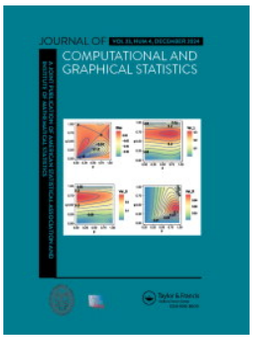

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ucgs20

Distortion corrected kernel density estimator on Riemannian manifolds

Fan Cheng, Rob J Hyndman & Anastasios Panagiotelis

To cite this article: Fan Cheng, Rob J Hyndman & Anastasios Panagiotelis (24 Oct 2024): Distortion corrected kernel density estimator on Riemannian manifolds, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2024.2415543

To link to this article: https://doi.org/10.1080/10618600.2024.2415543

Distortion corrected kernel density estimator on Riemannian manifolds

Fan Cheng[a], Rob J Hyndman[a,*], Anastasios Panagiotelis[b,†]

[a]Monash University

[b]The University of Sydney

*Rob J Hyndman's research is partially funded by the Australian Government through the Australian Research Council Industrial Transformation Training Centre in Optimisation Technologies, Integrated Methodologies, and Applications (OPTIMA), Project ID IC200100009.

†Corresponding Author: anastasios.panagiotelis@sydney.edu.au

Abstract

Manifold learning obtains a low-dimensional representation of an underlying Riemannian manifold supporting high-dimensional data. 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, with a bandwidth that depends on the estimated Riemannian metric at each data point. Exploiting the geometric information of the manifold leads to more accurate density estimation, which subsequently could be used for anomaly detection. To compare our proposed estimator with a fixed-bandwidth kernel density estimator, we run two simulations including one with data lying in a 100 dimensional ambient space. We demonstrate that the proposed DC-KDE improves the density estimates as long as the manifold learning embedding is of sufficient quality, and has higher rank correlations with the true manifold density. Further simulation results are provided via a supplementary R shiny app. The proposed method is applied to density estimation in statistical manifolds of electricity usage with the Irish smart meter data.

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

1 Introduction

Multivariate kernel density estimation (KDE see Parzen (1962); Scott (2015) and references therein) is a fundamental tool for exploratory data analysis and unsupervised learning. Applications of KDE include finding hot spots of traffic networks 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) and prime prediction via Twitter messages (Gerber, 2014). For *m*-dimensional data where *m* is even of moderate size, KDE is challenging due to the well-known curse of dimensionality. However, the manifold hypothesis suggests that many high-dimensional datasets belong to a *d*-dimensional manifold *M* where $d \ll m$. The problem of KDE on manifolds has been well studied in Hendriks (1990), Izenman (1991), Pelletier (2005), Henry, Munoz & Rodriguez (2013), and Berenfeld, Rosa & Rousseau (2022a). However, in these settings, it is assumed that the underlying manifold is known, which is rarely true in practice. There is only a small literature dealing with the case where the manifold is unknown. Recently, Berenfeld & Hoffmann (2021a), Berenfeld, Rosa & Rousseau (2022b) and Divol (2022) demonstrated the minimax optimality of certain types of KDE on manifolds. In contrast, our objective in this paper is to propose a new KDE for data that lie on some unknown manifold found using embedding techniques.

When dealing with data that lie on a manifold, it is common to use dimension-reduction techniques. These algorithms, which include ISOMAP, LLE, t-SNE, and UMAP among others, can be thought of as a mapping $\psi : M \to \mathbb{R}^d$ that take points on the manifold $\boldsymbol{p}$ to *d*-dimensional vectors $\boldsymbol{y}$, where the dimension *d* is either assumed to be known a priori or needs to be estimated separately. Details of these algorithms are summarized in Cayton (2005), Lee & Verleysen (2007), Izenman (2012), and Cheng, Panagiotelis & Hyndman (2021). One could easily apply KDE to the output vectors $\boldsymbol{y}$. However, this would be a misleading representation of the density of points as they lie on the manifold. To see how, consider that the density on the manifold must be defined with respect to an appropriate volume form, that is

$$Pr(\boldsymbol{p} \in \mathcal{A}) = \int_{\mathcal{A}} f(\boldsymbol{p}) dvol_M,$$

where *f* is the probability density function on the manifold, $\boldsymbol{p}$ is a point on the manifold, $\mathcal{A}$ is some region on the manifold and $dvol_M$ is a volume form on the manifold *M*. By a change of variables, this would imply the following density for $\boldsymbol{y}$,

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

where $\psi(\mathcal{A})$ is the image of $\mathcal{A}$ under ψ and $\boldsymbol{H}(\boldsymbol{y})$ is the Riemannian metric expressed in local coordinates given by the mapping ψ (Do Carmo, 1992). It can be seen that the density in terms of $\boldsymbol{y}$, differs from the density on the manifold due to the term $|\boldsymbol{H}(\boldsymbol{y})|^{1/2}$ which can be considered roughly analogous to the Jacobian determinant. Since ψ will typically be a

non-linear function, the value of $|\boldsymbol{H}(\boldsymbol{y})|^{1/2}$ will change across the manifold, in a way that measures the distortion induced by the dimension reduction algorithm. Notably, different choices of dimension reduction algorithm, imply different ψ, meaning that if KDE is applied to the output $\boldsymbol{y}$ coming from two different dimension reduction algorithms, this could lead to two very different sets of conclusions about the density of the same point.

In this paper, we propose a new KDE. Our estimator is also based on the output of dimension reduction algorithms but accounts for the distortion induced by the algorithm. For this reason, we refer to it as the distortion-corrected KDE (DC-KDE). Critical to our estimator, is obtaining an estimate of $|\boldsymbol{H}(\boldsymbol{y})|^{1/2}$ for this purpose we use the Learn Metric algorithm of Perrault-Joncas & Meila (2013) which augments any dimension reduction algorithm with an estimate of the Riemannian metric at each data point. One particular downstream task that we are interested in is anomaly detection, where anomalies are identified as points of low density. In a number of simulation studies, we demonstrate two benefits that arise when using our new estimator. The first is that using distortion correction leads to more accurate density estimates and better detection of anomalies compared to the case where distortion correction is not used. The second is that when the estimates are found using different dimension reduction algorithms (e.g. ISOMAP v UMAP), the results regarding the density estimates and the detection of anomalies are robust only when distortion correction is applied.

In terms of data visualization, our approach allows a two or three-dimensional scatterplot of $\boldsymbol{y}$ to be augmented by coloring points according to the magnitude of the density estimate, where the dimension of $\boldsymbol{y}$, d, is set as 2 or 3 in the dimension reduction algorithms. In this way, the user is able to tell whether a region that has a cluster of points in the output space $\boldsymbol{y}$ is truly a high-density region or if instead, this is an artifact of the dimension reduction algorithm 'squeezing' points together. This is particularly pertinent in the case of t-SNE which can form clusters of points even when no such clusters exist in the underlying data. Furthermore, our estimator can be used for manifolds with elements that are a variety of objects, for instance, rotational frames or images. In our application we consider a probabilistic manifold whose elements are densities, meaning that we estimate a density of densities. Finally, our proposed estimator also takes the form of a KDE with variable bandwidth meaning that it also contributes to extensive literature on this topic (see Section 6.6 of Scott, 2015, for a summary).

The rest of the paper is organized as follows. In section 2, we present our distortion-corrected kernel density estimator for Riemannian manifolds. Our estimator combines insights from two existing papers, the first is Pelletier (2005) who propose a KDE for data on manifolds that are known, the second is Perrault-Joncas & Meila (2013) who propose the Learn Metric algorithm which we use to estimate the distortion induced by a dimension reduction algorithm. section 3 is composed of two simulations; the first deals with 2-dimensional data from a Gaussian mixture model mapped onto a 3-dimensional 'twin peaks' manifold and the second with a 5-dimensional semi-hypersphere data embedded into a 100-dimensional space. section 4 contains an application to an Irish smart meter dataset that visualizes and identifies households with an anomalous distribution of energy usage. Conclusions and discussions are presented in section 5. Some additional background on important terms in Riemannian geometry required to understand our proposed method is provided in Appendix A.

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 subsection 2.1 how this generalizes to the estimator of Pelletier (2005), which can be applied when the data lie on some known manifold. In subsection 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 subsection 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 the following assumptions about the kernel function $K : \mathbb{R}_+ \to \mathbb{R}$,

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

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 $0 \le z \le 1$ 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.

2.1 Kernel Density estimation on manifolds

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

$$\hat{f}(\boldsymbol{p}) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{r^d \theta_{\boldsymbol{p}_i}(\boldsymbol{p})} K\left(\frac{d_g(\boldsymbol{p}, \boldsymbol{p}_i)}{r}\right), \quad (3)$$

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 a 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 therefore the volume density function must be included in the KDE to ensure that the estimated density integrates to one.

More formally, the volume density function can be explained as follows. Consider the exponential map around $\boldsymbol{p}$, given by $exp_{\boldsymbol{p}}(\boldsymbol{v})$, 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 traveling 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 then the square root of the determinant of the Riemannian metric expressed in this coordinate system. In proving that the estimator integrates to 1, the volume density function cancels with the term $|\boldsymbol{H}(\boldsymbol{y})|^{1/2}$ in Equation (1), where ψ is the logarithmic map around $\boldsymbol{p}_i$. For more on the volume density function, see Gallot, Hulin & Lafontaine (2004) and Le 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. Formally, the Riemannian metric is a symmetric and positive definite tensor field that defines an inner product on the tangent space $T_{\boldsymbol{p}}M$ for every point $\boldsymbol{p} \in M$. Using the coordinates from a dimension reduction algorithm $\boldsymbol{y}$, 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.

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 fixed-radius nearest neighbor of $\boldsymbol{p}_j$ or vice versa, and edge weights depending on the distance between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ on the manifold. The fixed-radius method considers all points within a specified radius of $\sqrt{\epsilon}$, allowing the graph to adapt to varying densities in the data and to capture local structure more effectively. As an alternative, K nearest neghbors could also be used. Second, the discrete Laplacian on this graph $\mathcal{L}_{\varepsilon,n}$ is estimated (Zhou & Belkin, 2011), where $\sqrt{\varepsilon}$ is the radius parameter in the previous step and a constant value $c = 0.25$ is used for the use of heat kernel in the weighted neighborhood graph. 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*. For computational efficiency, approximate nearest neighbor searching methods are implemented to construct the neighborhood graph (Cheng, Panagiotelis & Hyndman, 2021; Perrault-Joncas & Meila, 2013), which are then used in both steps 1 and 3. Two parameters, $c = 0.25$ and $\sqrt{\varepsilon} = 0.4$, are set as suggested in the *megaman* library. Note that in algorithm 1 c and $\sqrt{\varepsilon} = 0.4$ enter multiplicatively therefore only one of these parameters requires tuning. We considered different values of $\sqrt{\varepsilon}$, noting that results were stable for a range of values close to the default value of $\sqrt{\varepsilon} = 0.4$.

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 (See Chapter 3 of Lee & Verleysen (2007) for more about intrinsic dimension estimation). The *R* library *intRinsic* (Denti, 2021) implements this algorithm and is used in all examples involving real data where the intrinsic dimension is unknown.

