Two-Stage Manifold-Based Denoising and Outlier Detection for High-Dimensional Data

Lan Li, Dan Wang, Mingli Jing and Shang Bao

***Abstract*—To effectively remove noise and detect outliers from high-dimensional data, we propose a two-stage manifold denoising and outlier detection algorithm while preserving the manifold structure. In the pre-processing stage, the algorithm partitions the original data into multiple local patches covering the entire dataset. For each patch, the covariance matrix is computed, and the affine invariant riemannian metric (AIRM) is utilized to measure geometric differences between patches. Riemannian gradient descent is then applied on the symmetric positive definite (SPD) manifold to achieve denoising result. In the post-processing stage, manifold learning and the local outlier factor (LOF) algorithm are introduced to the dimension-reduced data for outlier detection, further enhancing data quality and usability. Finally, the proposed algorithm is comprehensively evaluated on eight synthetic datasets and six real-world datasets. The results demonstrate that the proposed algorithm significantly reduces MSE in recovering low-dimensional manifold structures, denoising reconstruction, and outlier detection, while achieving superior performance in metrics such as AUC, GDNRR, SNR, and manifold scores. Joint analysis of bandwidth and neighborhood distance thresholds, along with ablation experiments, further confirms the robustness of the two-stage processing in preserving both global and local geometric structures.**

***Index Terms*—Manifold Denoising, Outlier Detection, Manifold Learning, High-Dimensional.**

I. Introduction

IN high-dimensional data analysis, as the dimensionality of data increases, data points often become more and more sparse, with complex distributions, making traditional methods ineffective and unreliable for handling such data. Traditional dimensionality reduction methods, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), rely on linear assumptions about the data. However, for high-dimensional data with nonlinear structures, these traditional methods may fail to capture the geometric relationships within the data. Thus, it is necessary to explore more flexible nonlinear dimensional-reduction techniques to adapt to the complex structures of high-dimensional data.

Manuscript received December 4, 2024; revised June 17, 2025. This work was supported by the Natural Science Basic Research Program of Shaanxi (Program No. 2025JC-YBMS-078 and the Graduate Innovation and Practical Ability Training Project of Xi'an Shiyou University (YCS23213192).

Lan Li is a professor at the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (Corresponding author. Phone: +86-029-88382735, e-mail: lanli@xsyu.edu.cn).

Dan Wang is a postgraduate student at the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: wwdd529@163.com).

Mingli Jing is an associate professor of School of Electronic Engineering, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: mljingsy@xsyu.edu.cn).

Shang Bao is a postgraduate student at the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: rickybss@163.com).

Unlike traditional linear methods, manifold learning approaches are more suited for high-dimensional data with nonlinear structures. Manifold learning assumes that high-dimensional data are distributed on a low-dimensional manifold embedded in a high-dimensional space. By analyzing local geometric relationships in the data, manifold learning identifies the underlying low-dimensional structure and extracts low-dimensional embeddings, thereby revealing the intrinsic geometric structure of the data. Common manifold learning algorithms include Local Linear Embedding (LLE) [7], Isometric Mapping (ISOMAP) [8], and Laplacian Eigenmaps (LE) [9]. These algorithms preserve the geometric properties of local neighborhoods and effectively map high-dimensional data to a lower-dimensional space.

Manifold learning is widely recognized for its effectiveness in analyzing high-dimensional data and has been applied to various fields, including image processing [1], bioinformatics [2], computer vision [3], natural language processing [4], gene data analysis [5], and financial data modeling [6]. Manifold-based methods provide a robust analytical tool for handling complex high-dimensional data.

Noise is unavoidable in high-dimensional data collection, often arising from measurement errors or environmental factors and causing deviations from the true manifold structure. In such cases, the performance of manifold learning algorithms will degrade due to the presence of noise, preventing accurate representation of the underlying manifold. Therefore, noise removal is essential to ensure the effectiveness of manifold learning algorithms. Manifold denoising aims to preserve the low-dimensional manifold structure of data while reducing the interference of noise, thus generating clean data points that are closer to the underlying manifold.

To preserve the low-dimensional manifold structure of data during denoising, several studies develop different algorithms. These methods aim to effectively remove noise while retaining the main features of the data. Yao et al. [10] present the recursive empirical mode decomposition algorithm (EMD), an adaptive technique for decomposing nonlinear and non-stationary data. Park et al. [11] propose the non-local means (NLM) algorithm, which compares and weights similar patterns within the dataset to achieve denoising, particularly effective for removing Gaussian noise. Lin et al. [12] suggest a graph-based filtering approach. By applying filtering operations along graph edges, this method is well-suited for graph-structured data. Sober et al. [13] formulate the moving least squares (MLS) technique. This approach reduces noise by projecting noisy samples onto a smooth manifold while preserving the manifold structure. Lyu et al. [14] establish nonlinear robust

PCA (NRPCA), extending robust PCA to nonlinear settings by leveraging local patches and curvature information on data manifolds for denoising. Jiang et al. [15] design a manifold embedding-based domain adaptation algorithm (eSPDA) for decoding electroencephalogram (EEG) signals in motor imagery brain-computer interface (MI-BCI) systems. By integrating Laplacian embedding and domain alignment techniques, eSPDA generates low-dimensional submanifold embeddings, effectively reducing domain shifts while preserving local manifold geometric structures. Meng et al. [16] construct an unsupervised feature selection algorithm (LRLSP) based on Laplacian rank constraints and local structure preservation, which builds sparse block-diagonal similarity matrices to achieve unified feature selection and manifold structure preservation in high-dimensional data.

Outliers, refer to data points that significantly deviate from the distribution of most other points. These points are often caused by special events or anomalies and not random in nature but instead appear as extreme outliers. Detecting outliers is essential because these points can distort the overall structure of the data and negatively impact the performance of algorithms. The goal of manifold outlier detection is to identify points that deviate from the manifold structure. Widely applied algorithms for outlier detection include the local outlier factor (LOF) [17] and isolation forest [18], which detect outliers by comparing the local density of data points or by isolating outliers through random trees. Additionally, adversarial examples are often regarded as a specific type of outliers that deviate from the true manifold. Manifold-based adversarial denoisers, such as APE-GAN++ [19], utilize generative adversarial networks (GANs) to map adversarial examples back to the manifold region of benign examples, thereby enhancing the robustness of neural networks against adversarial attacks.

Despite significant progress made by existing methods, several issues and challenges remain:

• Current algorithms often focus on either noise removal or outlier detection, which reduces their effectiveness in handling complex data and preserving the manifold structure, as they lack the ability to address both simultaneously.

• Methods either focus on the fine processing of local structures, such as local regression and local smoothing techniques, or examine the global distribution, such as global probability models. This separation may fail to account for the interaction between local structures and global distributions, limiting the effectiveness of algorithms on complex data structures.

Based on the aforementioned problem, this paper proposes a two-stage denoising and outlier detection algorithm. In the preprocessing stage, the algorithm partitions the high-dimensional data into multiple patches collectively covering the entire dataset. Each patch is aligned by centering its coordinates and assessed for geometric similarity using the Affine Invariant Riemannian Metric (AIRM), thereby integrating local geometric features with the global data distribution. Riemannian gradient descent on the Symmetric Positive Definite (SPD) manifold improves denoising accuracy and effectively restores the underlying manifold structure. The post-processing stage employs the Local Outlier Factor (LOF) within the embedded low-dimensional manifold space to identify and mitigate outliers. For determining outlier thresholds, the quartile range method is utilized, offering a robust statistical approach that does not yield a normal distribution and thereby increasing the algorithm's adaptability.

The structure of the paper is as follows: Section II introduces the concept of patch manifolds and explains the construction of graph structures on high-dimensional manifolds. Section III presents a two-stage algorithm for manifold denoising and outlier detection. It describes the main steps in the pre-processing and post-processing stages. Section IV demonstrates the performance of the algorithm through experimental results on multiple datasets and provides a comparative analysis. Section V summarizes the results, discusses potential applications of the proposed algorithm.

II. Preliminaries

A. Patch Manifold

In the context of processing raw data containing noise, the dataset $\mathbf{X}$ is defined for N points in a D-dimensional space as: $\mathbf{X} = \{\mathbf{x}_i \mid i = 1, 2, \cdots, N\}, \mathbf{x}_i \in \mathbb{R}^D$.

The noisy data can be simply modeled by the equation:

$$\mathbf{X} = \mathbf{U} + \mathbf{E}, \tag{1}$$

where $\mathbf{U}$ is the matrix of clean data, and $\mathbf{E}$ is a noise matrix.

In the dataset, patches are initially defined by selecting a subset of M points from $\mathbf{X}$ to serve as the patch centers, denoted by $\{\mathbf{c}_m\}_{m=1}^{M} \subseteq \mathbf{X}$. A patch $\mathbf{P}_m$ is centered at $\mathbf{c}_m$ and formed by selecting its k-nearest neighbors based on Euclidean distance. These patches collectively cover the entire dataset, i.e., $\bigcup_{m=1}^{M} \mathbf{P}_m = \mathbf{X}$, and each patch $\mathbf{P}_m$ is represented in $\mathbb{R}^{D\times k}$.

It is assumed that these patches represent samples from a low-dimensional smooth manifold embedded in $\mathbb{R}^{D\times k}$, referred to as the patch manifold $M(\mathbf{U})$ of the dataset $\mathbf{U}$. To evaluate the similarity between patches, coordinates of $\mathbf{P}_m$ are translated relative to $\mathbf{c}_m$, positioning $\mathbf{c}_m$ at the origin of the D-dimensional space. resulting in the adjusted coordinates:

$$\mathbf{P}'_m = \mathbf{P}_m - \mathbf{c}_m \mathbf{1}^\top, \tag{2}$$

where $\mathbf{P}'_m$ represents the adjusted coordinates of patch m, $\mathbf{1}$ is a k-dimensional column vector of ones, and $\mathbf{1}^\top$ denotes the transpose of $\mathbf{1}$.

