE 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,

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

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_pM, 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 $B(0_p,r) \subset T_pM$ 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 $d\phi_p : T_pM \to T_{\phi(p)}E$, which pushes the tangent vector $u \in T_pM$ 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 $u \in T_pM$ 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 5, Figure 6, and Figure 7. 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.

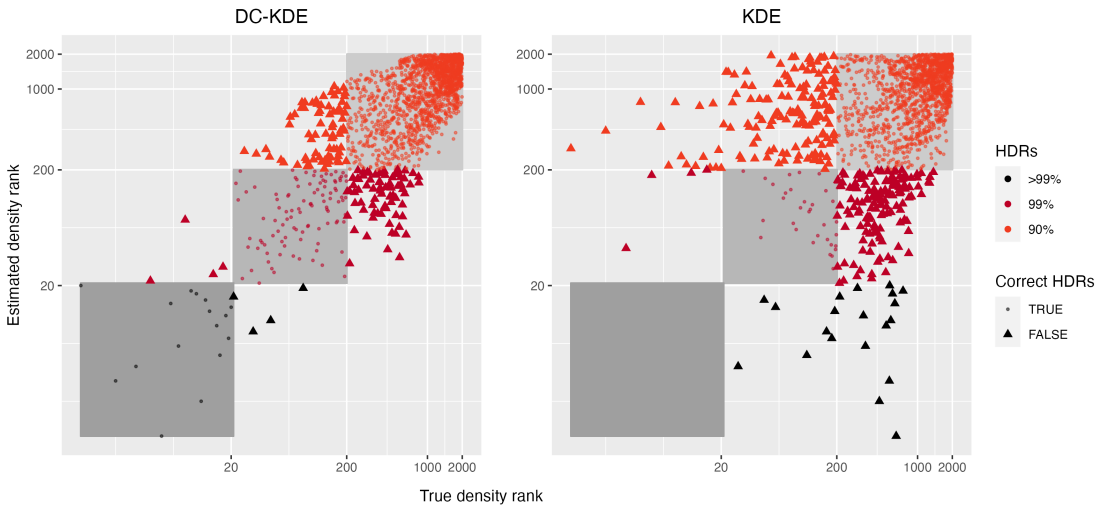

Figure 13: *Scatterplot of true density and estimated density ranks of LLE embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.*

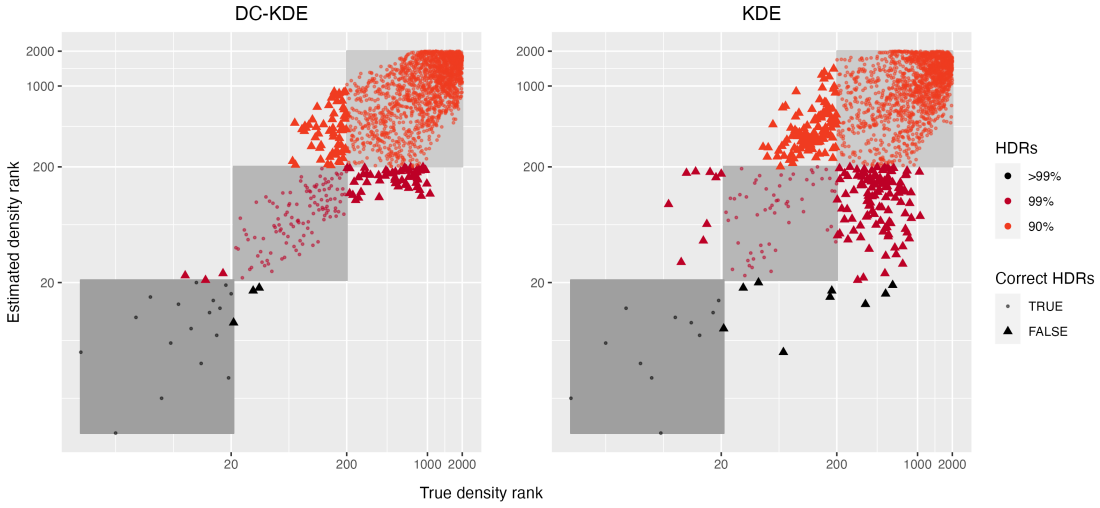

Figure 14: *Scatterplot of true density and estimated density ranks of Laplaxian Eigenmaps embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared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