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 $\boldsymbol{H}_i$ for all $i = 1, \ldots, N$

parameter : embedding dimension *d*, bandwidth parameter $\sqrt{\varepsilon}$, manifold learning algorithm EMBED

optimization parameter : manifold learning parameters in 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 $\mathcal{L}_{\varepsilon,N}$ by

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

where $c = 0.25$, $\tilde{D} = diag\tilde{W}\boldsymbol{1}$, $\tilde{W} = D^{-1}WD^{-1}$, and $D = diagW\boldsymbol{1}$;

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 $d \times d \times N$ tensor $\tilde{\boldsymbol{H}}$ by applying the graph Laplacian $\mathcal{L}_{\sqrt{\varepsilon},N}$ to the embedding coordinates matrix $\boldsymbol{Y}$ with elements in $\tilde{\boldsymbol{H}}$ found by

$$\tilde{\boldsymbol{H}}^{lk\cdot} = \frac{1}{2}\left[\mathcal{L}_{\varepsilon,N}\left(\boldsymbol{y}^l \odot \boldsymbol{y}^k\right) - \boldsymbol{y}^l \odot \left(\mathcal{L}_{\varepsilon,n}\boldsymbol{y}^k\right) - \boldsymbol{y}^k \odot \left(\mathcal{L}_{\varepsilon,n}\boldsymbol{y}^l\right)\right],$$

where $l, k = 1, \ldots, d$, $\tilde{\boldsymbol{H}}^{lki}$ is the element in row *l* and column *k* of the inverse Riemannian corresponding to observation *i*, and the $\odot$ calculation is the element-wise product between two vectors;

5: For each embedding data point $\boldsymbol{y}_i, i = 1, \ldots, N$, calculate its $d \times d$ Riemannian metric $\boldsymbol{H}_i$ as the rank *d* pseudo inverse of $\boldsymbol{H}^{-i}$ below

$$\boldsymbol{H}_i = \boldsymbol{U}\Lambda_d^{-1}\boldsymbol{U}',$$

where $[\boldsymbol{U}, \Lambda]$ is the eigendecomposition of the $d \times d$ matrix $\boldsymbol{H}^{-i}$, $\boldsymbol{U}$ is the matrix of column eigenvectors ordered by the eigenvalues in descending order, and Λ_d is a diagonal matrix containing the top *d* largest eigenvalues.

2.3 Distortion corrected KDE

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

$$\hat{f}(\boldsymbol{y}_j) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r^d}\left(\frac{|\det \boldsymbol{H}_j|}{|\det \boldsymbol{H}_i|}\right)^{1/2} K\left(\frac{\| \boldsymbol{H}_i^{-1/2}(\boldsymbol{y}_j - \boldsymbol{y}_i)\|}{r}\right). \quad (4)$$

The estimator has a similar structure to Equation (2.1) which can be seen as the counterpart to Equation (2.3) if the Riemannian is known exactly. There are however, some key differences between the two equations. 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 in 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 (2.1) 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 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. To account for distortion, this average should be weighted by the determinant of the Riemannian at each of the nearest neighbors. 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 the 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 shades. With this anomaly detection process, outliers based on the lowest densities could be detected more accurately regardless of the distortion in manifold learning.

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, subsection 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 subsection 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 $\Sigma_i = diag(0.016, 0.016), i = 1,2,3,4$. 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

$$\begin{aligned} x_1 &= v_1, \\ x_2 &= v_2, \\ x_3 &= \sin(\pi v_1)\tan(3v_2). \end{aligned} \qquad (5)$$

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.

It is important to note that the *true density* on the manifold is not simply a Gaussian mixture, since the mapping in Equation (3.1) 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 Γ_i for $i = 1,\ldots,n$ where Γ_i is the Riemannian metric 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 $f(\boldsymbol{p}_i) = f(\boldsymbol{v}_i)|\Gamma_i|^{1/2}$, where $f(\boldsymbol{v}_i)$ is the density of a four-component mixture of

normals. Knowledge of $\boldsymbol{v}$ and Γ_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 lighter colors. The anomalies are found around the edges of the plot, but there 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 (2.3) as $r = 0.5$. Colors show the different estimated densities at each point with anomalies shown in darker colors with blue indexing text, and higher density points shown in lighter colors. 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 towards the center. The distortion-corrected KDE can account for this, while KDE without distortion correction on the other hand does not 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 (Cai & Ma, 2022).

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.

In Figure 5, we plot ranks of the estimated density against the true density for the ISOMAP embedding with the 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 region. The middle shaded region contains 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 a 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, and 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 mixture components, $\mathcal{N}(\boldsymbol{\mu}_1,\Sigma_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2,\Sigma_2)$, with the same means $\boldsymbol{\mu}_1=\boldsymbol{\mu}_2=(0,0,0,0)'$ and different variance-covariance matrices $\Sigma_1=diag(1,1,1,1)$ and $\Sigma_2=diag(2,2,2,2)$. 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 a 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 $\mathcal{N}(\boldsymbol{\mu}_1,\Sigma_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{\sigma}_2)$, and the colors indicate the distance between each point and the center 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 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 subsection 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 densities 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, more distortion is induced through dimension 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 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 center 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 panels 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 proportions of correctly classified anomalies in the $>99\%$ and 99% highest density regions. Compared with KDE, the proportions are always higher when distortion correction methods are used. The proportions with distortion correction are very close for ISOMAP, 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 of 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 an 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 of the distribution of electricity demand at each one of the 336 half-hourly periods of the week is found. For any pair of households the total variation metric can be 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 Metric algorithm. The number of nearest neighbors in manifold learning K is set as 100 and the bandwidth parameter $r = 180$ to match the distances between the distributions of households.

Figure 10 presents the highest density regions of the 2-dimensional embedding for each dimension reduction method, ISOMAP, LLE, Laplacian Eigenmaps, t-SNE, and UMAP, with the left panels showing the densities estimated with the distortion corrected KDE and the right panels showing those estimate with KDE. There are six density regions colored in the plot, 1%, 5%, 50%, 90%, 99% and $>99\%$, and darker colors indicate lower density regions. The dark-colored points indexed with blue household IDs are the top 20 anomalies with the lowest densities. First of all, for the same electricity data, each dimension reduction method generates quite different embeddings, meaning that distortions exist in these embeddings. When comparing row-wise for the same embedding, it can be seen that ISOMAP and LLE give similar anomalies while the outliers are more clustered for Laplacian Eigenmaps, t-SNE, and UMAP with DC-KDE compared with the KDE. As the anomalous IDs are not easy to distinguish in the plots, we list them in Table 4 and highlight the IDs that are most identified as anomalies. The darker blue points are those households that are identified five times out of the ten cases shown in Figure 10, and lighter blue points are those identified four times. Again, KDE with distortion correction methods finds more anomalies in common regardless of the dimension reduction methods, compared with KDE where only LLE could find similar anomalies to other cases. Without distortion correction, the density estimates are different not only in scale, but also show inconsistency in the rank of density estimates.

Unlike for the simulations where the ground truth densities are known, in the applications, we have no knowledge of the truly anomalous households. To explore the robustness of anomaly detection both with and without distortion correction, we assume, in this case, that the anomalies given by UMAP are the 'true' ones and evaluate the performance of the other four dimension reduction methods, ISOMAP, LLE, Laplacian Eigenmaps and t-SNE. The results are summarized in Table 5 which can be interpreted similarly to Table 3. Table 5 presents the proportions of 'true' outliers (given by UMAP) where anomalies fall outside the $>99\%$ and 99% HDRs respectively that are also identified as anomalies by different dimension reduction methods. Both DC-KDE and KDE are shown in the table. Apart from the lower percentage of ISOMAP in $>99\%$ HDRs, distortion-corrected KDE gives higher proportions of true outliers in all other methods. The improvement brought by distortion correction is much more pronounced in Laplacian Eigenmaps and t-SNE which would otherwise detect almost completely different anomalies to UMAP. Figure 11 presents the visualization of density ranking in log scale. The grey shading area covers the density rank from 1 to 20, which are the anomalies listed in Table 4. Again, the plots show the robustness of density

estimated with distortion correction regardless of the dimension reduction methods. The robustness of anomaly detection to different methods when using distortion correction is also shown in Table 4. The household indices shown in darker shades of blue are identified as anomalies the most with dimension reduction algorithms. It is notable that without applying distortion correction, Laplacian Eigenmaps, t-SNE and UMAP do not identify any of the same anomalies as ISOMAP and LLE.

We now further examine households 3161, 4546, and 7049, which appear as anomalies for at least three dimension reduction algorithms as well as Households 2396, which is the most typical household with the highest density estimate. Their half-hourly electricity demands across the entire trial period are plotted in Figure 12. Compared with ID 2396, the other three households show different usage patterns. ID 3161 has much less electricity usage in summers, mostly below 1kWh, than in winters, above 3kWh. As this data set does not include the energy used for cooling or heating systems, it is possible that there are other large electronic appliances used in winter for this household. The other two anomalous IDs, 4546 and 7049, however, have nearly zero kWh electricity consumption for many days and some spikes for other days. It is likely that these two households are not occupied all the time during the trial and could be used as temporary accommodation (e.g. through a holiday rental service such as Airbnb).

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 12. The distribution of the 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 shown on some days of the week for household ID 3161 with small spikes at noon and midnight but not on Fridays and Saturdays. As for the bottom two households, the middle row in Figure 13 shows a distribution with zero median and a 75% quantile of 1kWh from around 9 pm to 6 am, while the distributions for ID 7049 do not have an obvious pattern across different periods of the week. These findings show that distortion correction methods work well in estimating the densities of the electricity usage and detecting anomalous households in the smart meter data despite the various embeddings from different dimension reduction 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. 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 reduction 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 assume that UMAP gives the true densities and compare both density estimation methods by looking at the distributions of the most typical households with the highest densities and three anomalous households that are identified most times. The distortion correction methods could identify the typical households with repeated time-of-the-week usage patterns, while the distributions of the outliers are anomalous in different ways.

There are several open questions to be explored. The first involves the selection of bandwidth parameters and kernel types for kernel density estimation, which has been widely explored in many KDE-related literature. Our method requires the choice of a single scale r that tunes the varying bandwidth matrix. In practice, this could be selected using a grid search according to some criteria that depends on the downstream task. For instance, for anomaly detection, this criterion may be a measure of the similarity of anomalies detected both in sample and out-of-sample found using cross validation. For densities on unknown manifolds, Berenfeld & Hoffmann (2021b) also provide some insights on the assumptions of the kernel function in (2). The second one is that the proposed density estimation method is only limited to the embedding points. Although for the downstream task of anomaly detection, only density estimated on these points is needed, it would be worth exploring the out-of-sample density estimation and estimation on the boundary of manifolds (Berry & Sauer, 2017) in the future. The last question 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 quantitative relationship between embedding quality and density estimation accuracy is not immediately 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 whether the distortion is corrected in the right way. The density estimates on the edges of the whole data structure could also be explored because most outer area points tend to be detected as outliers. However, the outperformance of DC-KDE over 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 Monash University.

Notes

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

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

References

Berenfeld, C & M Hoffmann (2021a). Density estimation on an unknown submanifold. Electronic Journal of Statistics **15**(1), 2179‑2223.

Berenfeld, C & M Hoffmann (2021b). Density estimation on an unknown submanifold. European Journal of Sport Science **15**(1), 2179‑2223.

Berenfeld, C, P Rosa & J Rousseau (2022a). Estimating a density near an unknown manifold: a Bayesian nonparametric approach. arXiv: 2205.15717 [math.ST].

Berenfeld, C, P Rosa & J Rousseau (2022b). Estimating a density near an unknown manifold: a Bayesian nonparametric approach. arXiv preprint arXiv:2205.15717.

Berry, T & T Sauer (2017). Density estimation on manifolds with boundary. Computational Statistics & Data Analysis **107**, 1‑17.

Breiman, L,WMeisel & E Purcell (1977). Variable Kernel Estimates of Multivariate Densities. Technometrics **19**(2), 135‑144.

Cai, TT & R Ma (2022). Theoretical foundations of t-sne for visualizing high-dimensional clustered data. Journal of Machine Learning Research **23**(301), 1‑54.

Cayton, L (2005). Algorithms for manifold learning. Univ. of California at San Diego Tech. Rep **12**(1-17), 1.

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

Cheng, F, A Panagiotelis & RJ Hyndman (2021). Computationally Efficient Learning of Statistical Manifolds. arXiv: 2103.11773 [cs.LG].

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].

Divol, V (2022). Measure estimation on manifolds: an optimal transport approach. Probability Theory and Related Fields **183**(1), 581‑647.

Do Carmo, MP (1992). “Riemannian manifolds”. In: Riemannian geometry. 2nd ed. Boston: Birkhauser. Chap. 3, pp. 35‑45.