B. Computation of Patch Distances with AIRM

To accurately compute the distance between two patches in high-dimensional space, the Affine Invariant Riemannian Metric (AIRM) is employed, leveraging the comparison of covariance matrices. Let $\mathbf{P}_m$ and $\mathbf{P}_n$ represent two patches, each containing k data points, denoted as $\mathbf{P}_m = \{\mathbf{x}_i^m\}_{i=1}^{k}$ and $\mathbf{P}_n = \{\mathbf{x}_i^n\}_{i=1}^{k}$, where $\mathbf{x}_i^m, \mathbf{x}_i^n \in \mathbb{R}^D$. The objective is to compute a distance d_{AIRM} that reflects the similarity between these two patches while accounting for their intrinsic manifold structure.

The process begins with calculating the sample covariance matrix for each patch. The sample mean vector for patch $\mathbf{P}_m$ is given by:

$$\boldsymbol{\mu}_m = \frac{1}{k} \sum_{i=1}^{k} \mathbf{x}_i^m, \tag{3}$$

and the sample covariance matrix for $\mathbf{P}_m$ can be calculated as follows:

$$\mathbf{\Sigma}_m = \frac{1}{k-1}\sum_{i=1}^{k}(\mathbf{x}_i^m - \boldsymbol{\mu}_m)(\mathbf{x}_i^m - \boldsymbol{\mu}_m)^\top, \quad (4)$$

and the covariance matrix for patch $\mathbf{P}_n$ is calculated similarly.

The computation of the covariance matrix for each patch is influenced by the bandwidth ϵ, which defines the local neighborhood of the patch. For a given patch $\mathbf{P}_m$ with center point $\mathbf{c}_m$, the local neighborhood $\mathcal{N}_m$ is defined as:

$$\mathcal{N}_m = \{\mathbf{x}_i \in \mathbf{X} \mid d(\mathbf{c}_m, \mathbf{x}_i) \leq \epsilon\},$$

where $d(\mathbf{c}_m, \mathbf{x}_i)$ denotes the distance metric, typically the Euclidean distance. The bandwidth ϵ controls the scale of the local neighborhood and directly affects the accuracy of the covariance matrix $\mathbf{\Sigma}_m$, thereby impacting subsequent denoising operations.

These covariance matrices reflect the variance and correlation structure of the data points within each patch, providing a basis for comparing the two patches. The AIRM is utilized to measure the distance between covariance matrix:

$$d_{\text{AIRM}}(\mathbf{\Sigma}_m, \mathbf{\Sigma}_n) = \left\| \log\left(\mathbf{\Sigma}_m^{-1/2}\mathbf{\Sigma}_n\mathbf{\Sigma}_m^{-1/2}\right)\right\|_F, \quad (5)$$

where $\log(\cdot)$ represents the matrix logarithm, and $\|\cdot\|_F$ is the Frobenius norm.

The AIRM measures the Riemannian distance on the manifold of covariance matrix. This distance is computed using the eigenvalues λ_i of $\mathbf{\Sigma}_m^{-1}\mathbf{\Sigma}_n$, obtained from the eigenvalue decomposition:

$$\mathbf{\Sigma}_m^{-1}\mathbf{\Sigma}_n\mathbf{v}_i = \lambda_i\mathbf{v}_i, \quad (6)$$

where λ_i and $\mathbf{v}_i$ are the i-th eigenvalue and eigenvector of $\mathbf{\Sigma}_m^{-1}\mathbf{\Sigma}_n$, respectively.

The distance is then given by:

$$d_{\text{AIRM}}(\mathbf{\Sigma}_m, \mathbf{\Sigma}_n) = \left(\sum_{i=1}^{D}\left[\ln\lambda_i\right]^2\right)^{1/2}. \quad (7)$$

C. Graph Construction

This distance is then incorporated into the graph construction, where the AIRM distance d_{AIRM} is used in the weight function for the graph edges. The weight between two patches $\mathbf{P}_m$ and $\mathbf{P}_n$ is defined as:

$$w_{mn} = \frac{\psi(d_{\text{AIRM}})}{(\rho_m\rho_n)^\alpha}, \quad (8)$$

where ρ_m and ρ_n represent the unnormalized degrees of the vertices corresponding to patches $\mathbf{P}_m$ and $\mathbf{P}_n$, respectively, and α is a parameter controlling the degree of normalization, typically selected within the range [0,1].

The unnormalized degree ρ_n for a patch is given by:

$$\rho_n = \sum_{m=1}^{M}\psi(d_{\text{AIRM}}), \quad (9)$$

where represents the local density of patches around $\mathbf{P}_n$.

The function $\psi(d_{\text{AIRM}})$ is a thresholded Gaussian function, defined as:

$$\psi(d_{\text{AIRM}}) = \begin{cases}\exp\left(-\frac{d_{\text{AIRM}}^2}{2\epsilon^2}\right), & \text{if } d_{\text{AIRM}} < r;\\ 0, & \text{otherwise,}\end{cases} \quad (10)$$

where ϵ is bandwidth parameter, and r is the neighborhood radius within which patches are considered connected. This weight function ensures that only patches within a certain distance r influence each other, thereby emphasizing locality in the graph construction.

III. Two-Stage Manifold Denoising and Outlier detection Algorithm

This section proposes a two-stage manifold denoising framework. Geometric similarity constraints and manifold regularization effectively address geometric differences between patches. The original high-dimensional data is divided into patches to capture local geometric structures, and optimization is performed using Riemannian gradient descent on the symmetric positive definite (SPD) manifold to achieve denoising. The post-processing stage further addresses the impact of outliers by adjusting the manifold structure to identify and mitigate their influence, ensuring optimal denoising results. The detailed implementation of each stage is provided below. The two-stage framework is illustrated in Fig. 1.

A. The Pre-Processing Stage: denoising

In the patch manifold denoising model, the dataset is decomposed into a series of patches that cover the entire original dataset. These patches may exhibit significant geometric differences and can be sampled from different manifolds of changing dimensions. This local processing effectively captures the underlying geometric structures of high-dimensional data, providing a foundation for noise reduction while preserving the inherent geometry. Each patch is treated as a local data block, facilitating the extraction of structural information from the high-dimensional space.

By leveraging the geometric similarities between patches, the model reduces the impact of noise while preserving local structural consistency. Imposing constraints on the geometric differences between neighbourhood patches ensures alignment across local structures and minimizes distortions during the denoising process. This approach is particularly beneficial when different patches are sampled from distinct manifolds, as the similarity constraints ensure that the complex geometric structure of the data remains intact after denoising.

The objective function for this algorithm is formulated as follows:

$$\min_{\mathbf{U}} \quad \frac{1}{2}\sum_{i=1}^{N}\|\mathbf{u}_i - \mathbf{x}_i\|^2 + \frac{\lambda}{2}\sum_{m,n} w_{mn}d_{\text{AIRM}}^2(\mathbf{\Sigma}_m, \mathbf{\Sigma}_n), \quad (11)$$

where $\mathbf{u}_i$ represents the denoised data points, and $\mathbf{x}_i$ denotes the original noisy data points, and λ is the regularization parameter. $\mathbf{\Sigma}_m$ and $\mathbf{\Sigma}_n$ are the covariance matrices for patches p_m and p_n, respectively, describing the local

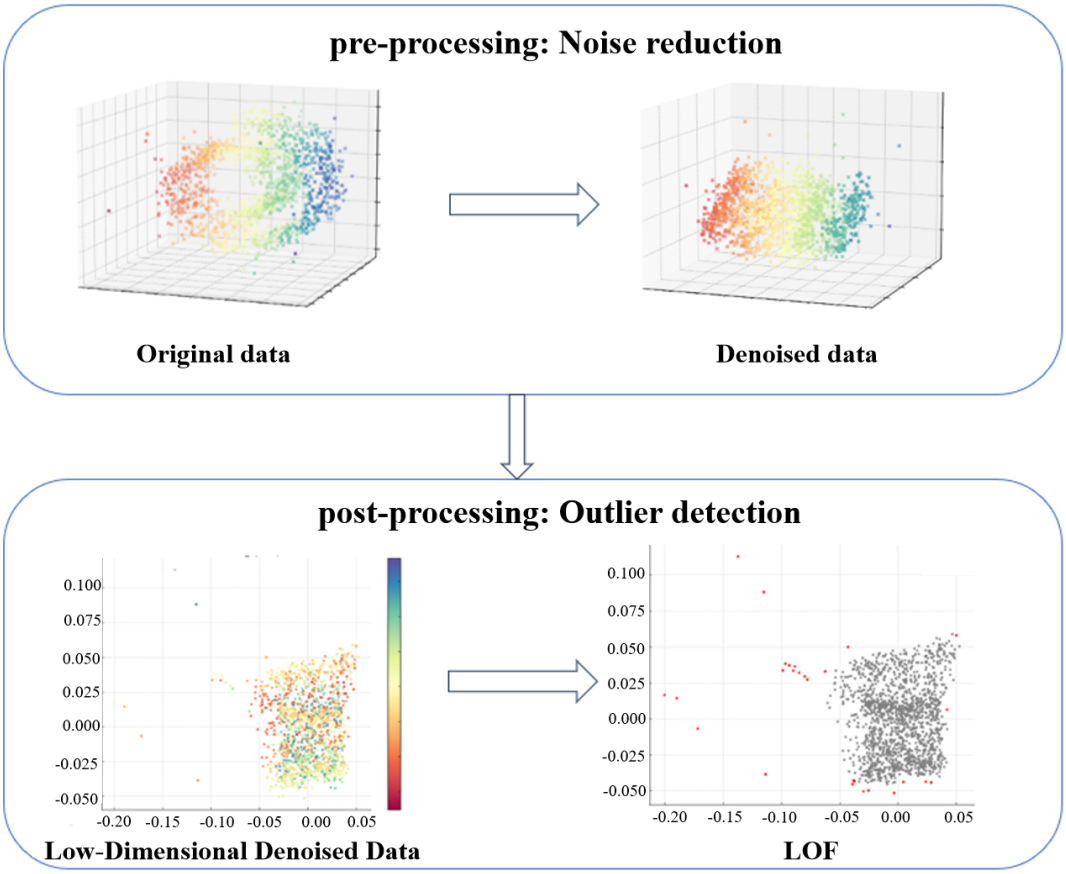


Fig. 1. Two-Stage manifold denoising and outlier detection algorithm framework

geometric structure of each patch. The AIRM distance $d_{\text{AIRM}}(\mathbf{\Sigma}_m, \mathbf{\Sigma}_n)$ is used to measure the geometric differences between these covariance matrices, and w_{mn} denotes the similarity weight between patches.