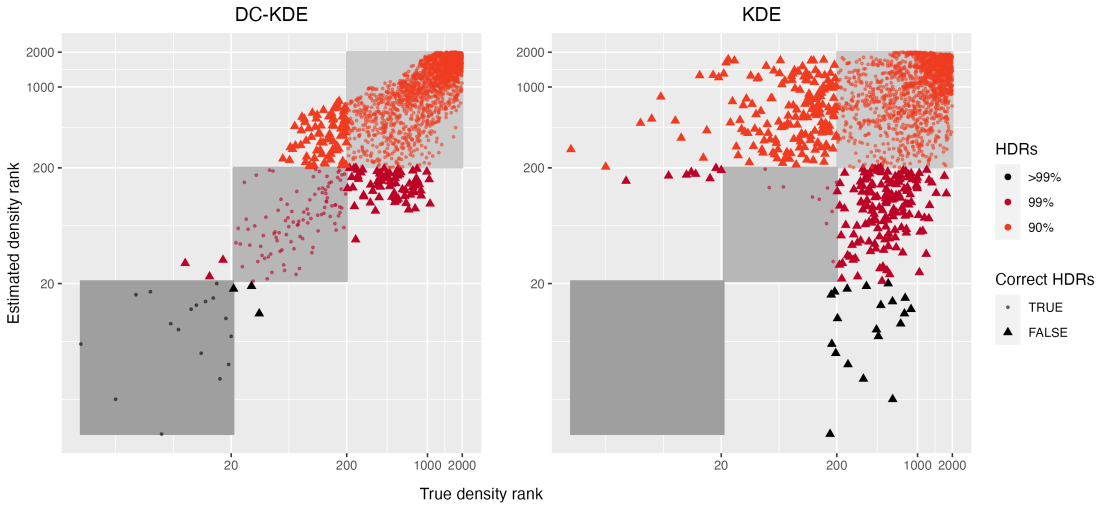

Figure 15: *Scatterplot of true density and estimated density ranks of UMAP embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.*

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, A Panagiotelis & RJ Hyndman (2021). Computationally Efficient Learning of Statistical Manifolds. *arXiv e-prints*, arXiv:2103.11773. https://ui.adsabs.harvard.edu/abs/2021arXiv210311773C.

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].