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.

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

Gallot, S, D Hulin & J Lafontaine (2004). "Riemannian metrics". In: Riemannian Geometry. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.51‑127.

Gerber, MS (2014). Predicting crime using Twitter and kernel density estimation. Decision Support Systems **61**, 115‑125.

Hendriks, H (1990). Nonparametric Estimation of a Probability Density on a Riemannian Manifold Using Fourier Expansions. Annals of Statistics **18**(2), 832‑849.

Henry, G, A Munoz & D Rodriguez (2013). Locally adaptive density estimation on Riemannian manifolds. SORT-Statistics and Operations Research Transactions **37**(2), 111‑130.

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

Izenman, AJ (1991). Review Papers: Recent Developments in Nonparametric Density Estimation. Journal of the American Statistical Association **86**(413), 205‑224.

Izenman, AJ (2012). Introduction to manifold learning. WIREs Comp Stat **4**(5), 439‑446.

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.

Jones, MC (1990). Variable kernel density estimates and variable kernel density estimates. The Australian Journal of Statistics **32**(3), 361‑371.

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

Lee, JA &MVerleysen (2007). Nonlinear Dimensionality Reduction. Springer Science & Business Media.

McQueen, J,MMeil˘a, J VanderPlas & Z Zhang (2016). Megaman: Scalable Manifold Learning in Python. J. Machine Learning Research **17**(148), 1‑5.

Nakahara, M (2018). Geometry, topology and physics. CRC press.

Okabe, A, T Satoh & K Sugihara (2009). A kernel density estimation method for networks, its computational method and a GIS-based tool. Geographical Information Systems **23**(1), 7‑32.

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].

Scott,DW(2015). Multivariate Density Estimation: Theory, Practice, and Visualization. JohnWiley & Sons.

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

Wickham, H, D Cook, H Hofmann & A Buja (2011). tourr: An R Package for Exploring Multivariate Data with Projections. Journal of Statistical Software **40**, 1‑18.

Xie, Z & J Yan (2008). Kernel density estimation of traffic accidents in a network space. Computers, Environment and Urban Systems **32**(5), 396‑406.

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. JMLRWorkshop and Conference Proceedings, pp.892‑900.

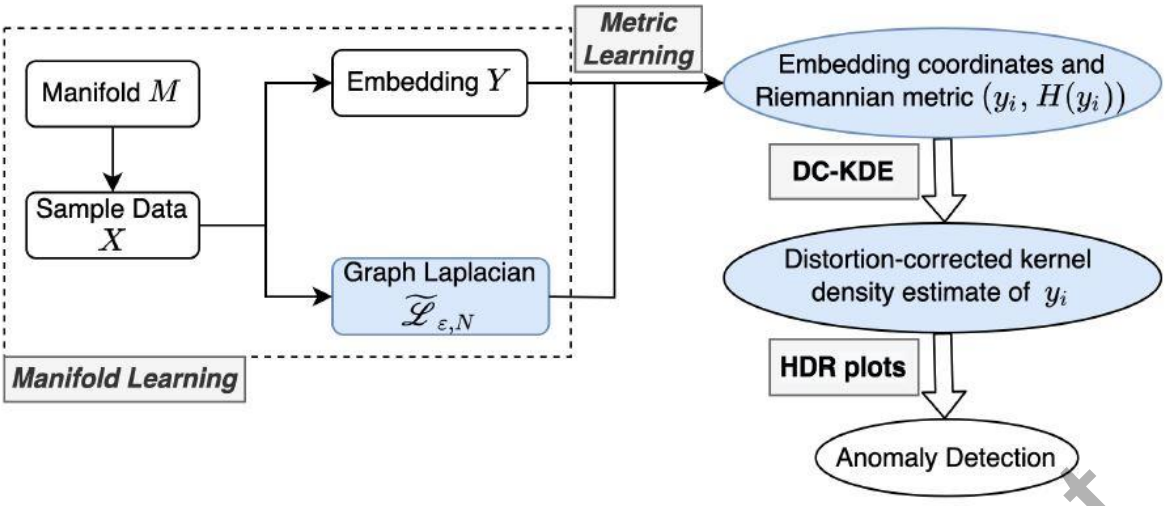

Figure 1: The proposed schematic for anomaly detection with distortion-corrected kernel density estimates.

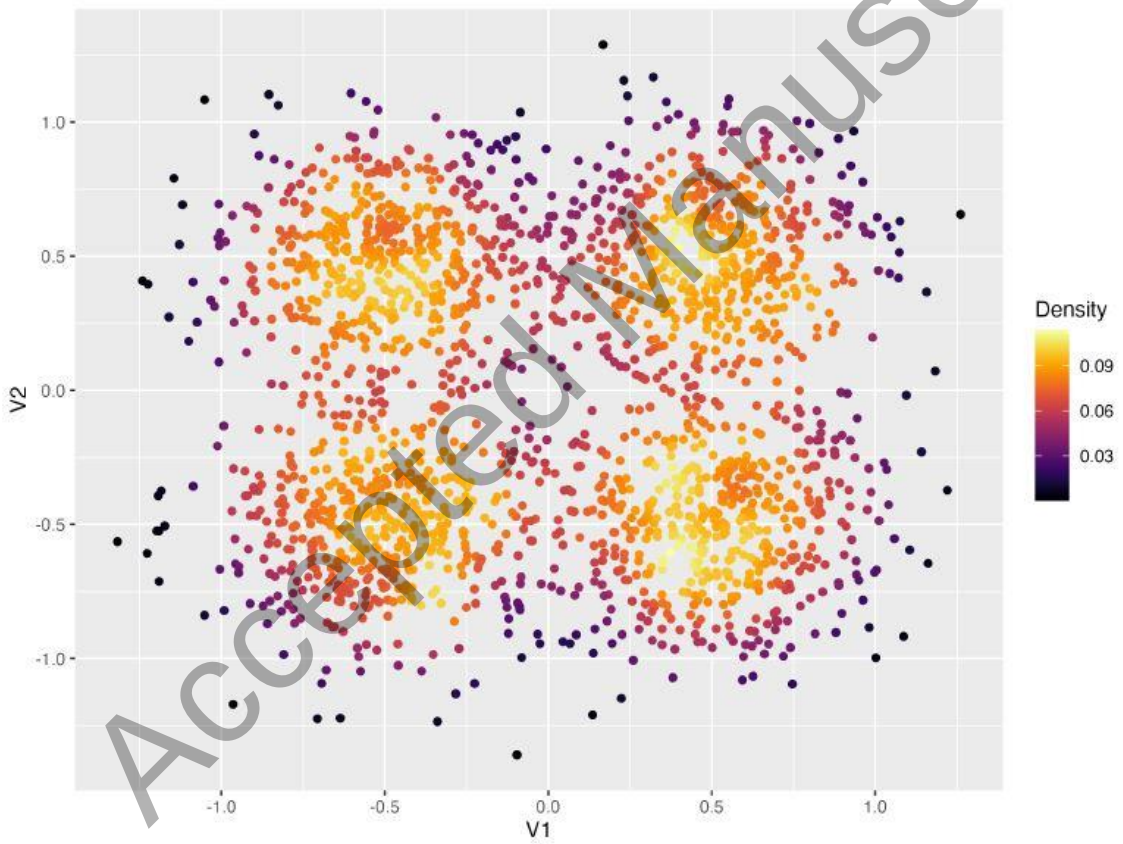

Figure 2: Underlying data for the Gaussian mixture model of four kernels with means $(0.25, 0.25), (0.25, 0.75), (0.75, 0.25), (0.75, 0.75)$ and the same variance-covariance matrix $diag(0.016, 0.016)$. The colors indicate the true density of the data when they are mapped via the twin peaks function.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

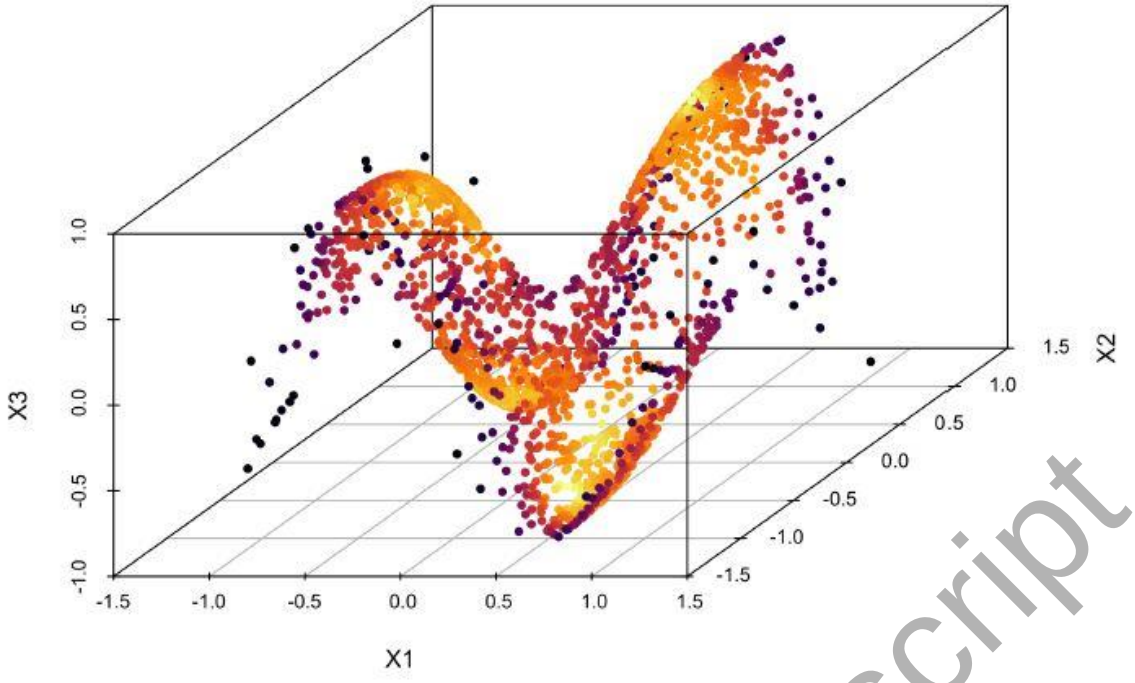

Figure 3: Scatterplot of the 3-D twin peaks data with the same colors indicating the true density as in Figure 2.