In this problem, the variable $\mathbf{U}$ is an element of the Euclidean space $\mathbb{R}^D$, whereas the covariance matrix $\mathbf{\Sigma}_m$ belongs to the SPD manifold. Consequently, computing the Riemannian gradient of the objective function with respect to $\mathbf{U}$ is performed on the SPD manifold.

Riemannian gradient descent operates on a curved manifold, accounting for its geometric structure rather than treating it as a flat Euclidean space. This method is essential for handling the structured covariance matrices $\mathbf{\Sigma}_m$, which belong to the SPD manifold.

For a function L defined on a manifold $\mathcal{M}$, the Riemannian gradient of L at a point $\mathbf{p} \in \mathcal{M}$ is denoted by $\text{grad}\, L(\mathbf{p})$. The Riemannian gradient satisfies the following condition:

$$\langle \text{grad}\, L(\mathbf{p}), \xi \rangle = DL[\xi], \quad \forall \xi \in T_p\mathcal{M}, \tag{12}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product defined on the tangent space $T_p\mathcal{M}$, $DL[\xi]$ is the directional derivative of L at $\mathbf{p}$ in the direction ξ, and $T_p\mathcal{M}$ is the tangent space at $\mathbf{p}$.

Based on (5), the Riemannian gradient of this distance metric with respect to $\mathbf{\Sigma}_m$ is expressed as:

$$\frac{\partial}{\partial \mathbf{\Sigma}_m}\left(\frac{1}{2} d^2_{\text{AIRM}}(\mathbf{\Sigma}_m, \mathbf{\Sigma}_n)\right) = -\mathbf{\Sigma}_m^{-1}\left(\log(\mathbf{\Sigma}_m^{-1}\mathbf{\Sigma}_n)\right)\mathbf{\Sigma}_m^{-1}. \tag{13}$$

The covariance matrix $\mathbf{\Sigma}_m$ is defined in terms of the patch data $\mathbf{u}_i^m$, and its gradient with respect to each $\mathbf{u}_i^m$ can be derived using the following:

$$\mathbf{\Sigma}_m = \frac{1}{k-1}\sum_{i=1}^{k}(\mathbf{u}_i^m - \boldsymbol{\mu}_m)(\mathbf{u}_i^m - \boldsymbol{\mu}_m)^\top. \tag{14}$$

To compute the gradient of $\mathbf{\Sigma}_m$ with respect to $\mathbf{u}_i^m$, the chain rule is applied, The gradient of the mean vector $\boldsymbol{\mu}_m$ with respect to $\mathbf{u}_i^m$ is:

$$\frac{\partial \boldsymbol{\mu}_m}{\partial \mathbf{u}_i^m} = \frac{1}{k}\mathbf{I}. \tag{15}$$

The gradient of the covariance matrix $\mathbf{\Sigma}_m$ with respect to $\mathbf{u}_i^m$ is:

$$\frac{\partial \mathbf{\Sigma}_m}{\partial \mathbf{u}_i^m} = \frac{1}{k}\mathbf{I}\left[(\mathbf{u}_i^m - \boldsymbol{\mu}_m) + (\mathbf{u}_i^m - \boldsymbol{\mu}_m)^\top\right]. \tag{16}$$

According to the chain rule, the gradient of $\mathbf{\Sigma}_m$ with respect to $\mathbf{u}_i^m$ can be calculated by considering both the mean $\boldsymbol{\mu}_m$ and the deviation terms. Since the gradient formulations for patch m and n are symmetric, the derivation for patch n follows the same steps as for patch m. Therefore, only the gradient for patch m is presented here. The gradient of the objective function with respect to $\mathbf{u}_i$ is finally expressed as:

$$\frac{\partial L}{\partial \mathbf{u}_i} = (\mathbf{u}_i - \mathbf{x}_i) + \lambda \sum_{m=1}^{M} \sum_{n} w_{mn} \left[\frac{\partial}{\partial \boldsymbol{\Sigma}_m} d^2_{\text{AIRM}}(\boldsymbol{\Sigma}_m, \boldsymbol{\Sigma}_n) \cdot \frac{\partial \boldsymbol{\Sigma}_m}{\partial \mathbf{u}_i} + \frac{\partial}{\partial \boldsymbol{\Sigma}_n} d^2_{\text{AIRM}}(\boldsymbol{\Sigma}_m, \boldsymbol{\Sigma}_n) \cdot \frac{\partial \boldsymbol{\Sigma}_n}{\partial \mathbf{u}_i} \right]. \tag{17}$$

After computing the Riemannian gradient of L with respect to $\mathbf{U}$, the variables are updated iteratively using the gradient descent rule:

$$\mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} - \eta \, \text{grad}\, L(\mathbf{u}_i^{(t)}), \tag{18}$$

where η is the step size, and $\text{grad}\, L(\mathbf{u}_i^{(t)})$ is the gradient of L at iteration t,

This iterative process continues until convergence, ensuring that the clean data $\mathbf{U}$ is recovered through iterative denoising.

B. The Post-processing Stage: outlier detection

While denoising effectively removes random errors and minor disturbances, the presence of outliers may still significantly affect the accuracy and robustness of subsequent analysis. Outliers differ from noise in that they represent pronounced deviations, often caused by special circumstances or extreme events in the data generation process. Therefore, outlier detection is a crucial post-denoising step to ensure the quality and reliability of the data.

In high-dimensional data, the "curse of dimensionality" exacerbates the complexity of outlier detection. Traditional outlier detection methods often fail in such environments due to distance or density. To address this, manifold learning assumes that the data lie on a lower-dimensional manifold M_H, and maps the data to a low-dimensional space while preserving their local geometric structure. Depending on the characteristics of the data distribution, different manifold dimensionality reduction methods, such as LLE, T-SNE, or UMAP, can be employed to effectively identify outliers in the embedded space.

Subsequently, local outlier factor (LOF) is employed to identify and eliminate outliers based on the low-dimensional manifold structure of the denoised data. This step calculates each point's local reachability density and outlier factor, not only reducing noise interference but also improving the accuracy and generalization of data analysis. The LOF provides a quantitative measure for identifying outliers by comparing each data point's local reachability density against those of its neighbors. A point is considered an outlier if its density is significantly lower than that of its neighbors. Mathematically, LOF is calculated based on the ratio of these local densities, with higher values indicating potential outliers.

The k-distance of a point p, denoted as k-distance(p) , measures the relative sparsity of an object within its neighborhood. It is defined by the distance $d(p, o)$ that satisfies:

- At least k points exist such that $d(p, o') \leq d(p, o)$,
- At most k-1 points exist such that $d(p, o') < d(p, o)$.

where, o denotes the "critical" point used to determine the k-distance of p, o' is a general notation referring to all data points compared with p in terms of distance.

The k-distance neighborhood of a point p includes all points whose distance to p does not exceed the k-distance. These points are termed p's k-nearest neighbors. Mathematically, it is defined as:

$$N_{k\text{-dist}}(p) = \{q \in \mathbf{D} | d(p, q) \leq k\text{-distance}(p)\}, \tag{19}$$

where $\mathbf{D}$ represents the dataset.

The reachability distance between two points p and o is given by:

$$\text{reach-dist}(p, o) = \max(\text{k-distance}(o), d(p, o)) \tag{20}$$

The local reachability density of p is calculated as:

$$\text{lrd}(p) = \frac{1}{\frac{\sum_{o \in N_k(p)} \text{reach-dist}(p,o)}{|N_k(p)|}}, \tag{21}$$

where $N_k(p)$ denotes the k-nearest neighborhood of p.

The local outlier factor of p is determined by comparing the local reachability density of p to those of its neighbors:

$$\text{LOF}(p) = \frac{\sum_{o \in N_k(p)} \frac{\text{lrd}(o)}{\text{lrd}(p)}}{|N_k(p)|}. \tag{22}$$

Calculate the LOF values for all points in the dataset. A point P is considered an outlier if its LOF is significantly greater than 1, indicating that the density around P is much lower compared to its neighbors:

$$\text{Outlier if LOF}(P) > 1. \tag{23}$$

Outliers can be generally classified into distributional outliers and structural outliers, based on their relative positions on the manifold. Given a high-dimensional observation space H and a low-dimensional parameter space $\Theta \subset \mathbb{R}^d$, the data are generated from a probability distribution P and mapped onto a high-dimensional manifold $M_H \subset H$. This mapping is performed via an isometric mapping $\phi : \Theta \to M_H$, while an embedding space $Y \subset \mathbb{R}^{d'}$ and an embedding map $e : M_H \to Y$ preserves the structure of M_H in the low-dimensional space. Based on this framework, the types of outliers can be precisely defined:

Let $\phi : \Theta \to M_c \subset \mathbb{R}^d$ be a mapping from the parameter space Θ to the main data manifold M_c, the main data manifold $M_c = M_{\Theta_c, \phi_c}$ and an alternative manifold $M_a = M_{\Theta_a, \phi_a}$.

For a dataset $X \subset M_a \cup M_c$, and assuming the structural outlier ratio r is given by:

$$r = \frac{|\{x_i \in X \mid x_i \in M_a \wedge x_i \notin M_c\}|}{|\{x_i \in X \mid x_i \in M_c\}|} \ll 1,$$

then an observation $x_i \in X$ is classified as:

- If $x_i \in M_a$ and $x_i \notin M_c$, then x_i is a structural outlier.
- If $x_i \in M_c$ is a distributional outlier, where $\theta_i \in \Theta_c$ and $x_i = \phi_c(\theta_i)$, Ω^*_α represents the minimum volume set generated by the probability distribution, containing at least a probability mass of α.

This outlier detection framework utilizes the low-dimensional manifold structure of high-dimensional data. The framework is built on two core assumptions: first, the manifold assumption, which posits that

observed high-dimensional data is distributed across a low-dimensional manifold; second, it assumes outliers are structural anomalies from different data generating processes, or distributional anomalies that are structurally similar to the main data but still significantly different in some respects. These mappings and the LOF scoring facilitate a robust approach to identifying and analyzing outliers in high-dimensional data by leveraging the underlying manifold structure.

