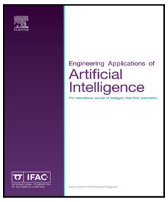
Engineering Applications of Artificial Intelligence 167 (2026) 113821

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Symmetric Positive Definite manifold deep metric learning for bearing fault diagnosis

Junshi Cheng [a], Ruisheng Ran [a,*], Bin Fang [b], Benchao Li [c,d]

[a] College of Computer and Information Science, Chongqing Normal University, Chongqing, 401331, China
[b] College of Computer Science, Chongqing University, Chongqing, 400044, China
[c] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, 611756, China
[d] Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, Chengdu, 611756, China

ARTICLE INFO

Keywords:
Fault diagnosis
Symmetric Positive Definite manifold
Deep metric learning
Denoising autoencoder

ABSTRACT

Rolling bearings are essential components in rotating machinery, and their failures can lead to unplanned downtime, substantial economic losses, and even severe safety risks. Consequently, bearing fault diagnosis has become a crucial task in modern industry, with machine learning methods playing a central role. However, many existing methods are designed in Euclidean space, limiting their ability to capture nonlinear features in bearing signals Additionally, they often have excessive parameters and irrelevant features, making it difficult to learn the correct data distribution. To address these challenges, this paper proposes a Symmetric Positive Definite (SPD) manifold deep metric learning method for bearing fault diagnosis, based on a supervised learning. This method transforms the original data into a SPD manifold, and constructs a SPD sparse denoising autoencoder for feature extraction. And then, a multi-class N-pair loss term on SPD manifold is used to improve classification ability. Comprehensive experiments have shown that this method has strong robustness to noise and low computational complexity. Due to the nonlinear expression ability of SPD manifolds, this method improves classification accuracy and is superior to existing methods in Euclidean space.

1. Introduction

In numerous industrial sectors, such as manufacturing, aerospace, and electronics, rotating machinery is widely used. Rolling bearings, as key components in such machinery, have a direct impact on operational efficiency. Bearing failures can interrupt normal operation, cause severe equipment damage and economic losses, and even lead to human casualties (Li et al., 2015). Studies have shown that 40%–50% of rotating machinery failures are directly caused by rolling bearing faults (Thorsen and Dalva, 2002), making timely identification of such faults essential. Because rolling bearings are critical structural elements, defect detection typically requires manual disassembly by professionals, which may disturb other components and introduce additional risks. In practice, the importance of bearing fault diagnosis is amplified by the growing use of automated and predictive maintenance systems in modern industry. Accurate detection and diagnosis extend the service life of critical machinery and help prevent catastrophic failures, costly downtime, and safety incidents (Althubaiti et al., 2022). Consequently, large-scale industrial operations increasingly integrate advanced diagnostic and intelligent monitoring systems to detect early bearing faults, reduce unplanned maintenance costs, and improve operational efficiency. In the context of Industry 4.0, leveraging data-driven insights for proactive maintenance further makes the development of robust and reliable bearing fault diagnosis methods a critical research topic. Therefore, the development of artificial intelligence (AI) models that diagnose faults by analyzing operational signals from machinery has become a widely adopted approach.

Traditional machine learning methods were widely applied to bearing fault diagnosis long before the rapid development of deep learning. Lu et al. (Shuang and Meng, 2007) proposed a diagnostic method based on Principal Component Analysis (PCA) and Support Vector Machine (SVM) theory. They first used PCA to transform the multidimensional correlated variables of vibration signals into low-dimensional independent feature vectors and then employed SVM for fault pattern recognition and nonlinear regression. In the study by Van and Kang (2015), a Particle Swarm Optimization (PSO) algorithm was employed to enhance the feature selection process and identify effective features to improve the accuracy of multi-fault classification. Based on these selected features, they used SVM and an innovative decision fusion

[*] Corresponding author.
E-mail addresses: 2023210516027@stu.cqnu.edu.cn (J. Cheng), rshran@cqnu.edu.cn (R. Ran), fb@cqu.edu.cn (B. Fang), libenchao517@my.swjtu.edu.cn (B. Li).