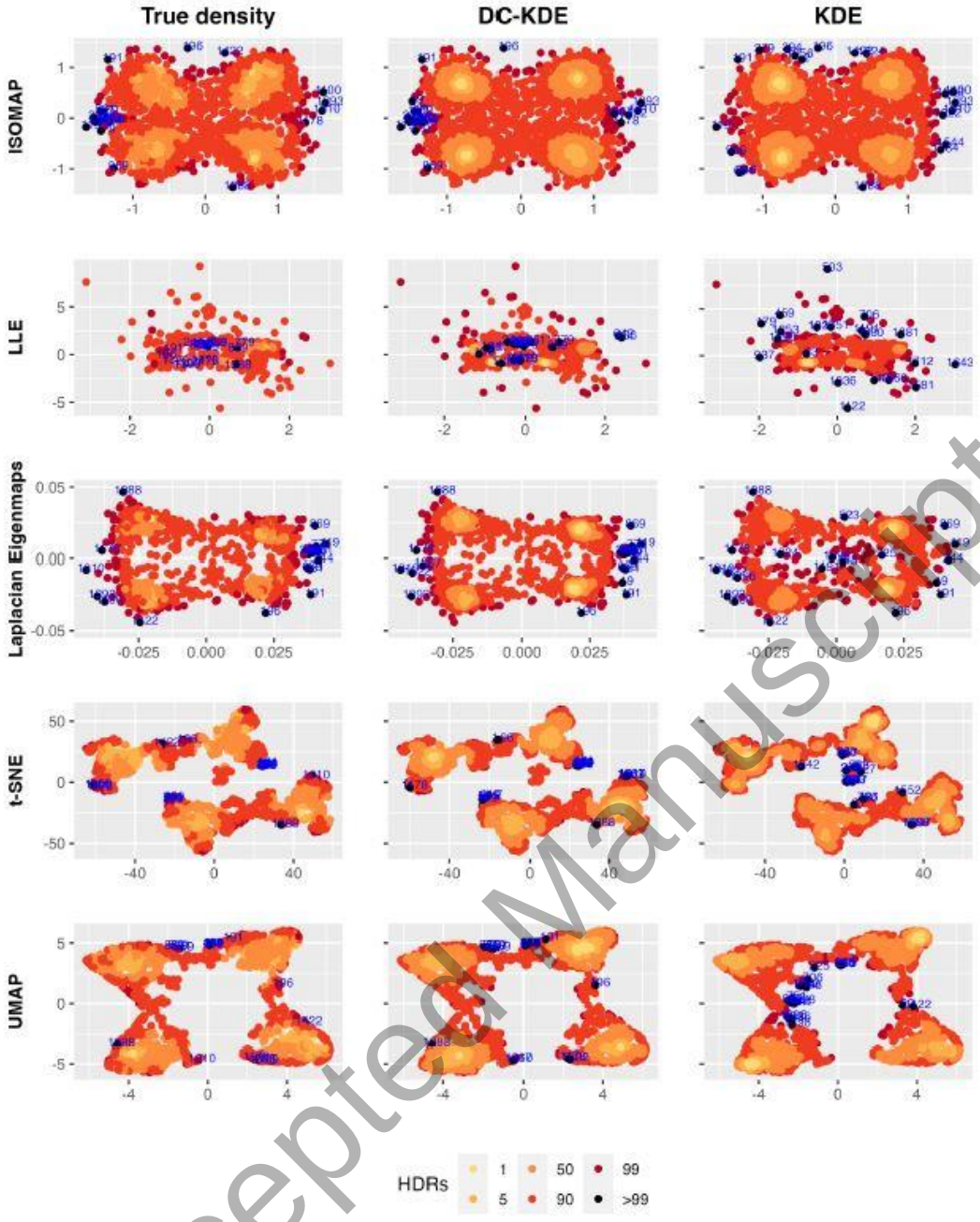

Figure 4: Highest density region plots of five manifold learning embeddings of the twin peaks data in each row. The top 20 outliers, highlighted in black and indexed in blue text, are found by the true manifold density (left panel), DC-KDE (middle panel), and KDE (right panel). DC-KDE finds more true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

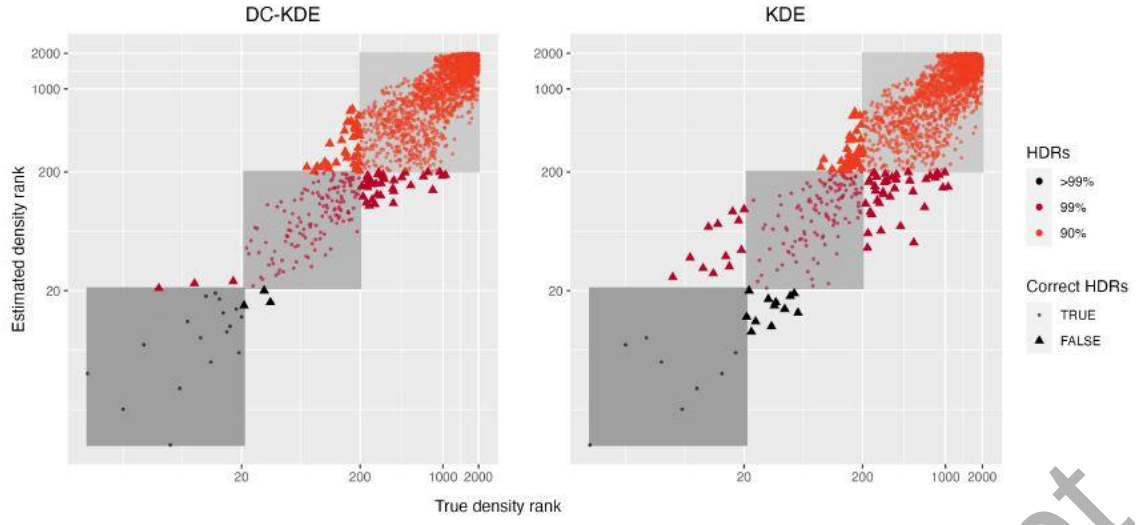

Figure 5: Scatterplot of log scale ranks of true density and estimated density ranks for DC-KDE (in the left panel) and KDE (in the right panel) based on ISOMAP embedding of the twin peak data. The colors indicate different levels of highest density regions and the shapes indicate whether the density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

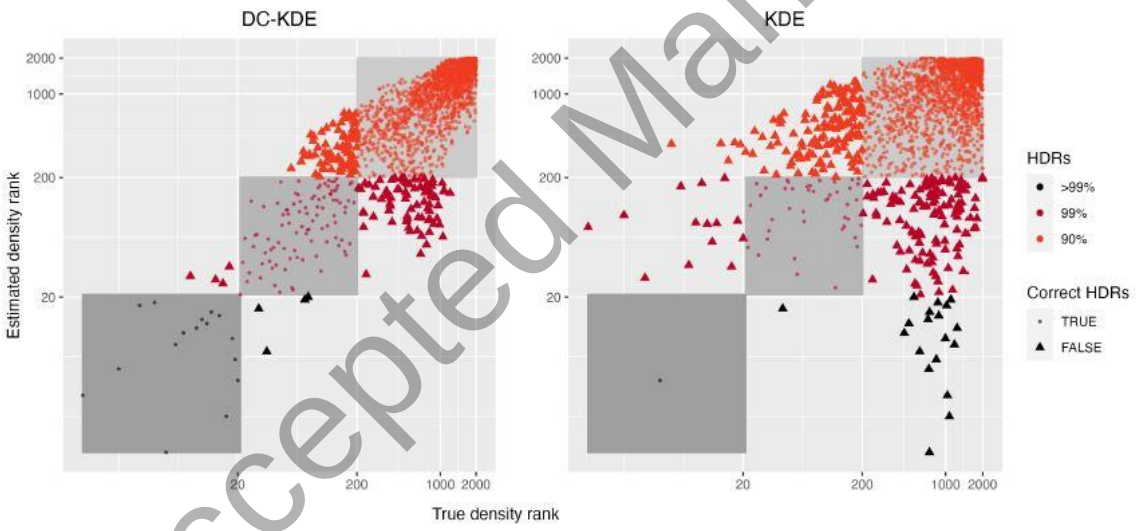

Figure 6: Scatterplot of log scale ranks of true density and estimated density ranks for DC-KDE (in the left panel) and KDE (in the right panel) based on t-SNE embedding of the twin peak data. KDE without distortion correction gives many more misclassified anomalies.

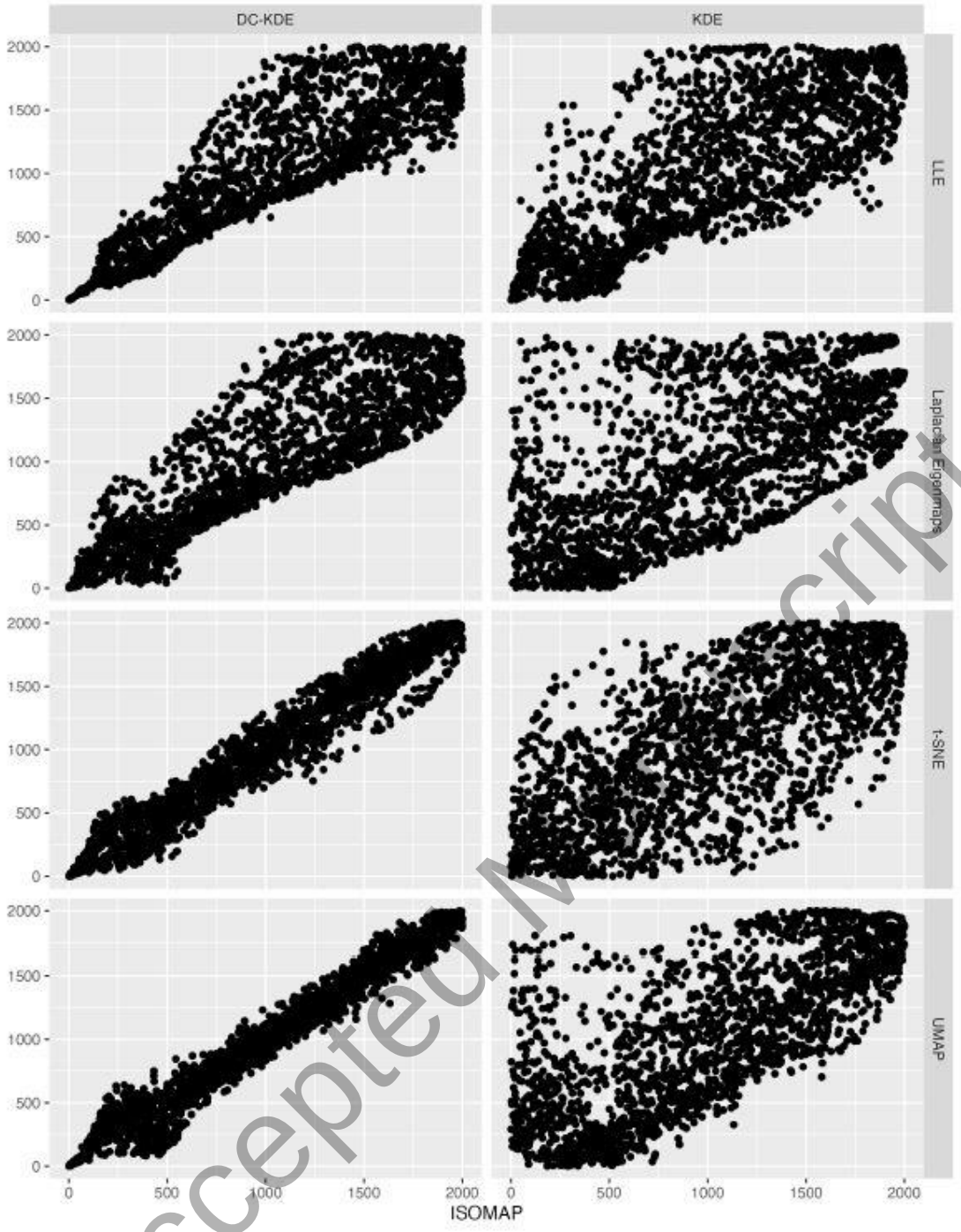

Figure 7: Comparison of ranks of the estimated densities for the twin peak example based on ISOMAP and four other dimension reduction algorithms in each row. Distortion-corrected KDE (on the left panel) and KDE (on the right panel) are compared and DC-KDE shows the robustness to the use of different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