IV. Experimental Results and Analysis

A. Datasets

To evaluate the performance of the proposed algorithm, 8 synthetic datasets and 6 real-world datasets were selected for the experiments. A summary of the datasets, including sample sizes and dimensions, is provided in Table 1.

The synthetic datasets comprise Swiss Roll, S-curve, Helix, Torus, Spiral, and Sphere. From the Swiss Roll and S-curve datasets, 1,000 sample points were selected, each with added Gaussian noise (standard deviation $\sigma = 0.1$) and 50 outliers. The Helix dataset consists of 1,000 sample points generated by uniformly sampling the parameter $t \in [0, 4\pi]$, with Gaussian noise ($\sigma = 0.1$) and 50 outliers. The Torus dataset contains 500 sample points generated using a parametric equation with a major radius of $R = 10$ and a minor radius of $r = 2$, with Gaussian noise ($\sigma = 0.1$) and 50 outliers. The Spiral dataset includes 1,000 sample points generated by uniformly sampling $angle \in [0, 4\pi]$ and $height \in [-1, 1]$, with Gaussian noise ($\sigma = 0.1$) and 50 outliers. The Sphere dataset comprises 500 sample points generated by uniformly sampling parameters $\phi \in [0, \pi]$ and $\theta \in [0, 2\pi]$, with Gaussian noise ($\sigma = 0.1$) and 50 outliers. Additionally, the MNIST dataset consists of 70,000 handwritten digit images (28×28 pixels) divided into 10 classes (digits 0 to 9) with added Gaussian noise ($\sigma = 0.4$). The Fashion-MNIST dataset contains 70,000 clothing item images (28×28 pixels) categorized into 10 classes (e.g., T-shirts, shoes), also with Gaussian noise ($\sigma = 0.4$).

The real-world datasets include ECG Heartbeat Categorization, Libras Movement, FordA, Iris, Adult Income, and Banknote Authentication. The ECG dataset contains electrocardiogram signals, with each sample comprising 187 features representing time-series signal amplitudes. The Libras Movement dataset consists of hand gesture motion data, with each sample containing 15 features representing time-series points of hand movement trajectories. The FordA dataset includes time-series sensor data from Ford vehicles, with 500 features representing sensor signals for each sample. The Iris dataset contains 4 features (sepal length, sepal width, petal length, and petal width) divided into 3 classes corresponding to different iris species. The Adult Income dataset contains census data, with each sample comprising 14 features such as age, occupation, education level, and marital status. Finally, the Banknote Authentication dataset includes features extracted from banknotes, with each sample consisting of 4 features representing wavelet transform coefficients, variance, skewness, and entropy.

The following evaluation metrics were employed in the experiments:

• **Mean Squared Error (MSE)** MSE is used to measure the difference between the data processed by the proposed algorithm and the clean, ground-truth data. It evaluates the algorithm's ability to restore the original data structure by simultaneously removing noise and outliers. Smaller MSE values indicate higher reconstruction accuracy and better preservation of the underlying data characteristics.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2, \tag{24}$$

where x_i represents the ground-truth data, $\hat{x}_i$ is the processed data, and N is the total number of data points.

• **Area Under the Curve (AUC)** AUC quantifies the overall performance of outlier detection, with values ranging from 0 to 1. Higher values indicate better discrimination between normal samples and outliers. AUC was used to validate the effectiveness of the two-stage approach across multiple datasets.

$$\text{AUC} = \int_0^1 TPR(FPR)\,\mathrm{d}FPR, \tag{25}$$

where TPR is the True Positive Rate and FPR is the False Positive Rate.

• **Geodesic Distance Neighborhood Retention Rate (GDNRR)** GDNRR reflects the preservation of manifold structure after denoising and is defined as:

$$\text{GDNRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathcal{N}_i^{\text{true}} \cap \mathcal{N}_i^{\text{denoised}}|}{k}, \tag{26}$$

where $\mathcal{N}_i^{\text{true}}$ and $\mathcal{N}_i^{\text{denoised}}$ represent the k-nearest neighbors of point i in the original and denoised data, respectively. A higher GDNRR indicates better preservation of the local geometry.

• **Signal-to-Noise Ratio (SNR)** SNR evaluates the effectiveness of outlier detection and is defined as:

$$SNR = 20\lg\left(\frac{R_I}{R_O}\right), \tag{27}$$

where R_I represents the number of retained inliers, and R_O denotes the number of undetected outliers. Higher SNR values indicate better detection accuracy and retention of normal samples.

• **Manifold Score** Manifold score evaluates the alignment of denoised data with the ideal manifold structure, ranging from 0 to 1. Higher scores indicate better global geometric consistency.

B. Experimental Results

1) Synthetic Dataset: Experiments were conducted on synthetic datasets with clear geometric structures. These datasets clearly show the impact of noise and outliers on manifold distributions and demonstrate the algorithm's effectiveness.

As shown in Fig.2, the first row of results illustrates the disruption of the original manifold structure caused by noise and outliers. Specifically, noise introduces local distortions on the manifold surface, while outliers appear as isolated points distant from the main manifold. For instance, in the Swiss Roll dataset, noise interferes with the smoothness of the manifold, and outliers compromise the overall spiral shape. In the Torus dataset, noise and outliers obscure the originally symmetric geometric properties. For the Helix dataset, noise blurs local structures, and outliers

TABLE I

SUMMARY OF SYNTHETIC AND REAL-WORLD DATASETS.

Type	Dataset	Samples	Dimensions
Synthetic Dataset	Swiss Roll	1,000	3
	S-curve	1,000	3
	Helix	1,000	3
	Torus	500	3
	Spiral	1,000	3
	Sphere	500	3
	MNIST	70,000	784
	Fashion-MNIST	70,000	784
Real-world Dataset	ECG Heartbeat Categorization	5,000	187
	Libras Movement	360	15
	FordA	3,600	500
	Iris	150	4
	Adult Income	48,842	14
	Banknote Authentication	1,372	4

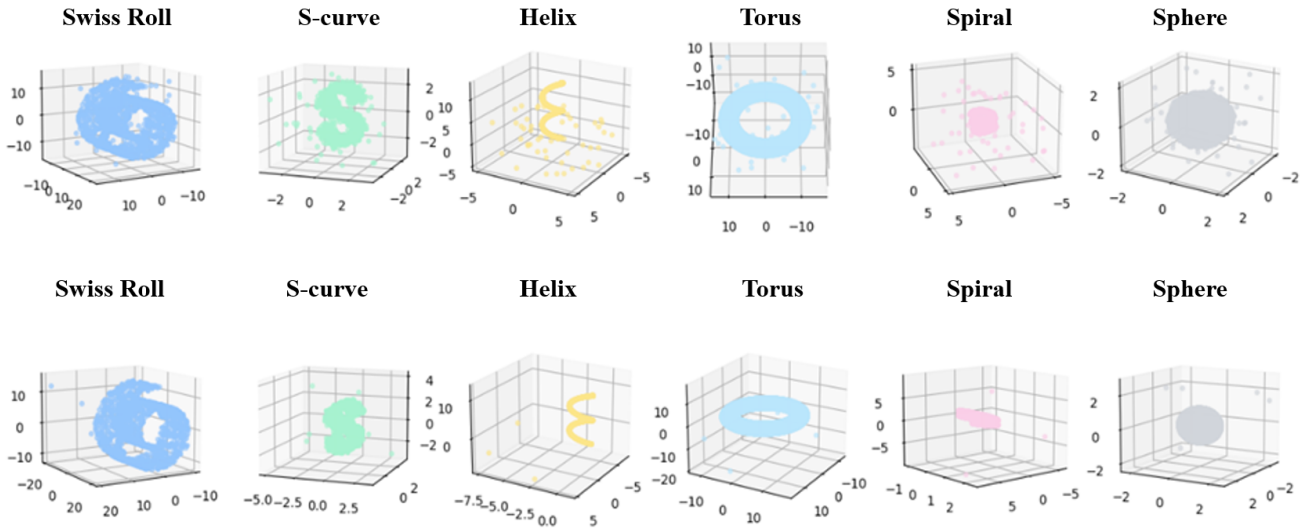


Fig. 2. Visualization of the dataset before and after processing with the proposed algorithm

disrupt the continuity of the helical manifold. Similarly, for the S-curve and Spiral datasets, noise and outliers jointly damage the overall distribution of low-dimensional and nonlinear manifolds.

The second-row results illustrate the data distribution after algorithm processing. It can be observed that the geometric structure of the manifolds is restored, and most noise and outliers are successfully removed. In the Swiss Roll and S-curve datasets, the smoothness and overall shapes of the manifolds are significantly improved. In the Helix and Spiral datasets, local consistency and continuity are restored. For the Torus and Sphere datasets, the algorithm not only removes outliers but also effectively preserves the global symmetry of the original manifolds. These results demonstrate that the proposed algorithm is adaptable to various manifold shapes and exhibits high robustness and applicability across diverse scenarios.

To further evaluate the specific effects of each stage, experiments were conducted using the MNIST and Fashion-MNIST datasets.

The results of the first stage, shown in Fig.3, demonstrate the denoising process. The original dataset contains clear images, but after adding Gaussian noise, image details significantly degrade, and noise severely impacts visual quality. After denoising, the image quality is effectively restored, preserving important details.

In Fig. 4, we see the results of the second stage. In the t-SNE visualization, normal samples form compact clusters, while outliers are well-separated from the main clusters. Using the LOF method, outliers are successfully detected and marked as red points. These results highlight the proposed method's ability to reconstruct low-dimensional manifold structures in high-noise environments and accurately detect outliers.

As show in Fig.5 and Fig.6, the proposed algorithm results the Fashion-MNIST dataset. The denoising stage restores image features and visual quality, while the outlier detection stage effectively separates normal samples from outliers, further validating the effectiveness and robustness of the proposed method.