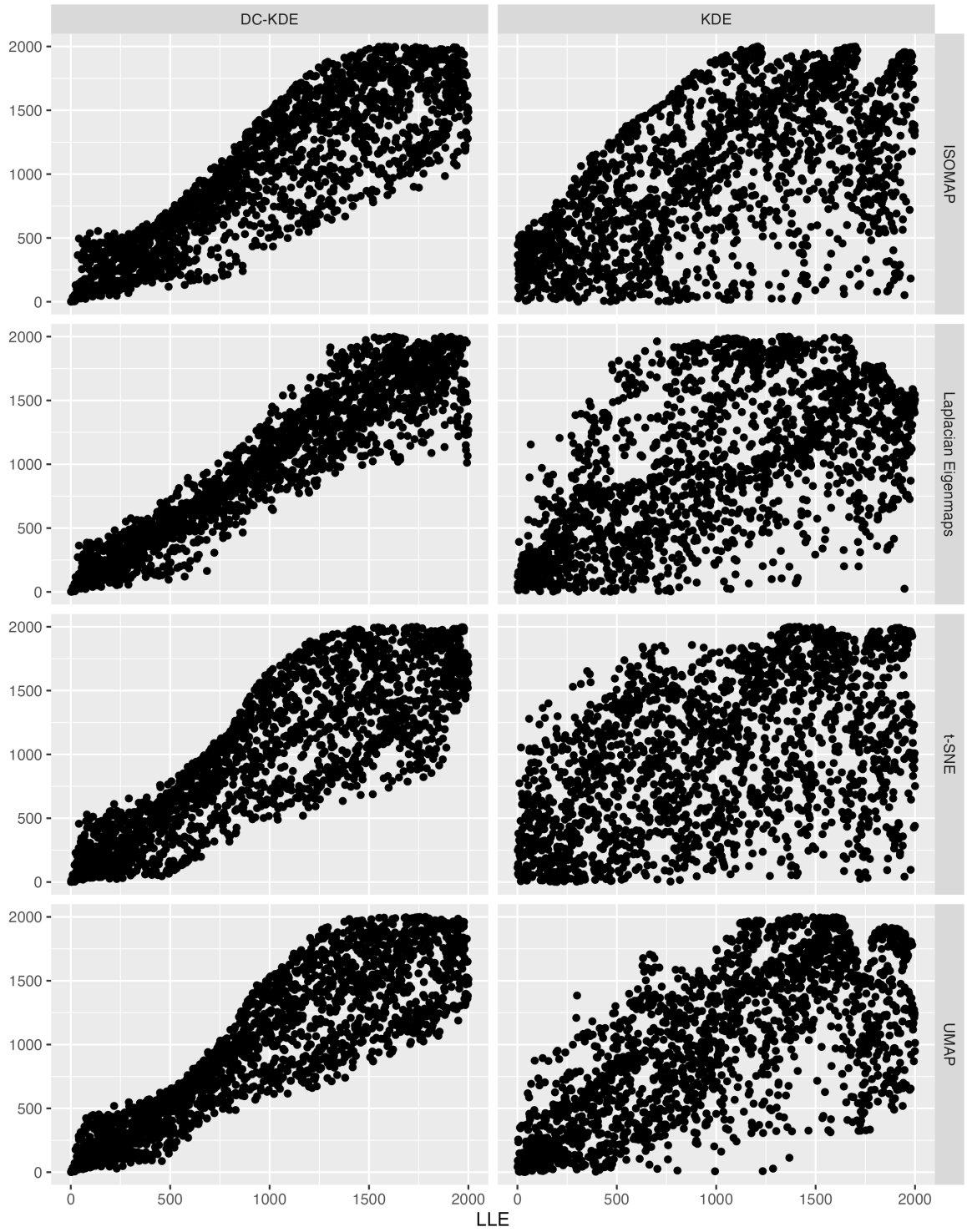

Figure 16: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless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