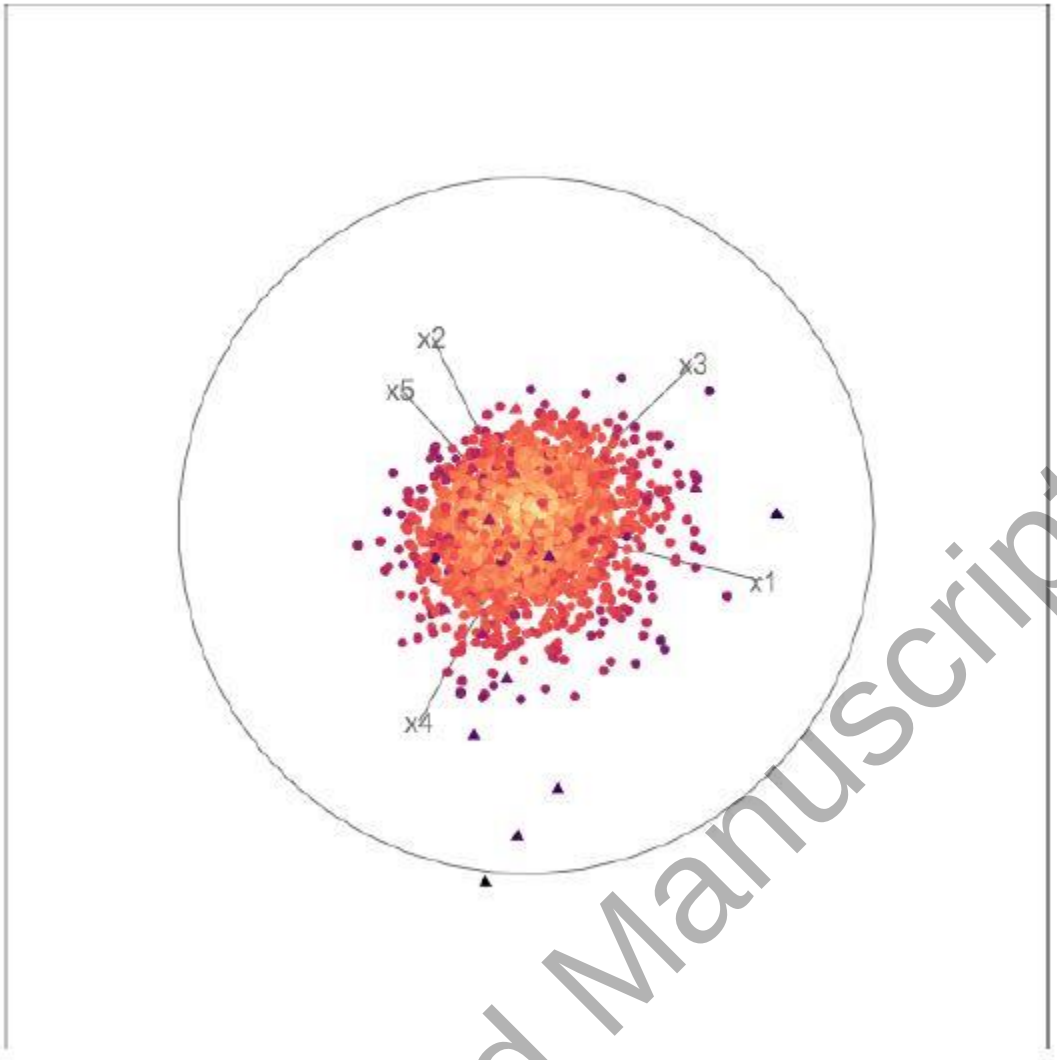

Figure 8: Scatterplot display of the animation of a 5-D tour path with shapes indexing two Gaussian mixture components and the colors showing the distance to the kernel cores. Distant points in darker colors could be seen as anomalies.

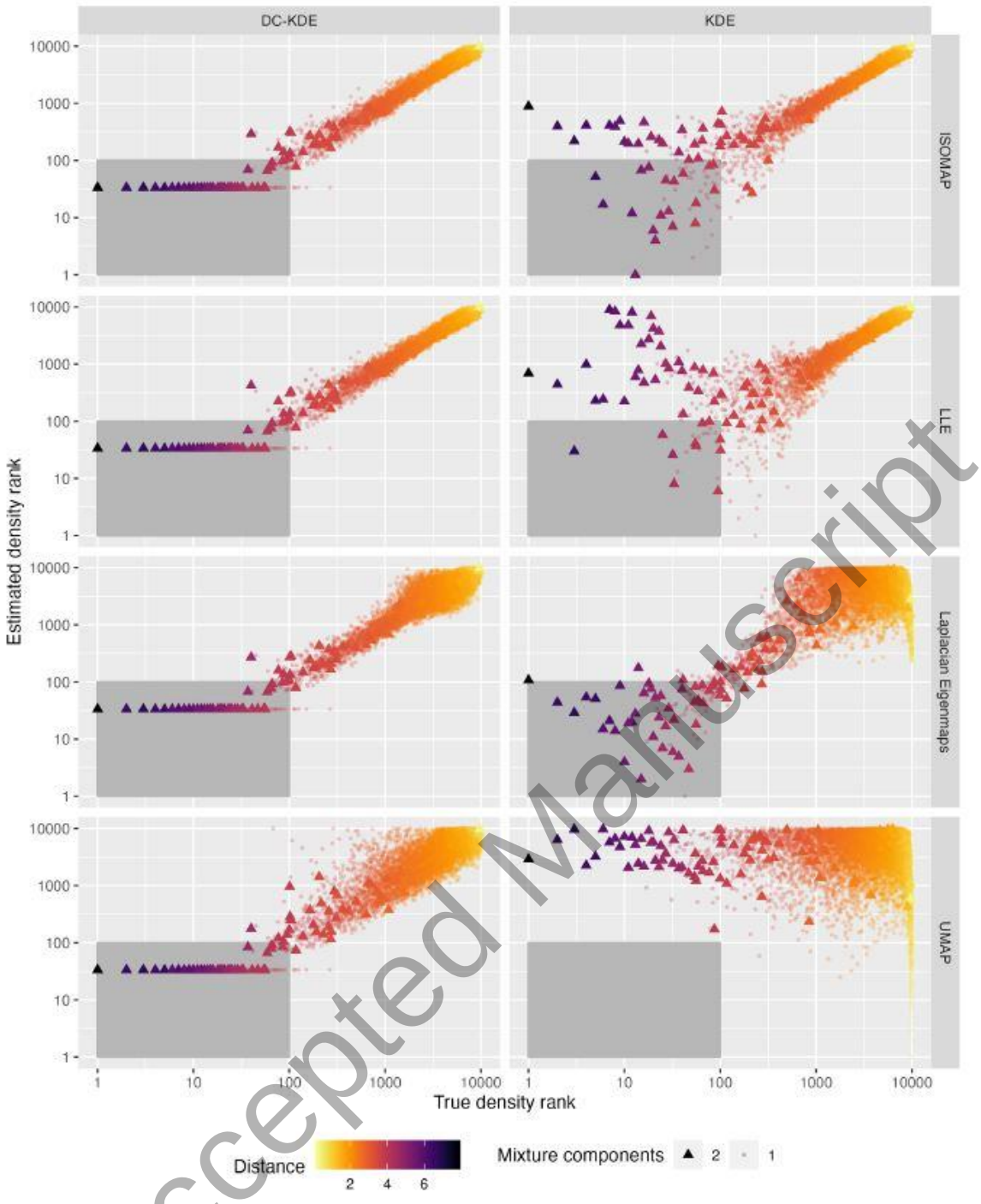

Figure 9: Rank comparison between the true density and estimated density from both DC-KDE and KDE for the semi-hypersphere example. Four manifold learning methods are used row-wise. The point shapes indicate whether they are the true outliers, and the grey shading highlights the top 1% rank region. The colors show the distance to the center of the semisphere, with darker points being distant from the center.

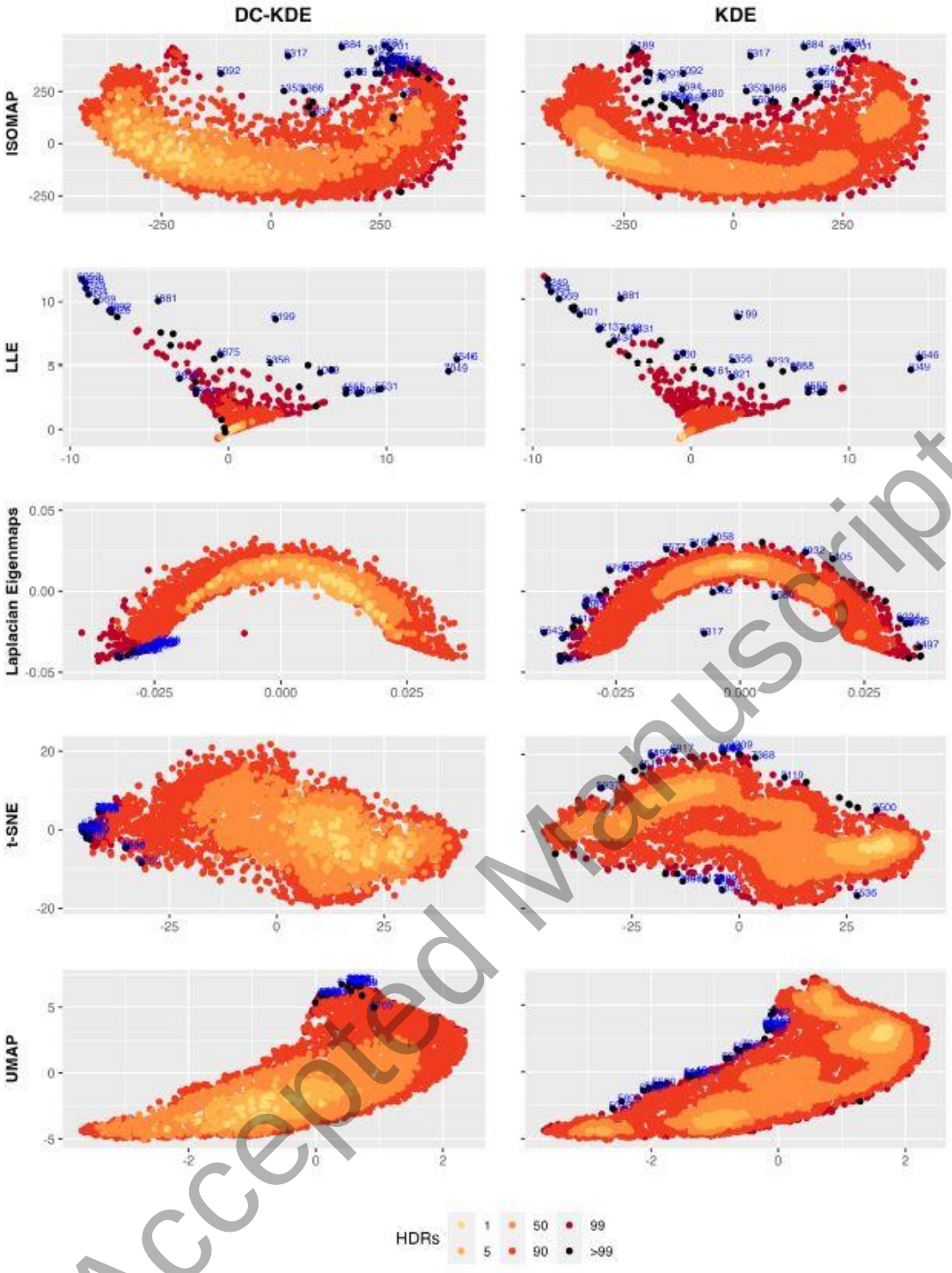

Figure 10: Highest density region plots of the 2-D embedding for 3, 639 households, with each point representing the distribution of one household and colors indicating the density estimated from DC-KDE (left panel) and KDE (right panel). The black points with blue text indexing are the top 20 anomalous households found by each dimension reduction method in each row.