2) Real-world Dataset: To further verify the effectiveness of the denoising and outlier detection steps, the ROC-AUC metric was used to evaluate the performance under conditions of high noise and outliers. Specifically, the LOF was employed to perform outlier detection on the dataset by ranking outliers. During this process, different outlier ratios were considered ($r \in \{0.01, 0.025, 0.05, 0.1\}$). Each setting r was repeated 50 times with random outlier sampling, and ROC-AUC was used as the final evaluation metric.

Fig.7 shows the visualization of three typical datasets: ECG Heartbeat, Libras Movement, and FordA. The normal samples are represented in gray, while the outlier samples

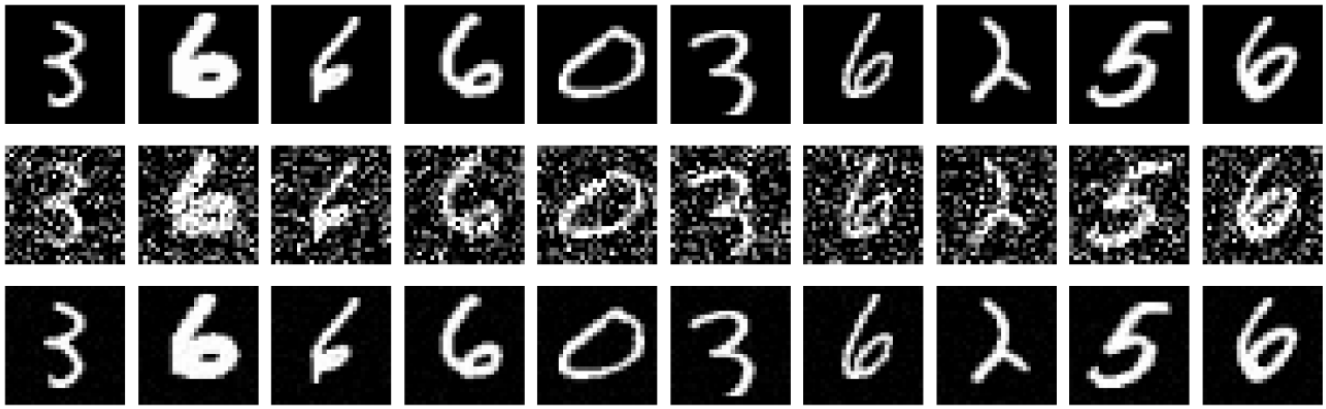


Fig. 3. MNIST sample images before and after denoising

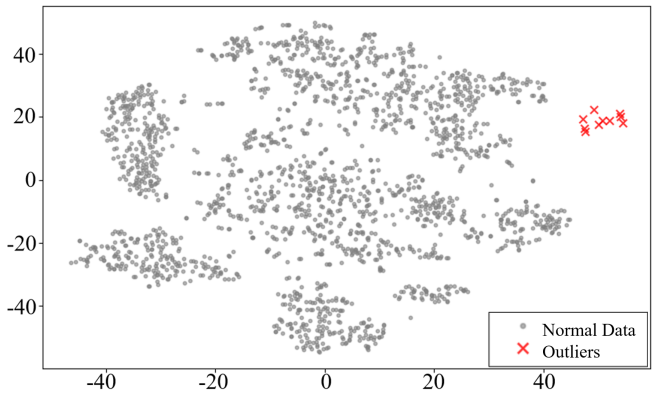


Fig. 4. The t-SNE visualization of the MNIST dataset after denoising. Gray points represent normal samples, red crosses indicate the detected outliers

are represented in red. From (a) ECG Heartbeat to (b) Libras Movement, and to (c) FordA datasets, a trend of normal samples concentrating within the distribution can be observed, while outlier samples show significant deviation. This deviation serves as the foundation for subsequent LOF-based outlier detection.

The manifold denoising in the preprocessing stage effectively reduced noise in the dataset, making the true outliers more prominent within the data structure. The difference between outliers and normal data points was thereby enhanced. Based on this, the LOF detected outliers more accurately by comparing the local density of each point with that of its neighbors. The experimental results showed that, after manifold denoising, the low-dimensional structure of the dataset was well preserved, and the added outliers were easier to identify due to their significant deviation from the data manifold.

Table II shows, for all datasets, the two-stage approach of manifold denoising followed by LOF resulted in high ROC-AUC values (mostly above 0.95). This indicates that manifold denoising effectively improved the performance of LOF in outlier detection by removing noise from the dataset. For example, in the ECG Heartbeat Categorization and FordA datasets, despite the complexity of time series features and inherent noise, manifold denoising allowed LOF to maintain high detection accuracy across various outlier ratios. For the ECG Heartbeat dataset, the AUC value reached 0.95 when $r = 0.01$, demonstrating high detection

TABLE II

THE ROC-AUC VALUES OF DIFFERENT DATASETS.

Dataset	k	r: 0.01	r: 0.025	r: 0.05	r: 0.1
ECG	0.01n	0.75	0.72	0.70	0.58
	0.1n	0.95	0.97	0.95	0.90
	0.75n	0.85	0.85	0.88	0.85
	0.9n	0.70	0.70	0.71	0.70
Libras	0.01n	0.85	0.62	0.67	0.67
	0.1n	0.98	1.00	0.98	0.77
	0.75n	0.90	0.90	0.90	0.90
	0.9n	0.80	0.80	0.81	0.99
FordA	0.01n	0.70	0.62	0.58	0.66
	0.1n	0.90	0.92	0.92	0.90
	0.75n	0.95	0.95	0.92	0.95
	0.9n	0.92	0.92	0.92	0.89
Iris	0.01n	0.98	0.75	0.60	0.59
	0.1n	0.99	1.00	1.00	0.70
	0.75n	1.00	1.00	1.00	1.00
	0.9n	0.99	1.00	1.00	1.00
Adult Income	0.01n	0.70	0.62	0.62	0.60
	0.1n	0.95	0.97	0.97	0.95
	0.75n	0.92	0.93	0.93	0.92
	0.9n	0.90	0.91	0.91	0.92
Banknote	0.01n	0.70	0.60	0.58	0.59
	0.1n	0.95	0.95	0.95	0.95
	0.75n	0.98	1.00	1.00	1.00
	0.9n	0.97	1.00	1.00	0.97

performance. In the FordA dataset, AUC values remained stable (around 0.9) as the neighborhood size k varied (from $0.1n$ to $0.9n$), indicating robustness of the algorithm.

For the Libras Movement and Iris datasets, the results demonstrated that manifold denoising effectively supported LOF in detecting outliers, even in low-dimensional datasets. In particular, the AUC value for Libras Movement was 0.85 when $r = 0.01$, which highlighted the effectiveness of the preprocessing step, even with a small neighborhood size (e.g., $0.01n$).

In datasets with complex features, such as Adult Income and Banknote Authentication, the results showed that the LOF exhibited a certain degree of robustness to changes in neighborhood size and outlier ratio. Specifically, at a small outlier ratio ($r = 0.01$), detection performance was not sensitive to parameter variations. For most datasets, when using an appropriate neighborhood size (e.g., $0.75n$), LOF maintained good detection performance even at higher outlier ratios (e.g., $r = 0.1$).

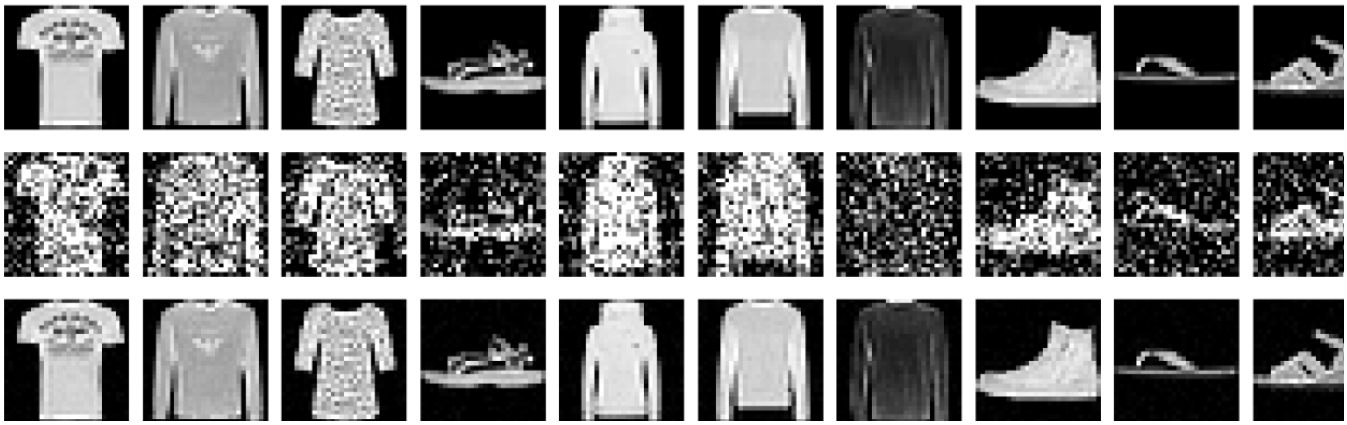


Fig. 5. Fashion-MNIST sample images before and after denoising

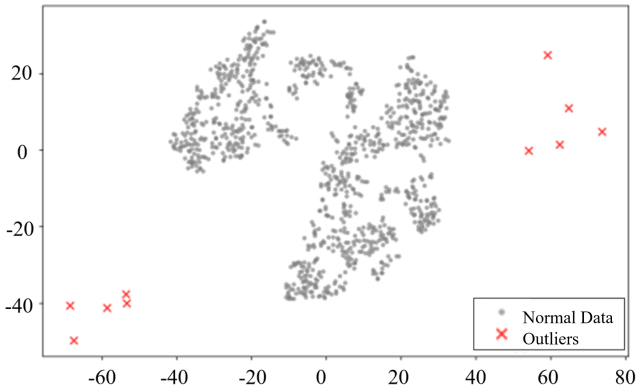
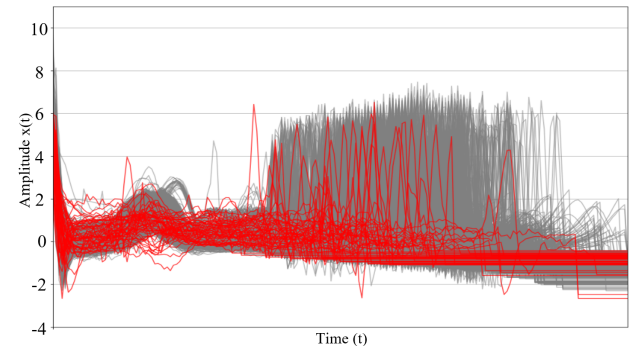
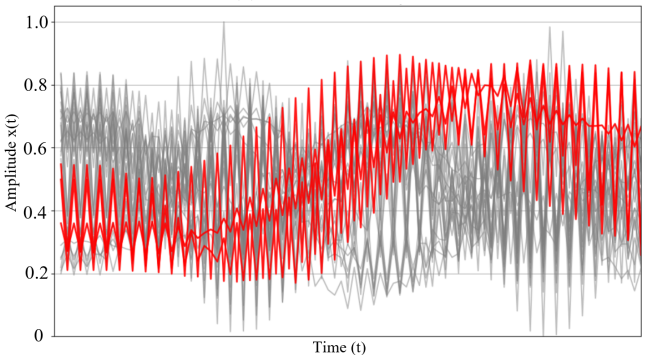