https://doi.org/10.1016/j.engappai.2026.113821
Received 23 July 2025; Received in revised form 1 December 2025; Accepted 9 January 2026
Available online 12 January 2026
0952-1976/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

strategy to achieve effective classification of various faults. Other machine learning methods, such as k-Nearest Neighbors (KNN) (Cover and Hart, 1967) and Random Forest (RF), have also been applied to signal fault diagnosis. Beyond rotating machinery, related studies in structural health monitoring further demonstrate the potential of classical machine learning and manifold-learning techniques for damage detection. Torzoni et al. (2022) employ a reduced-order numerical model together with a Siamese convolutional network to learn a damage-oriented embedding. Peng et al. (2022) reconstruct the phase-space representation using the natural frequencies of the structure, and apply manifold learning and Gaussian regression to extract reliable damage indicators from the learned manifold. Entezami et al. (2025) further combine manifold learning, clustering, and multi-fidelity hyperparameter optimization to perform anomaly detection on large bridges, while Sarmadi et al. (2022) propose an anomaly detection method that integrates unsupervised feature selection with local metric learning, automatically selecting features from long-term natural frequencies and learning local Mahalanobis distance metrics.

In parallel with these advances in fault and damage diagnosis, intelligent optimization algorithms have been widely used to solve complex industrial decision-making problems. For example, a recent study proposed a multi-strategy quantum differential evolution algorithm with a cooperative co-evolution framework and hybrid local search for large-scale capacitated vehicle routing problems, achieving high-quality solutions and fast convergence on challenging benchmarks (Deng et al., 2025). However, traditional machine learning methods have significant limitations. They require extensive hyperparameter tuning to achieve optimal results and depend on manual extraction of low-dimensional features for model training when dealing with high-dimensional data. This manual feature extraction relies heavily on prior knowledge, which is a highly subjective and time-consuming process.

In recent years, the rapid development of deep learning has opened new avenues for addressing challenges in the field of bearing fault diagnosis. Deep learning methods can automatically learn features from raw vibration signals, overcoming the limitations of traditional machine learning methods. These approaches have achieved remarkable and compelling results in this domain. Eren (2017) proposed a bearing fault detection system using a one-dimensional convolutional neural network (1DCNN) (Kim, 2014), where raw vibration signals are input into the network. This approach reduces computational complexity without compromising fault detection accuracy. To handle high noise vibration data, Zhao et al. (2019) inserted a soft threshold layer as a nonlinear transformation into the classical CNN variant ResNet, effectively eliminating noise-related features. Chen et al. (2021) combined multi-scale CNNs with Long Short-Term Memory (LSTM) networks. They used two 1DCNNs with different kernel sizes for feature extraction and employed LSTM to identify fault types.

More recently, several studies have focused on improving robustness to distribution shift and data scarcity in bearing fault diagnosis. Li et al. (2025a) propose a cross-domain adaptation method that combines maximum classifier discrepancy with deep feature alignment, thereby significantly improving transfer accuracy under variable working conditions. Zhao et al. (2025) proposed a highly generalizable model-agnostic meta-learning approach (MAML-GA) that incorporates a task augmentation strategy into the meta-learning framework and employs an adaptive parameter update operator, thereby significantly enhancing the cross-domain generalization ability of the diagnostic model. In addition, a dual-branch Instance–Batch Normalization (IBN) MixStyle network, coupled with a dynamic-weighting invariant risk minimization strategy, has been developed to simultaneously learn discriminative and domain-invariant features from multi-source vibration data (Li et al., 2025b).

Generally, many machine learning models are typically developed in Euclidean space due to their simple and flat structure, where the distance between any two samples can be measured using Euclidean distance. However, in the real world, data often exhibits nonlinear characteristics, making Euclidean space insufficient to accurately describe the relationships between data points. In contrast, Riemannian space accounts for the curvature of the space, making it a nonlinear domain where the distance between any two samples is measured by the length of the geodesic. This allows for more accurate distance computation for nonlinear data. As a result, Riemannian space is more suitable than Euclidean space for handling complex, nonlinear datasets (Diepeveen, 2024).

Li et al. (2021) proposed a gearbox fault diagnosis method based on the Feature Fusion Covariance Matrix (FFCM) and Multivariate Riemannian Kernel Ridge Regression (MRKRR). The method integrates multi-sensor statistical features to construct the FFCM and applies the MRKRR model within the Riemannian manifold framework, avoiding kernel function selection while leveraging the structural information of the FFCM. Ye et al. (2022) introduced a new probabilistic autoencoder, AIRMAE, by incorporating the Affine-Invariant Riemannian Metric (AIRM) (Pennec et al., 2006) as a regularization term. They combined AIRMAE with machine learning classifiers for supervised, data-driven fault diagnosis of multilevel converters. Liu et al. (2022) extended the Semi-Supervised Locally Linear Embedding (SSLLE) method into Riemannian space. They transformed the raw dataset into a *symmetric positive definite (SPD) manifold* and employed SSLLE in the manifold's tangent space to perform dimensionality reduction. Their experiments demonstrated that this approach has stronger fault diagnosis capabilities compared to directly applying other machine learning methods to the raw data set.