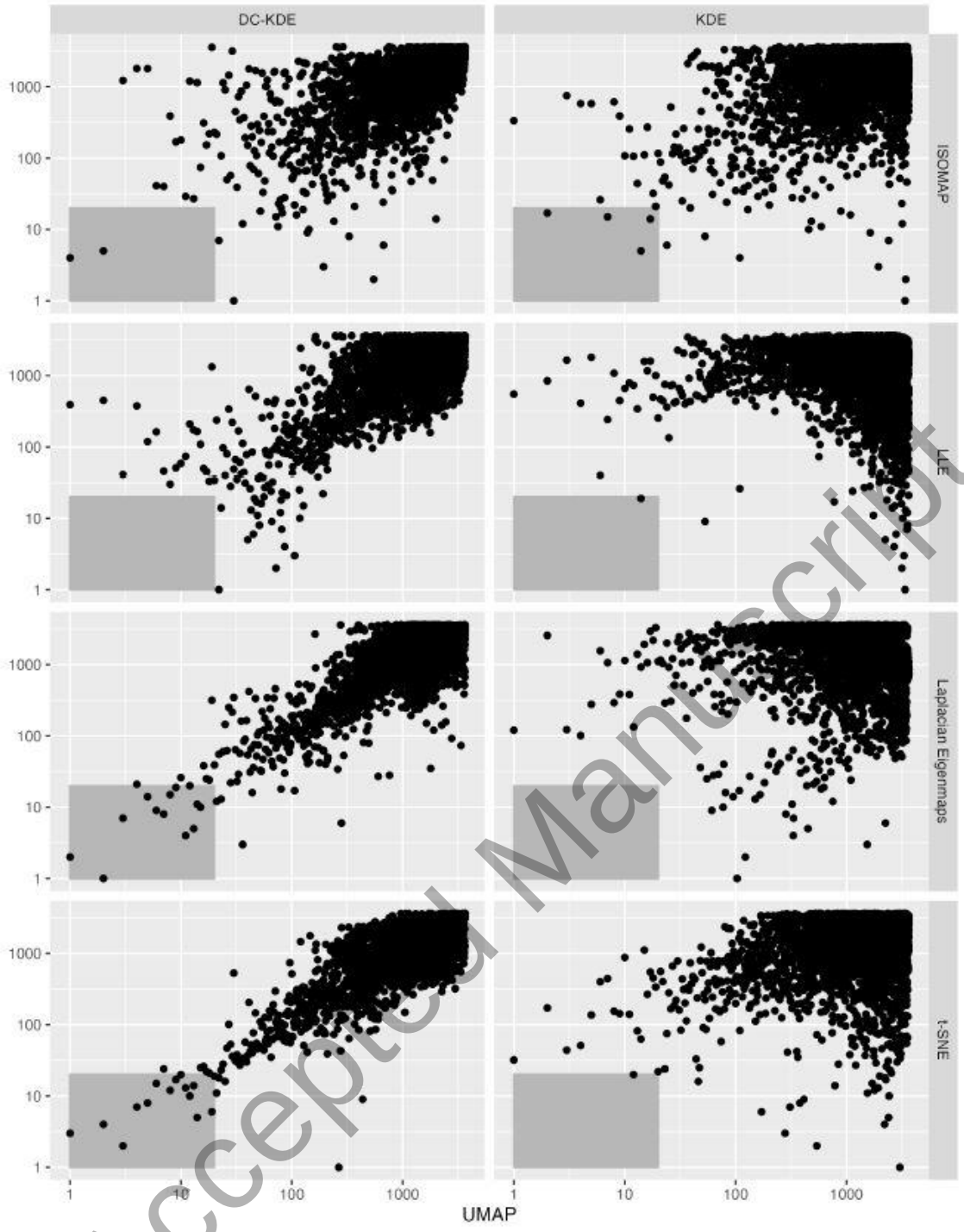

Figure 11: Comparison of ranks of the estimated densities based on UMAP and four dimension reduction algorithms in each row. Distortion-corrected KDE (on the left panel) and KDE (on the right panel) are compared and DC-KDE shows a higher level of robustness to the use of different dimension reduction methods.

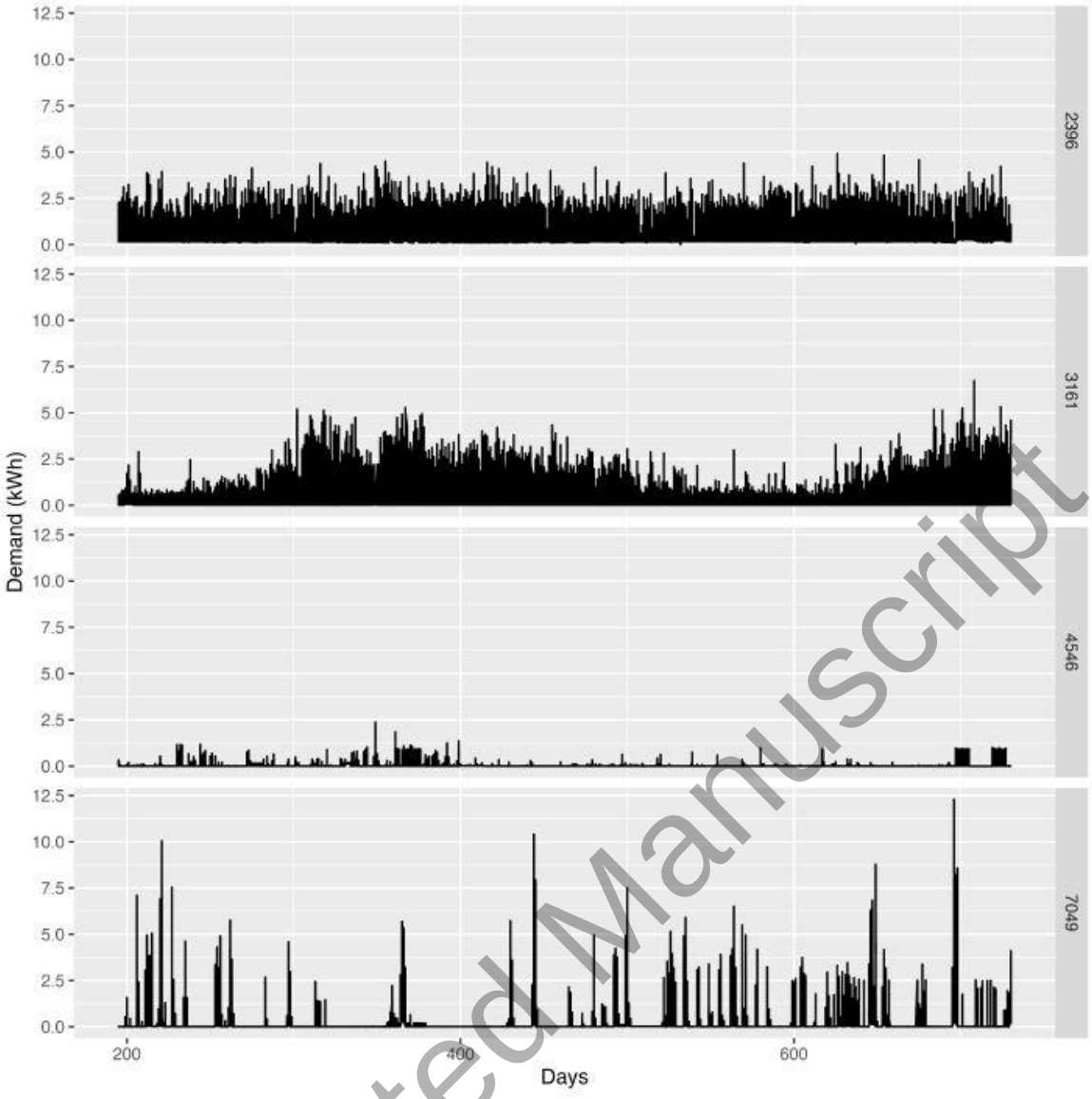

Figure 12: Electricity usage plots of all 535 days for the most typical household (top row) and three anomalies.

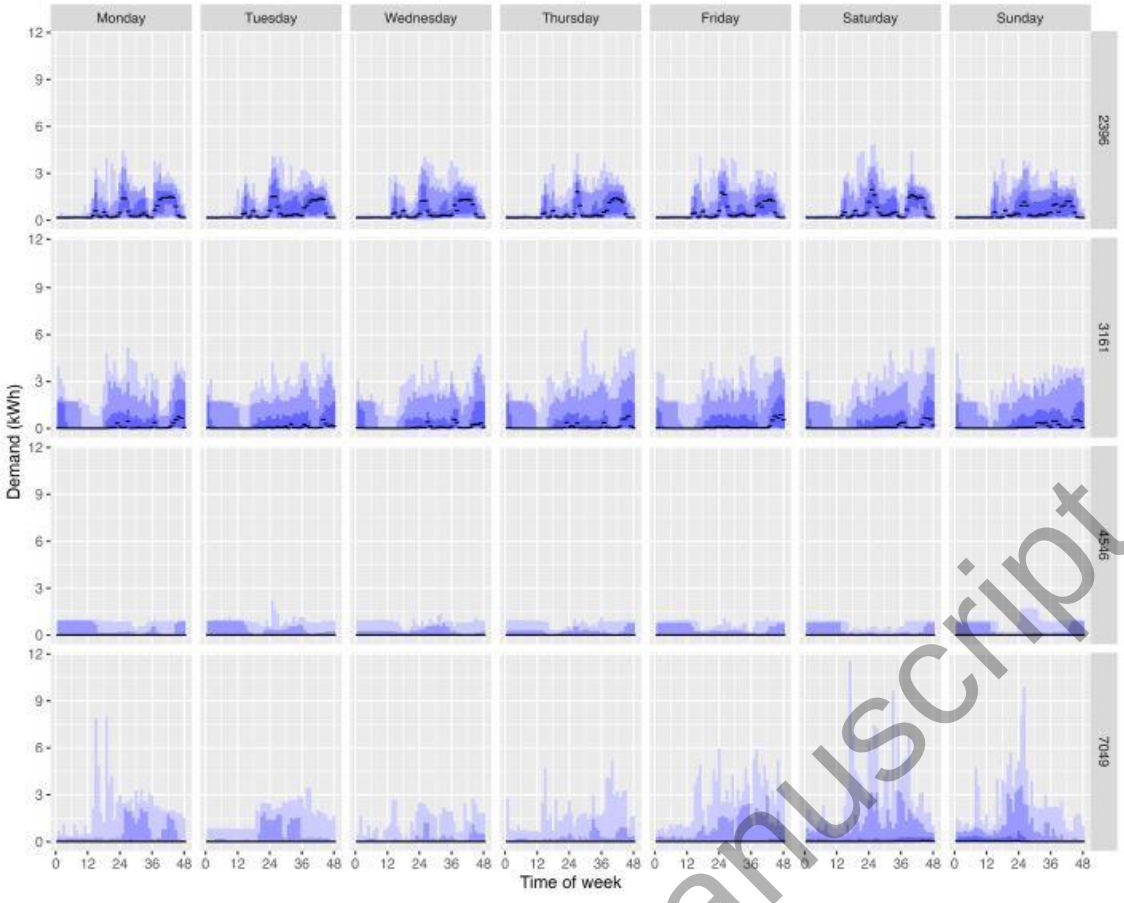

Figure 13: Quantile region plots of electricity demand against the time of the week for one typical household 1472 and three anomalies, 3161, 4546, and 7049. The day-of-the-week patterns are much more different for Household 4546 and 7049 compared with 3161.

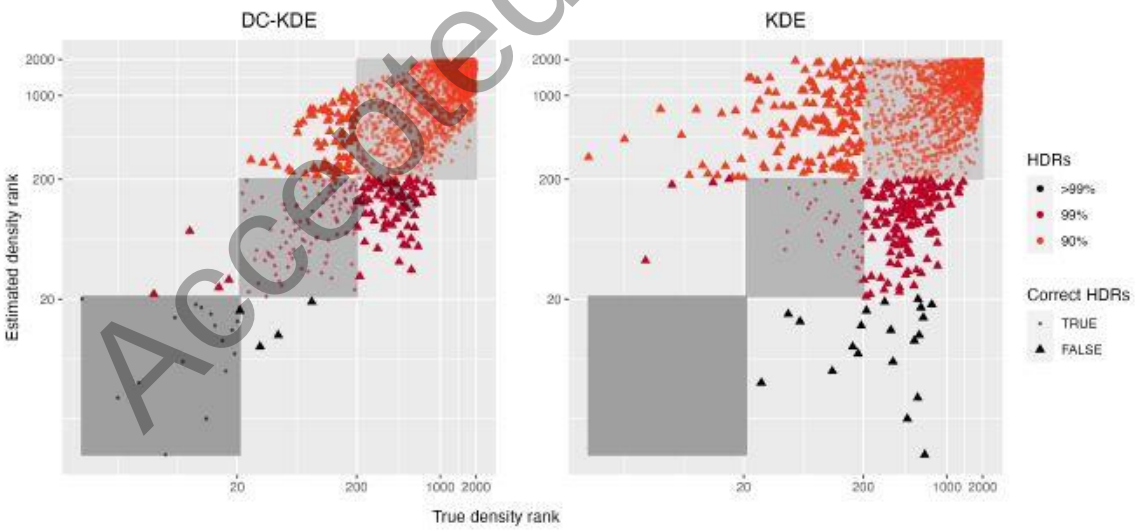

Figure 14: Scatterplot of true density and estimated density ranks of LLE embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.