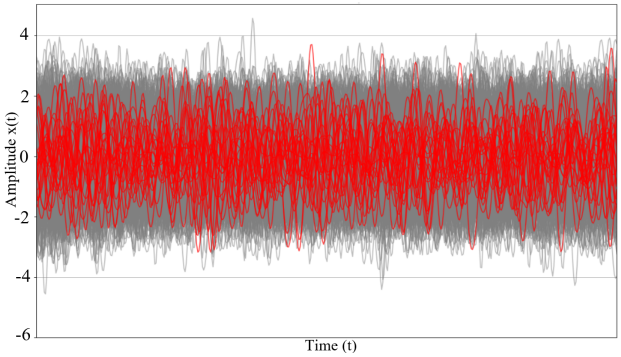
Fig. 6. The t-SNE visualization of the Fashion-MNIST dataset after denoising. Gray points represent normal samples, red crosses indicate the detected outliers



(a) ECG Heartbeat



(b) Libras Movement



(c) FordA

Fig. 7. Visualizations of the three datasets

C. *Effect of Parameter Settings*

To verify the effect of bandwidth selection on local geometric structure estimation, the distribution of eigenvalues of local covariance matrices under different bandwidth conditions was analyzed. Manifold denoising relies on the accurate estimation of local covariance matrices, and the bandwidth choice directly affects the estimation of local neighborhoods by covariance matrices. Therefore, analyzing the effect of bandwidth on the distribution of eigenvalues of covariance matrices can help understand how bandwidth controls the size of local neighborhoods, thereby influencing the estimation of local geometric features.

The experimental settings are as follows: the S-curve dataset contains 1000 sampled points, which are embedded in a 30-dimensional space. Local covariance matrices are computed using the 20 nearest neighbors of each center point. The experiment includes both noise-free data and data with added Gaussian noise (standard deviation $\sigma = 0.1$). The bandwidth parameter ϵ takes values of 0.15, 0.2, 0.25, and 0.3 for the noise-free data; for the data with noise, the values are 0.5, 1, 1.5, 2, 2.5, and 3.

1) Effect of Bandwidth ϵ: Fig.8 shows the eigenvalue distributions of local covariance matrices under different bandwidths for the S-curve in the noise-free case. The horizontal axis represents the index of eigenvalues i, ranging from 1 to 30, and the vertical axis represents the magnitude of the eigenvalues. Under all bandwidth conditions, the first two eigenvalues are significantly larger than the rest, and the remaining eigenvalues rapidly decrease to near zero from the third eigenvalue onward. This indicates that there are two principal directions in the local covariance matrix, consistent with the intrinsic geometry of the S-curve as a two-dimensional manifold. For smaller bandwidths (e.g.,

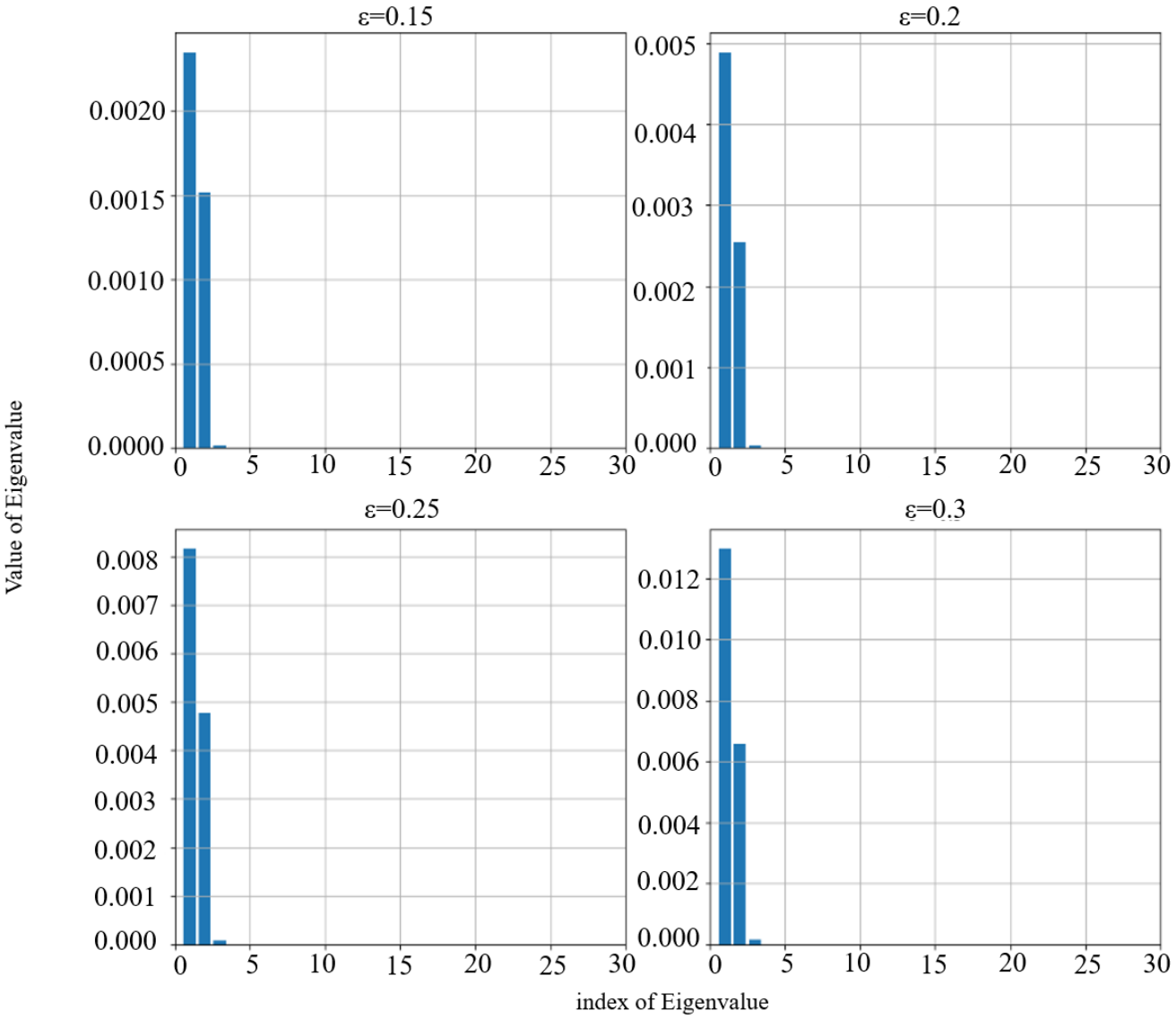


Fig. 8. Eigenvalue distribution of local covariance matrices under different bandwidths (noise-free case).

$\epsilon = 0.15, 0.2$), the eigenvalues of the local covariance matrix accurately reflect the local structure of the manifold, with the first two eigenvalues being significant and the remaining ones approaching zero. As the bandwidth increases (e.g., $\epsilon = 0.25, 0.3$), the first two eigenvalues increase, indicating that the increased bandwidth includes more neighborhood information, but the overall eigenvalue distribution still reflects the low-dimensional nature of the manifold.

Fig.9 shows the eigenvalue distribution with added Gaussian noise. For smaller bandwidths (e.g., $\epsilon = 0.5, 1$), the eigenvalue distribution still shows a trend where the first two eigenvalues are larger and the subsequent eigenvalues quickly approach zero, but the magnitude of the first two eigenvalues is larger than in the noise-free case, indicating that noise has affected the estimation of the local covariance matrices. As the bandwidth increases (e.g., $\epsilon = 1.5, 2, 2.5, 3$), the first few eigenvalues increase significantly. Especially when $\epsilon = 3$, the first five eigenvalues all have large values, indicating that the larger bandwidth introduces more distant neighborhood points, resulting in a significant impact of noise on the estimation of the local geometric structure by the covariance matrices.

2) Joint Analysis of Bandwidth ϵ and Neighborhood Distance Threshold r: On the Swiss roll dataset, the Pareto front is used to select the optimal bandwidth ϵ and neighborhood distance threshold r, aiming to achieve the best balance between reconstruction error (MSE) and geodesic distance neighborhood retention rate (GDNRR). The bandwidth ϵ determines the range of neighboring points included in the denoising process. A large ϵ may lead to an overly smooth denoising result that blurs the original geometric structure, while a small ϵ may result in insufficient denoising, leaving residual noise. The neighborhood distance threshold r controls the connectivity between points. A high r can lead to incorrect connections between unrelated points, while a low r may cause local structures to be disconnected. Selecting suitable values for ϵ and r enables effective noise suppression and accurate preservation of the manifold structure.