However, the methods based on Riemannian manifolds do not address the following issues: Firstly, in practical engineering applications, environmental factors often introduce noise interference, resulting in deviations between the feature distributions of simulated and actual signals. These deviations can induce negative transfer effects, as referenced in Li et al. (2020), ultimately causing a decrement in the model's classification performance. Secondly, in fault diagnosis, the fault signals from equipment are typically high-dimensional and often contain a large amount of irrelevant or redundant information (Gu et al., 2018). This redundancy makes it difficult for the model to capture the core features of the data. Thirdly, although deep learning has shown good performance in bearing fault diagnosis, effectively recognizing and classifying different faults, limited data in real-world applications complicates model training, hinders the model's ability to learn the sample distribution, and restricts its generalization ability (Zhu et al., 2023).

Therefore, this paper proposes an **SPD manifold deep metric learning** model for feature extraction of bearing signal data, which extends the *denoising autoencoder* and *sparse penalty term* from deep learning, as well as the *multi-class* N*-pair loss* from deep metric learning, to the SPD manifold, and then the SVM is used as classifier. The diagnosis accuracy of this method is better than that of the Euclidean deep learning model. The contributions of this paper are as follows:

1. We construct a *sparse denoising autoencoder* based on *SPDNet*. By artificially adding noise to clean inputs, the clean signal is locally disrupted. The hidden layer units learn more robust features, effectively reducing the impact of noise on data representation. This allows the proposed network to capture the important features of the data. The proposed noise injection method for SPD matrices can effectively simulate real-world noise, demonstrating strong practicality and adaptability.
2. We extend the sparse penalty term in Euclidean space to the SPD manifold using *Log Determinant (LogDet) divergence*. This sparse penalty term can inject sparse priors into the model, allowing it to learn task-driven sparse structures while maintaining the geometry of the SPD manifold, thereby improving the computational efficiency, and generalization ability of the model.

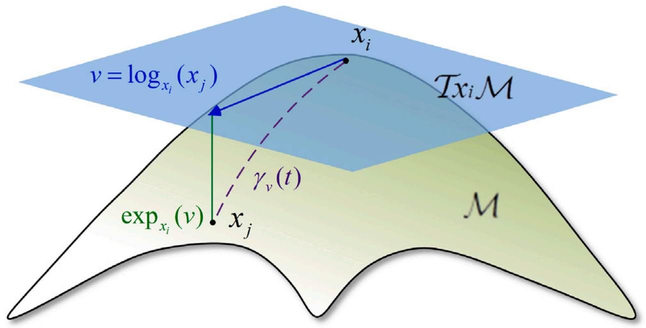


Fig. 1. Schematic diagram of SPD manifold.

3. We extend the multi-class N-pair loss to SPD manifolds and propose a novel deep metric learning on the SPD manifold. During each training process, the anchor SPD interacts with all negative-class SPD samples, maximizing the distance between heterogeneous samples and minimizing the distance between homogeneous samples on the SPD manifold, which enhances the performance of bearing fault diagnosis.

The structure of this paper is as follows: Section 2 introduces the SPD manifold, SPDNet, and the multi-class N-pair loss. Section 3 describes the proposed fault diagnosis method, and Section 4 introduces the experimental results. Finally, Section 5 summarizes the entire text and provides an outlook for future work.

2. Related work

2.1. SPD manifold

The SPD manifold, as a type of Riemannian manifold, is widely used in fields like electroencephalogram (EEG) signal classification (Suh and Kim, 2021). As a space with non-zero curvature, a Riemannian manifold has a geometric structure distinct from that of Euclidean space. For all non-zero vectors $v \in R^d$, real SPD matrices $X \in R^{d\times d}$ have the properties of $v^T X v > 0$ and $X^T = X$. The space formed by a set of SPD matrices $X_i \in R^{d\times d}$ lies inside the convex cone of an $d\times(d+1)/2$ dimensional Euclidean space, denoted as S^d_{++}. When S^d_{++} is endowed with a Riemannian metric, it becomes a specific Riemannian manifold, known as the SPD manifold. Several Riemannian metrics have been proposed for SPD manifolds, such as the Affine Invariant Riemannian Metric (AIRM) (Pennec et al., 2006) and the logarithmic Euclidean Metric (LEM) (Arsigny et al., 2007). For any two SPD matrices X_1 and X_2, directly using the distance calculation method from Euclidean space may lead to inaccurate results, as SPD matrices do not lie in Euclidean space but rather on the SPD manifold. Hence, their distance should be computed via the geodesic on the SPD manifold to ensure a more accurate and natural measurement. Fig. 1 provides a schematic illustration of the geometric structure of the SPD manifold, including the tangent space and the log–exp mappings, which helps visualize the relationships introduced in this section.