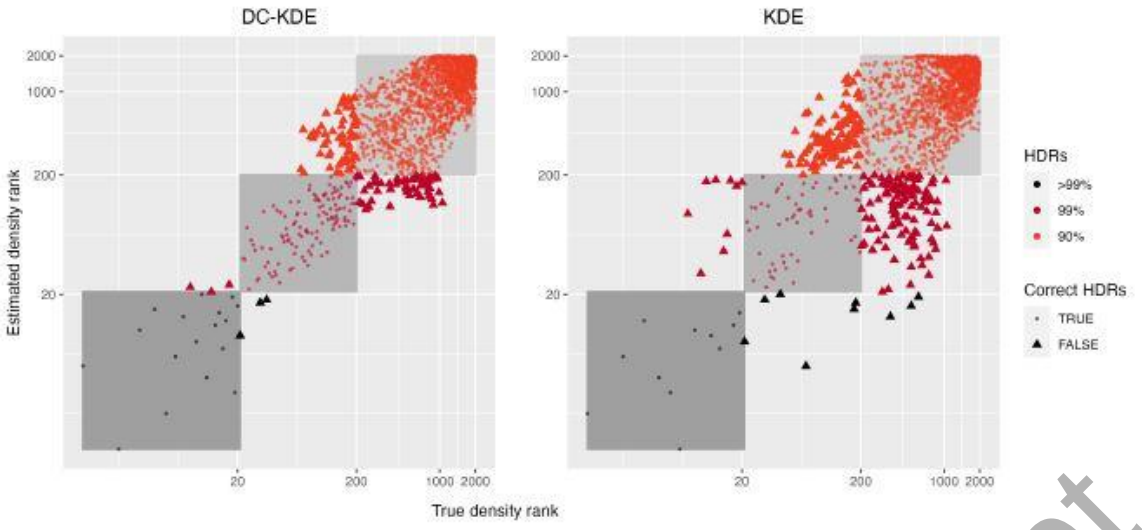

Figure 15: Scatterplot of true density and estimated density ranks of Laplaxian Eigenmaps embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared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

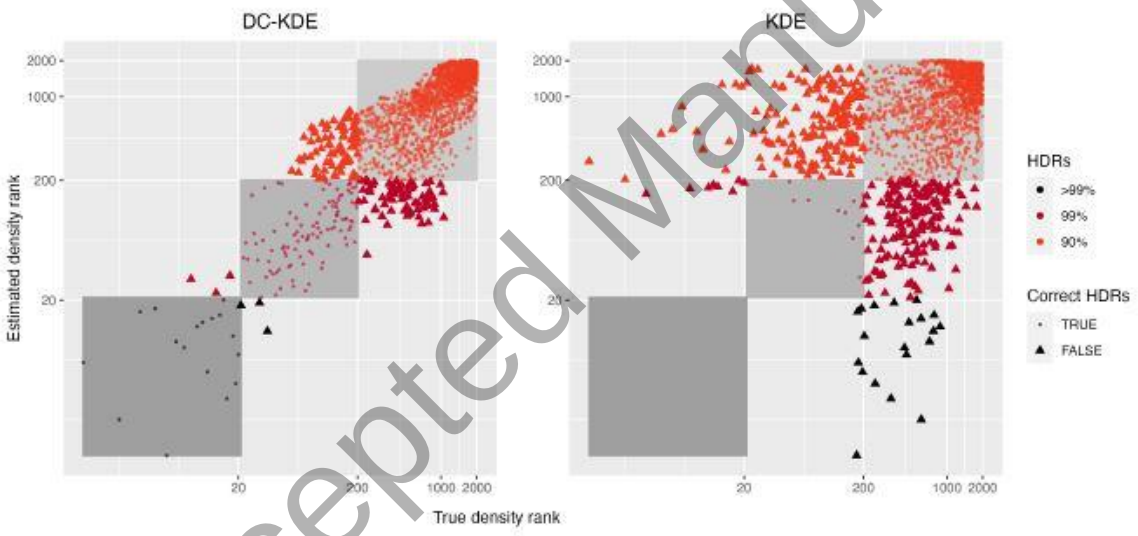

Figure 16: Scatterplot of true density and estimated density ranks of UMAP embedding for DC-KDE and KDE, with colors indicating the absolute rank errors weighted by the sum of true and estimated ranks. DC-KDE shows a strong linear positive relationship with a higher rank correlation compared to KDE.

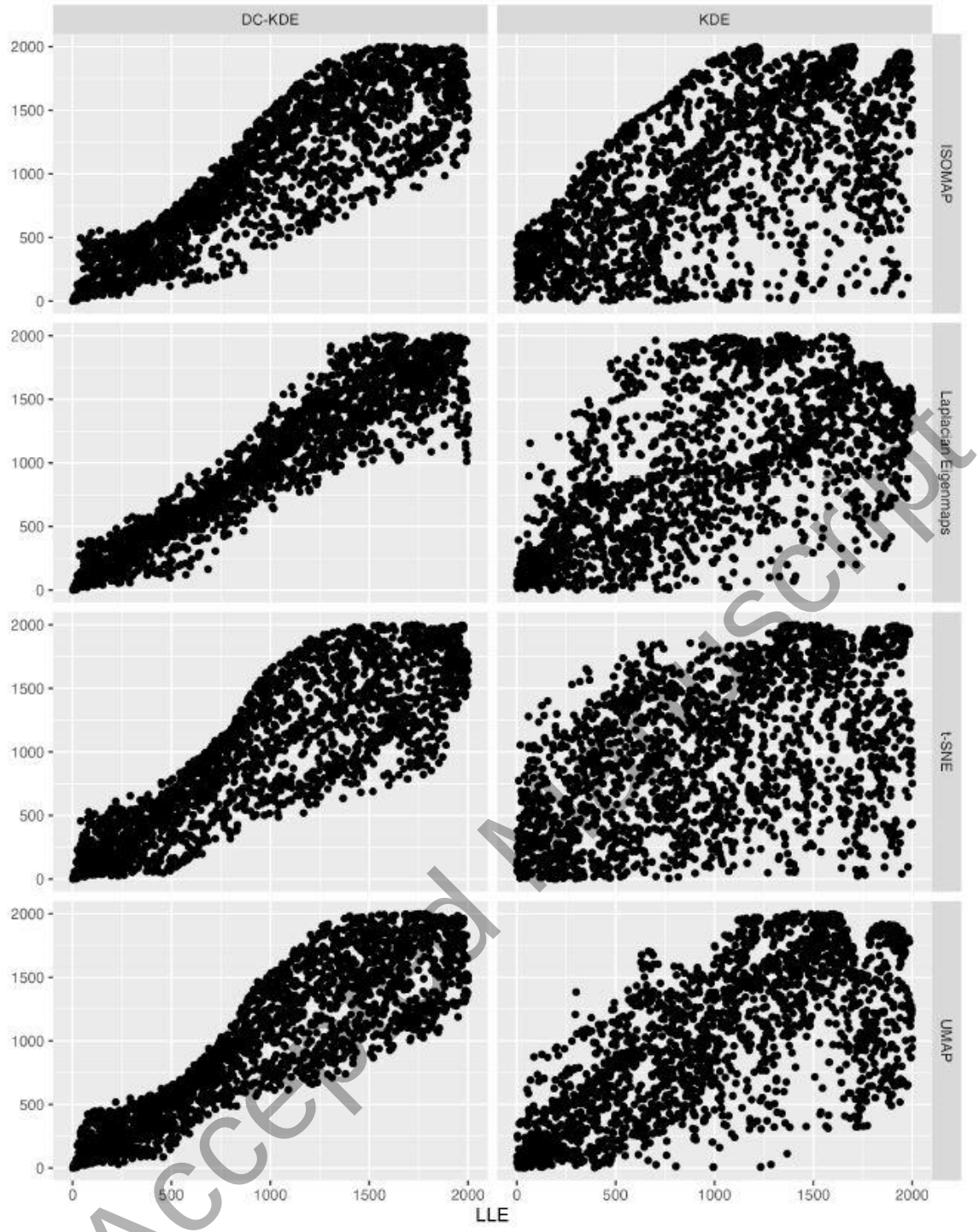

Figure 17: Comparison of outliers found by LLE compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel). The four colors and shapes represent the four gaussian kernels in the 2-D metadata. Outliers found by DC-KDE are more consistent regardless of the manifold learning embedding.

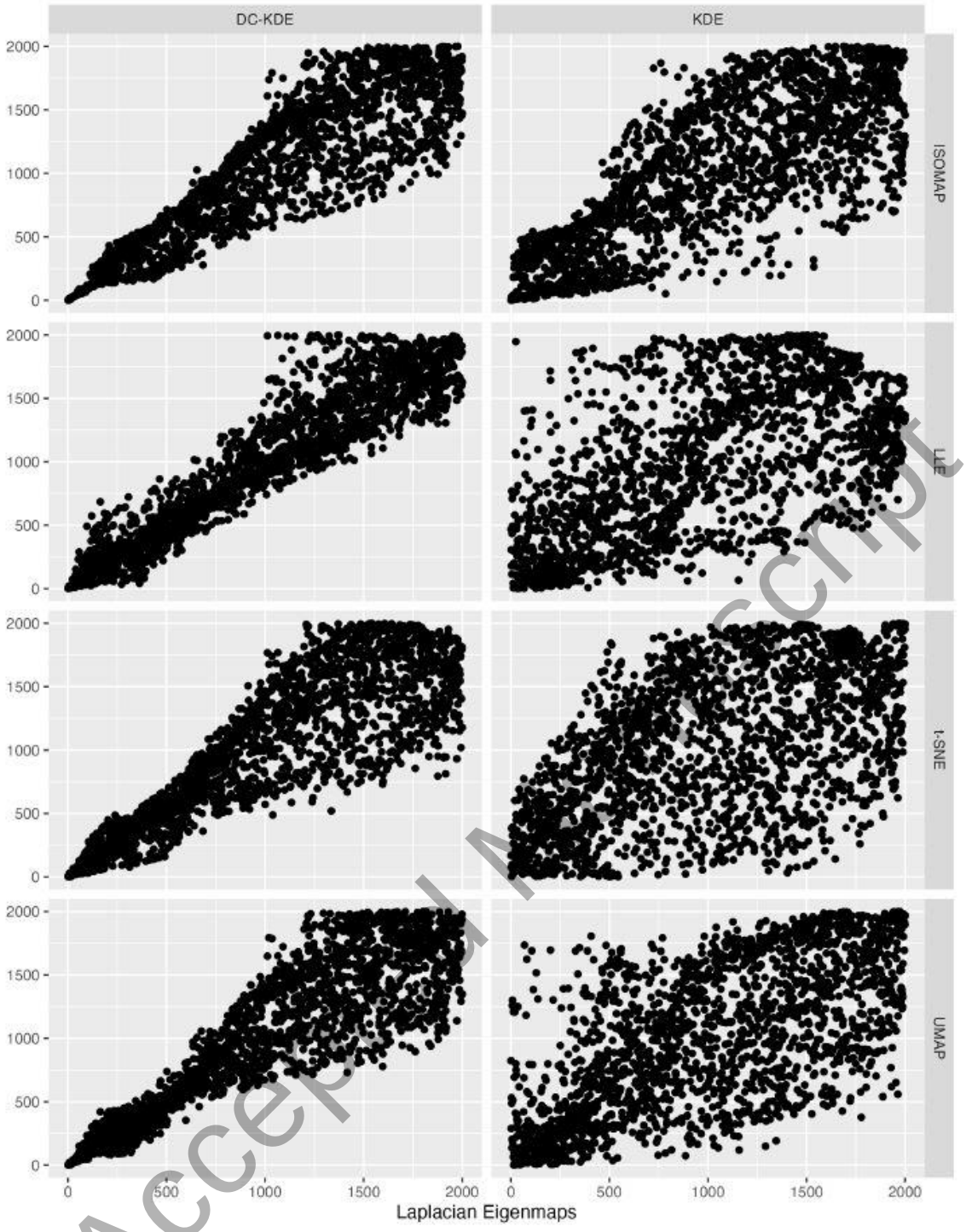

Figure 18: Comparison of outliers found by Laplacian Eigemaps compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel).