Gaussian noise with a standard deviation of 0.01 is added to the noise-free Swiss roll data. The bandwidth ϵ is set to values between $[0.1, 1.0]$ with a step size of 0.1, and the neighborhood distance threshold r ranges from $[0.5, 2.0]$ with a step size of 0.1. For each (ϵ, r) combination, the

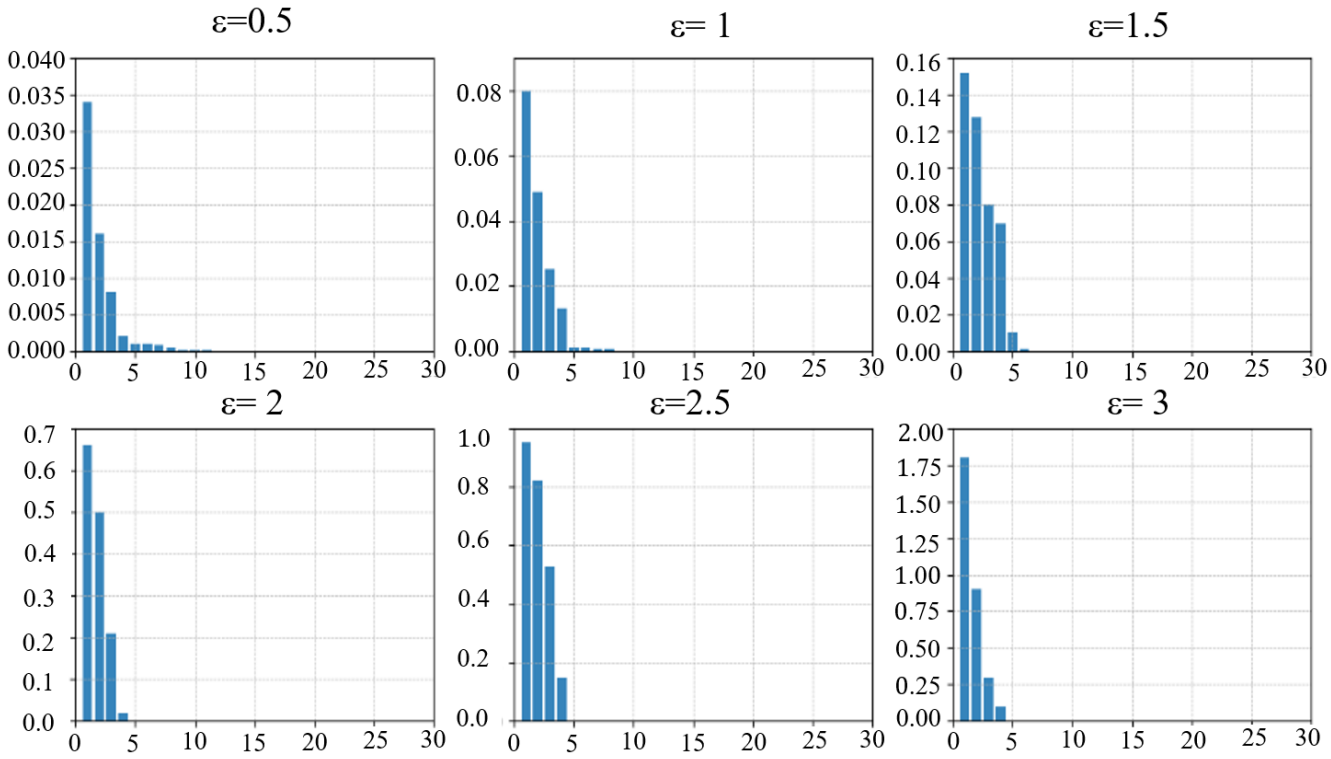


Fig. 9. Eigenvalue distribution of local covariance matrices under different bandwidths (gaussian noise $\sigma = 0.1$).

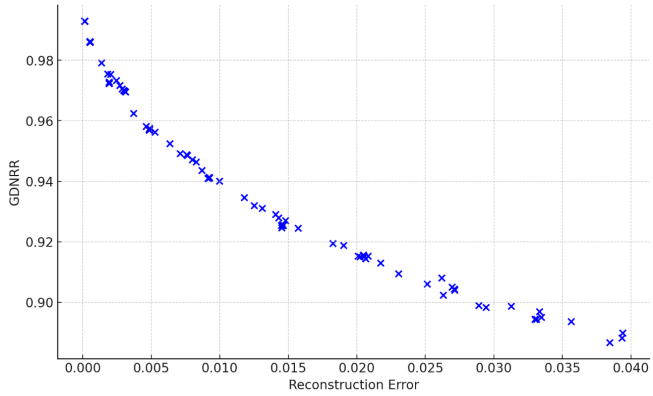


Fig. 10. Pareto Optimal ϵ and r combinations

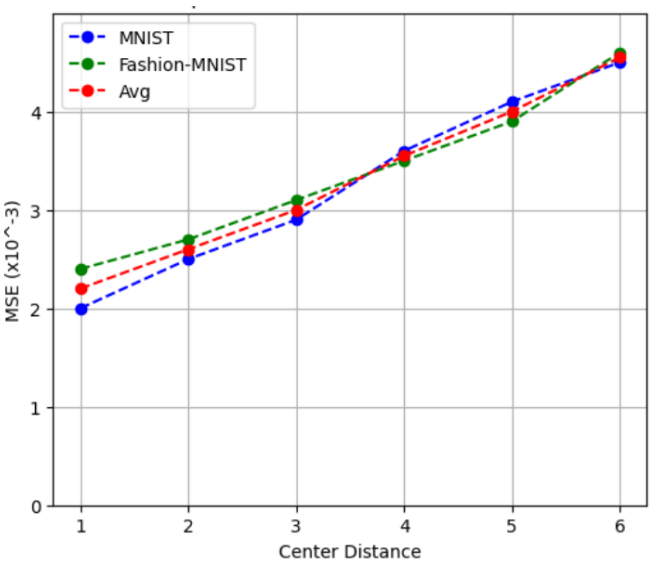


Fig. 11. Effect of two patch center distance difference on MSE for MNIST and Fashion-MNIST

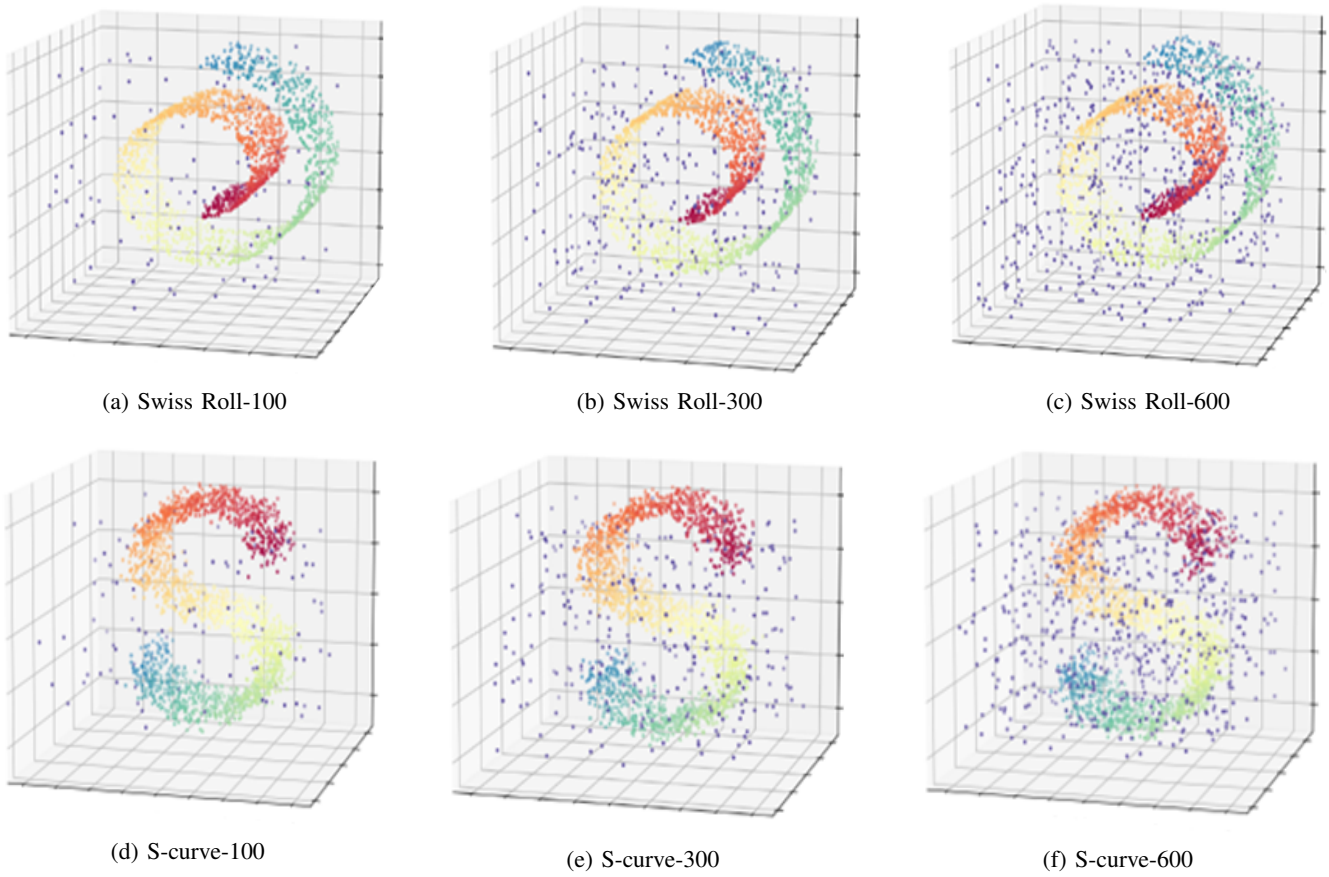
denoising algorithm is run, and both MSE and GDNRR are computed to evaluate the denoising precision and preservation of the manifold structure.

As shown in Fig.10, the horizontal axis represents MSE, and the vertical axis represents GDNRR. Each point represents the denoising performance for a specific (ϵ, r) combination. In the upper left region of the plot, the Pareto front appears, indicating optimal solutions that achieve a balance between MSE and GDNRR. Combinations with $\epsilon = 0.1$ and r between 1.0 and 2.0 are located on the Pareto front, demonstrating an effective balance between MSE (0.000132) and GDNRR (0.9929).