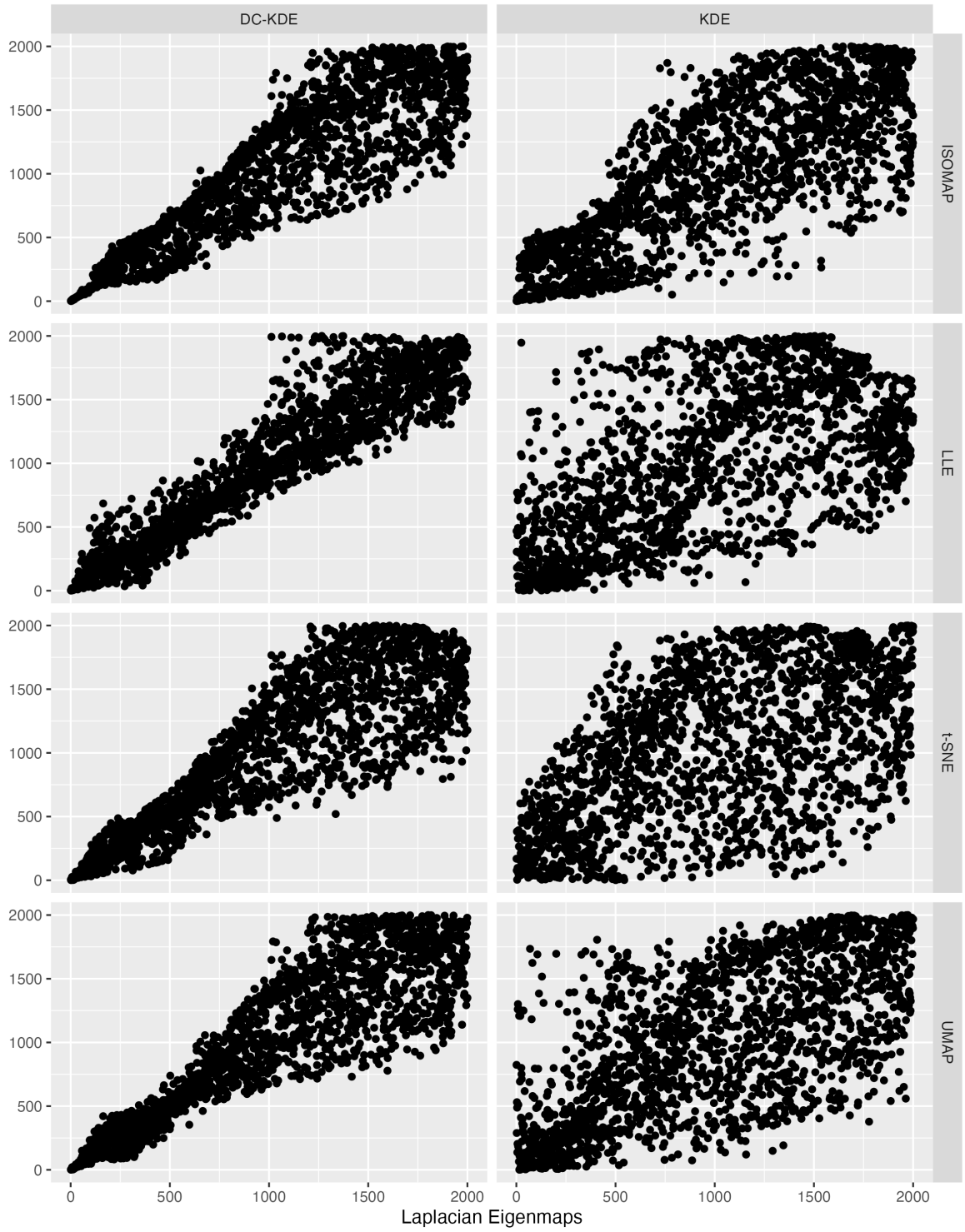

Figure 17: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless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

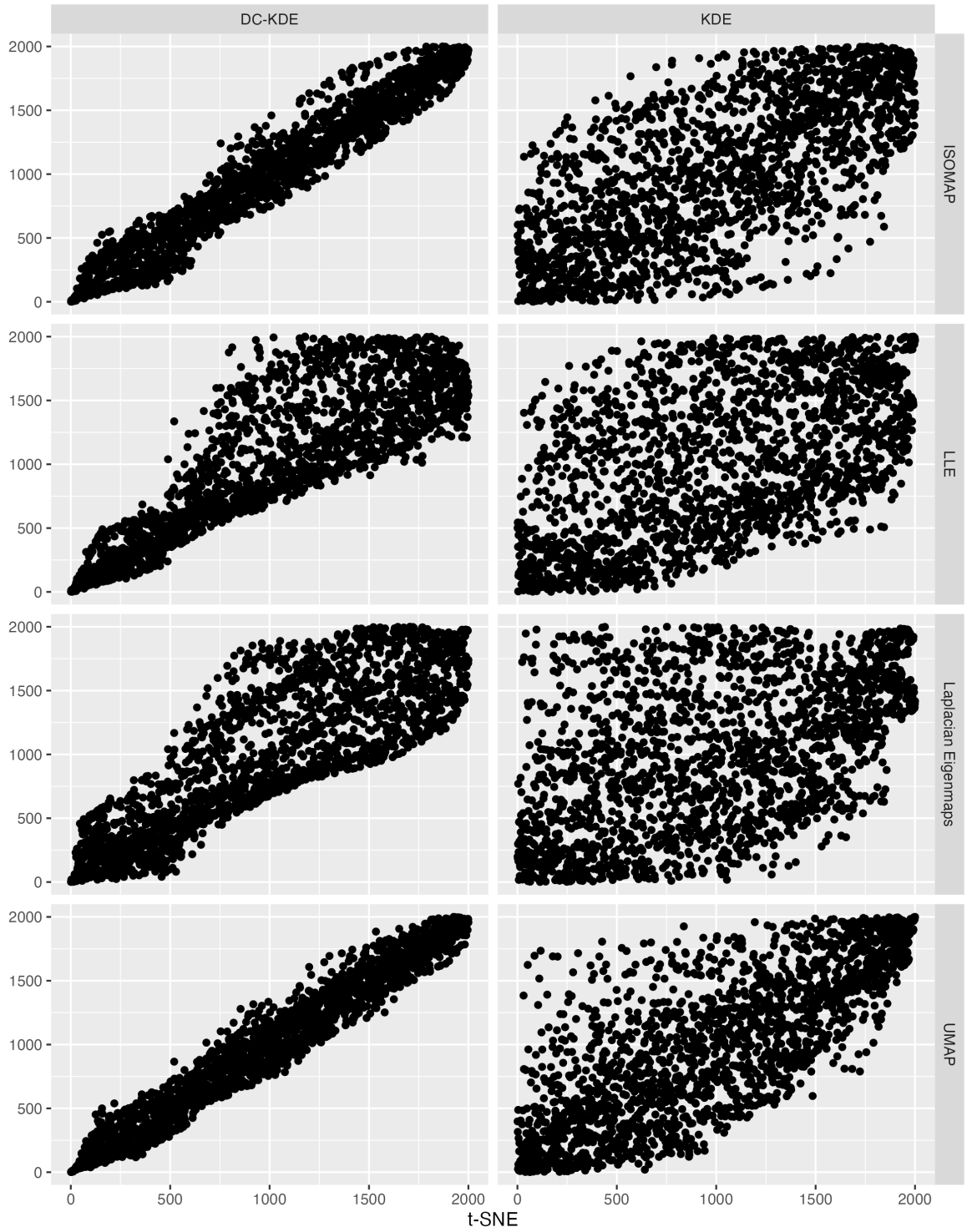

Figure 18: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless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