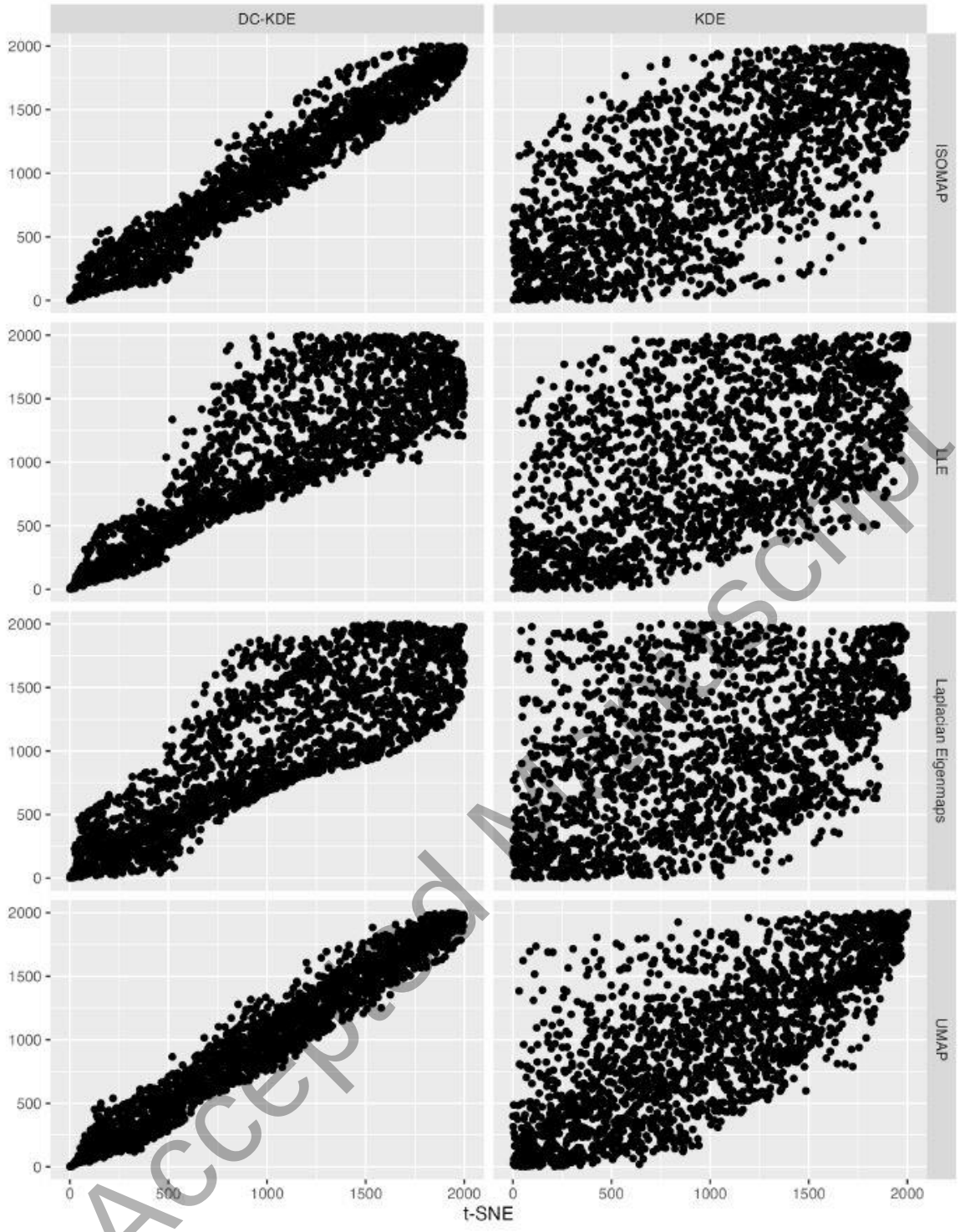

Figure 19: Comparison of outliers found by t-SNE compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel).

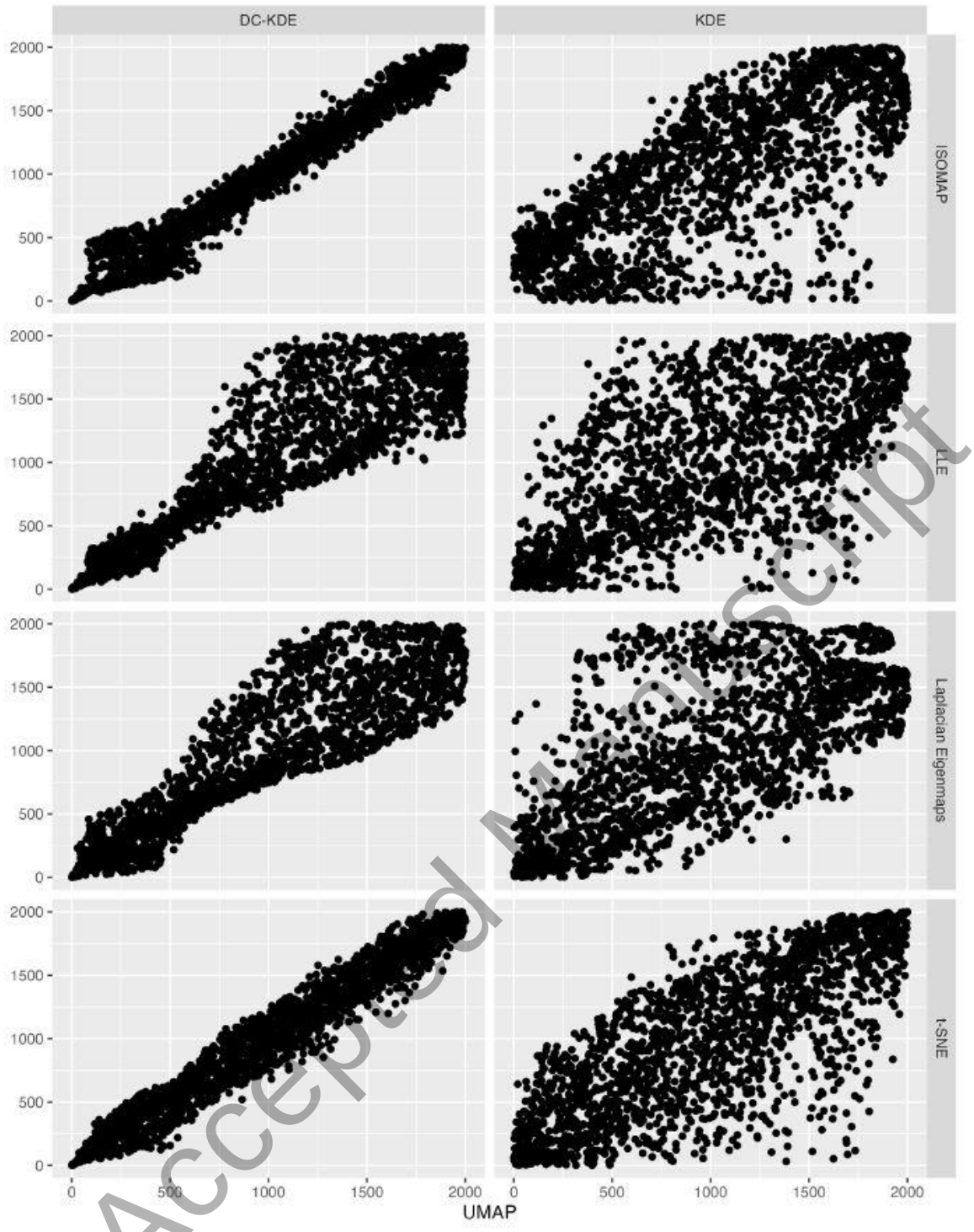

Figure 20: Comparison of outliers found by UMAP compared to the other four for DC-KDE (on the left panel) and KDE (on the right panel).

Table 1: Correlation between true density ranking and estimated density ranking for different manifold learning embeddings of the twin peak data. Distortion-corrected KDE outperforms 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

Table 2: Correlation between true density and estimated density for four manifold learning embeddings of the 100-D data. DC-KDE has a slightly lower rank correlation for ISOMAP and LLE above 0.96, but significantly outperfoms KDE for Laplacian Eigenmaps and UMAP when more distortion is induced in dimension reduction.

	ISOMAP	LLE	Laplacian.Eigenmaps	UMAP
DC-KDE	0.968	0.970	**0.867**	**0.782**
KDE	**0.976**	**0.971**	0.033	-0.181

Table 3: Proportion comparison of outliers in correct highest density regions in density estimation of four manifold learning embeddings for the 100-D data.

	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

Table 4: Top 20 anomalous household IDs using density estimates of different 2-dimensional embedding. The left five columns are anomalies by DC-KDE, while the right five columns are anomalies using KDE. The blue-colored IDs are IDs that are detected over three times, with the darker one indicating five times and the lighter one indicating four times.

	DC-KDE					KDE				
	ISOMAP	LLE	Laplacian Eigenmaps	t-SNE	UMAP	ISOMAP	LLE	Laplacian Eigenmaps	t-SNE	UMAP
1	6317	**7049**	**3161**	4826	**4546**	6317	1881	6317	4190	6533
2	1353	**4546**	5269	1069	**7049**	4884	**6199**	6643	6817	5899
3	4884	1881	4875	**4546**	1069	5092	**5356**	6764	6492	6345
4	**4546**	6952	7300	**7049**	4862	1353	4233	1058	2309	1884
5	**7049**	6838	4880	**1533**	3639	5580	**7049**	5105	2546	3278
6	1366	3874	5599	4265	**1964**	5594	1821	6577	2907	5463
7	**3161**	1069	3222	4862	2249	2576	**4546**	3880	4536	5504
8	1074	2249	1776	3639	**6199**	1366	3431	5358	6898	1782
9	6681	**1964**	1161	5546	3892	6681	1569	1366	4128	5965
10	1701	1569	3655	4896	6209	5291	**1964**	5545	7368	6081
11	3136	**1533**	6169	4555	6952	**3161**	6568	2160	2232	4894
12	1881	4555	4830	**6199**	4896	1701	7300	6585	6273	5937
13	6785	6526	2284	6952	6838	4749	7438	5824	1557	5522
14	5092	3639	6526	6838	**1533**	6699	4401	4932	3500	5580
15	5269	**6199**	1694	**1964**	1569	5504	3434	6410	1932	6902
16	2576	5531	3431	7104	3213	3170	2249	3023	3119	7243
17	1262	4875	2523	3892	4401	5899	3213	1497	3809	6699
18	5966	**5356**	**5356**	**3161**	6766	2658	**3161**	7293	6108	5549
19	5937	3892	2233	6799	4265	5189	**1533**	5580	4516	5048
20	**5356**	4896	5125	6209	6799	2669	4555	6224	5937	1923

Table 5: Proportion of the same outliers as UMAP in two highest density regions when estimating densities for four manifold learning embeddings.

	ISOMAP		LLE		Laplacian Eigenmaps		t-SNE	
	DC-KDE	KDE	DC-KDE	KDE	DC-KDE	KDE	DC-KDE	KDE
> 99% HDR	0.189	**0.243**	**0.189**	0.027	**0.622**	0.000	**0.865**	0.108
99% HDR	**0.359**	0.248	**0.503**	0.000	**0.697**	0.007	**0.628**	0.083