3) Impact of Patch Center Distance: Fig.11 shows the variation in Mean Squared Error (MSE) for the MNIST and Fashion-MNIST datasets under different patch center distances. The horizontal axis represents the distance between the centers of two patches, while the vertical axis shows the MSE, measured in 10^{-3}. The figure also includes an average MSE curve to assess the overall performance as the patch center distance changes. It can be observed that the MSE for both the MNIST and Fashion-MNIST datasets increases with the patch center distance. This trend is consistent for both datasets. The results indicate that as the distance between patch centers grows, the overlapping region between patches decreases. This reduction in overlap reduces the algorithm's ability to capture local image information, leading to higher reconstruction errors.

D. Ablation Experiment

To evaluate the algorithm's ability to preserve the global geometric structure during the manifold denoising stage, the manifold score is introduced as a performance metric. Unlike GDNRR, which validates local neighborhood preservation, the manifold score focuses on global consistency and



(a) Swiss Roll-100 (b) Swiss Roll-300 (c) Swiss Roll-600

(d) S-curve-100 (e) S-curve-300 (f) S-curve-600

Fig. 12. Swiss Roll and S-curve with 100, 300, 600 outliers.

measures the alignment of denoised data with the ideal manifold's overall geometric properties.

As shown in Table III, the manifold score improves consistently with increasing sample size, indicating that the algorithm effectively suppresses noise and restores the global geometric structure across datasets of varying scales.

Accurate recovery of the global geometric structure is critical for the subsequent outlier detection process. The combination of local structure preservation (validated by GDNRR) and global consistency enhances the algorithm's ability to identify structural anomalies while minimizing the influence of geometric distortions introduced during the denoising stage.

To verify the effectiveness of the proposed method in the outlier detection post-processing stage, experiments were conducted on the Swiss Roll and S-curve datasets. In these experiments, only post-processing was performed, and different numbers of outliers were directly added to a dataset containing a total of 2000 samples to evaluate the effectiveness of the post-processing method in removing outliers. Table IV shows the performance comparison between the proposed method and NRPCA on the Swiss Roll and S-curve datasets under different numbers of outliers, listing the comparison results for the number of identified inliers (RI), the number of correctly identified outliers (RO), and the signal-to-noise ratio (SNR).

Fig.12 (a)-(c) represent the Swiss Roll dataset with 100, 300, and 600 outliers, respectively, while Fig.12 (d)-(f) show the corresponding distribution for the S-curve dataset. It can be observed that, as the number of outliers increases, the distribution of outliers becomes more complex, the more outliers deviating from the original manifold structure.

TABLE III

MANIFOLD SCORES OF SIX DATASETS.

Dataset	200 Samples	500 Samples	1000 Samples
Swiss Roll	0.8975	0.9019	0.9571
S-curve	0.8512	0.8921	0.9315
Helix	0.8125	0.8732	0.9176
Torus	0.8301	0.8705	0.9107
Spiral	0.7890	0.8523	0.8981
Sphere	0.8954	0.9213	0.9478

TABLE IV

THE EXPERIMENTAL RESULTS COMPARING NRPCA AND THE PROPOSED ALGORITHM.

Dataset	Outliers	NRPCA			Ours		
		RI	RO	SNR	RI	RO	SNR
Swiss Roll	100	1837	29	36.03	1895	23	38.32
	300	1873	148	22.05	1859	104	25.04
	600	1815	406	13.01	1815	322	15.02
S-curve	100	1805	28	36.19	1886	23	38.28
	300	1860	147	22.04	1870	105	25.01
	600	1802	403	13.00	1896	337	15.00

Experimental results indicate that, for both the Swiss Roll and S-curve datasets, the performance trend of the proposed method is similar to that of NRPCA under different numbers of outliers. When the number of outliers is small (100), the performance of both methods is comparable, and the SNR difference is insignificant. However, as the

number of outliers increases, the proposed method gradually demonstrates a more pronounced advantage, particularly when the number of outliers reaches 600. This result shows that the proposed method is more effective in detecting outliers at a high outlier ratio, successfully removing outliers that deviate from the manifold structure while retaining a higher number of inliers.

To verify the necessity of the post-processing module (outlier detection), this experiment compares the manifold denoising only approach with the two-stage approach (manifold denoising + outlier detection) on the MNIST and Fashion-MNIST datasets. Fig. 13 shows that the two-stage approach significantly reduces MSE in datasets (from 2.0 to 1.5 for MNIST and from 2.4 to 1.8 for Fashion-MNIST). The post-processing with outlier detection is essential for improving image reconstruction quality.

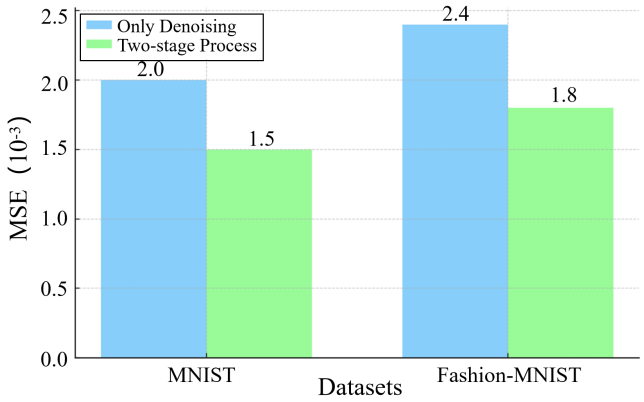


Fig. 13. MSE comparison for MNIST and Fashion-MNIST: only denoising vs two-stage process

V. Conclusions

This paper proposes a two-stage manifold denoising and anomaly detection algorithm that effectively reduces noise in high-dimensional data and accurately identifies anomalies. The proposed algorithm employs the AIRM and gradient descent on the SPD manifold to preserve the intrinsic geometric structure of the data. In the post-processing stage, the LOF is utilized to detect and remove anomalies. By leveraging a patch-based analysis, the proposed algorithm integrates local and global information, enhancing its robustness and applicability to complex data structures. Experimental results demonstrate the proposed algorithm's effectiveness in preserving data structures and improving denoising performance. Its effective performance across multiple quantitative metrics, further validated by visual results, highlights its robustness and reliability.

References

[1] Agnolucci Lorenzo, Galteri Leonardo, Bertini Marco and Del Bimbo Alberto, "Arniqa: Learning Distortion Manifold for Image Quality Assessment," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp189-198.

[2] Dehua Peng, Zhipeng Gui, Wenzhang Wei and Huayi Wu, "Scalable Manifold Learning by Uniform Landmark Sampling and Constrained Locally Linear Embedding," arXiv preprint arXiv:2401.01100, 2024.

[3] Shamsolmoali Pourya, Zareapoor Masoumeh, Zhou Huiyu, Tao Dacheng and Li Xuelong, "Vtae: Variational Transformer Autoencoder with Manifolds Learning," IEEE Transactions on Image Processing, vol. 32, pp4486-4500, 2023.

[4] Yonghe Chu, Heling Cao, Yufeng Diao and Hongfei Lin, "Refined SBERT: Representing Sentence BERT in Manifold Space," Neurocomputing, vol. 555, pp126453, 2023.

[5] Zare Masoumeh, Azizizadeh Najmeh and Kazemipour Ali, "Supervised Feature Selection on Gene Expression Microarray Datasets using Manifold Learning," Chemometrics and Intelligent Laboratory Systems, vol. 237, pp104828, 2023.

[6] Yan Huang, "Detecting Phase Transitions in Financial Market based on Manifold Learning," in Proceedings of the 2024 4th Asia Conference on Information Engineering (ACIE), IEEE, 2024, pp1-5.

[7] Roweis Sam T and Saul Lawrence K, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, vol. 290, no.5500, pp2323-2326, 2000.

[8] Tenenbaum Joshua B, Silva Vin de and Langford John C, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, vol. 290, no.5500, pp2319-2323, 2000.

[9] Belkin Mikhail and Niyogi Partha, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," Neural Computation, vol. 15, no.6, pp1373-1396, 2003.

[10] Zhenjian Yao, Zhongyu Wang, Xiaojun Liu, Chenchen Wang and Zhendong Shang, "An Improved Low-Frequency Noise Reduction Method in Shock Wave Pressure Measurement based on Mode Classification and Recursion Extraction," ISA Transactions, vol. 109, pp315-326, 2021.

[11] Park Chan Rok and Lee Youngjin, "Fast Non-local Means Noise Reduction Algorithm with Acceleration Function for Improvement of Image Quality in Gamma Camera System: a Phantom Study," Nuclear Engineering and Technology, vol. 51, no.3, pp719-722, 2019.

[12] Zhiping Lin, Zhao Kang, Lizong Zhang and Ling Tian, "Multi-view Attributed Graph Clustering," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no.2, pp1872-1880, 2023.

[13] Sober Barak, Aizenbud Yariv and Levin David, "Approximation of Functions over Manifolds: a Moving Least-Squares Approach," Journal of Computational and Applied Mathematics, vol. 383, pp113140, 2021.

[14] Lyu He, Ningyu Sha, Shuyang Qin, Ming Yan, Yuying Xie and Rongrong Wang, "Manifold Denoising by Nonlinear Robust Principal Component Analysis," in Advances in Neural Information Processing Systems, vol. 32, pp13390–13400, 2019.

[15] Qin Jiang, Yi Zhang, Wei Wang and Qian Huang, "Manifold Embedded Domain Adaptation for Motor Imagery EEG Decoding," IAENG International Journal of Computer Science, vol. 51, no.8, pp985–997, 2024.

[16] Yingying Meng, Qiaoyan Li, Xiaofei Yang and Xuezhen Dai, "Unsupervised Feature Selection Algorithm based on Laplace Rank Constraint and Local Structure Preservation," IAENG International Journal of Computer Science, vol. 51, no.12, pp1914-1923, 2024.

[17] Breunig Markus M, Kriegel Hans-Peter, Ng Raymond T and Sander J, "LOF: Identifying Density-based Local Outliers," Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, vol. 29, no.2, pp93-104, 2000.

[18] Liu Fei Tony, Kaiming Ting and Zhihua Zhou, "Isolation Forest," in 2008 Eighth IEEE International Conference on Data Mining, 2008, pp413-422.

[19] Rui Yang, Xiuqing Chen and Tianjie Cao, "APE-GAN++: An Improved APE-GAN to Eliminate Adversarial Perturbations," IAENG International Journal of Computer Science, vol. 48, pp827-844, 2021.