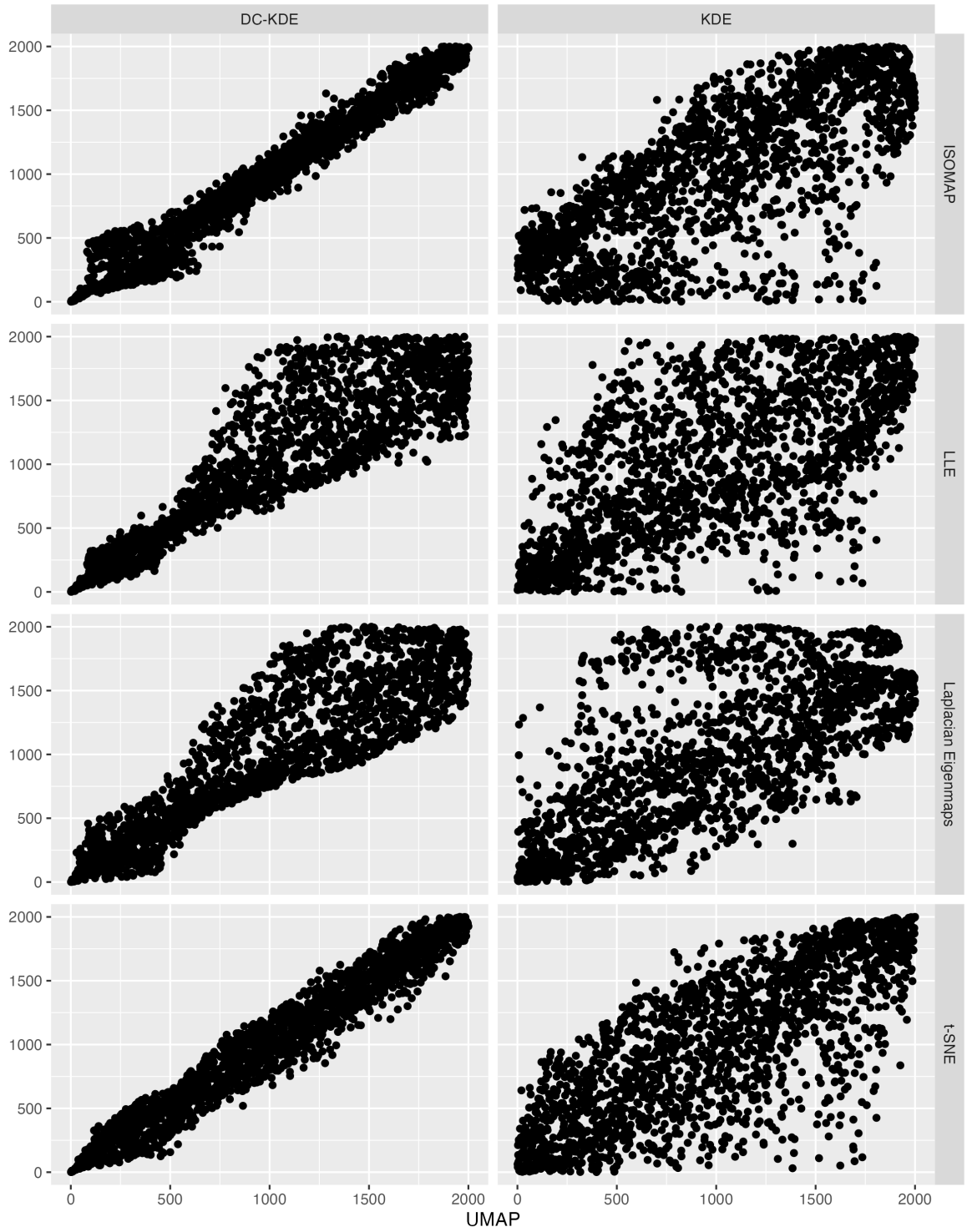

Figure 19: *Comparison of outliers found by one manifold learning method compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represents the four gaussian kernels in the 2-D meta data. Outliers found by DC-KDE are more consistent regardless of the manifold learning embedding.*

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.

Wickham, H, D Cook, H Hofmann & A Buja (2011). tourr: An R Package for Exploring Multivariate Data with Projections. en. *Journal of statistical software* **40**, 1–18. https://www.jstatsoft.org/article/view/v040i02.

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.