As shown in Fig. 1, $\mathcal{M}$ is an SPD manifold, and $T_{x_i}\mathcal{M}$ is the tangent space of X_i on $\mathcal{M}$. The logarithmic map projects a point on the manifold onto the tangent space, while the exponential map projects a point from the tangent space back to the manifold. The tangent space, logarithmic map, and exponential map enable the calculation of the geodesic distance $\gamma_v(t)$ between two SPD matrices x_i and x_j. The specific computation process is not detailed here, but for more information on geodesic calculations, refer to AIRM (Pennec et al., 2006) and other references.

2.2. SPDNet

Inspired by CNN, Huang and Van Gool (2017) extended the CNN paradigm to the SPD manifold and developed SPDNet. SPDNet is an end-to-end Riemannian neural network that introduces a deep nonlinear learning mechanism for SPD matrices. SPDNet consists of a Bilinear Mapping (*BiMap*) layer, an Eigenvalue Rectification (*ReEig*) layer, and an eigenvalue logarithm (*LogEig*) layer used for generating Euclidean representations. Suppose the input SPD matrix at the kth layer is $X_{k-1} \in R^{d_{k-1}\times d_{k-1}}$, then the layers of SPDNet are defined as follows:

BiMap Layer: This layer is similar to a fully connected convolutional layer, which maps the input SPD matrix to a new SPD matrix through bilinear mapping f_b. It can be represented as:

$$X_k = f_b^{(k)}\left(W_k, X_{k-1}\right) = W_k X_{k-1} W_k^T, \tag{1}$$

where $W_k \in R^{d_k\times d_{k-1}}\left(d_k < d_{k-1}\right)$ is a learnable transformation matrix and $X_k \in R^{d_k\times d_k}$ is the output matrix of this layer. To ensure that the data after the bilinear mapping f_b remains an SPD matrix, W_k needs to be a full column rank matrix (Huang and Van Gool, 2017).

ReEig Layer: This layer is similar to the Rectified Linear Unit (ReLU) layer in CNN architectures. It adjusts the small positive eigenvalues of each input SPD matrix through a nonlinear rectification function f_r, thereby injecting nonlinearity into SPDNet. It can be represented as:

$$X_k = f_r^{(k)}\left(X_{k-1}\right) = U \max\left(\epsilon I, \Sigma\right) U^T, \tag{2}$$

where ϵ represents the eigenvalues, $X_{k-1} = U\Sigma U^T$ performs eigenvalue decomposition on X_{k-1}. In all experiments, we set the ReEig threshold to $\epsilon = 10^{-4}$. This value is several orders of magnitude smaller than the typical eigenvalues of the covariance matrices constructed from vibration data, so it mainly suppresses numerically spurious near-zero modes while preserving the meaningful spectral structure. This choice keeps the matrices strictly positive definite and controls their condition numbers without distorting the underlying geometry.

LogEig Layer: This layer is designed to project elements from the SPD manifold onto the tangent space, enabling the use of Euclidean methods. It is represented as:

$$X_k = f_l^{(k)}\left(X_{k-1}\right) = U \log\left(\Sigma\right) U^T, \tag{3}$$

where $X_{k-1} = U\Sigma U^T$ performs eigenvalue decomposition on X_{k-1}. After the LogEig layer, conventional Euclidean network layers can be directly applied.

2.3. Comparison with existing SPD manifold methods

Existing SPD-based learning approaches, including deep learning models such as SPDNet and machine-learning-based methods such as SPD-SSLLE, demonstrate the utility of Riemannian geometry for feature representation; however, their structural designs inherently limit their robustness to noise, redundant feature correlations, and distributional uncertainty. SPDNet relies on BiMap, ReEig, and LogEig layers to extract manifold-aware features, yet it does not explicitly suppress irrelevant dimensions or correct noise-induced distortions in the SPD structure, so disturbances in vibration signals and spurious inter-channel correlations are directly propagated through the network. SPD-SSLLE performs dimensionality reduction in the tangent space without any denoising or sparsity-inducing prior, making it sensitive to high-dimensional redundancy.

In contrast, the proposed SPD sparse denoising autoencoder introduces a different modeling objective by enforcing robustness directly on the manifold: clean SPD matrices are reconstructed from intentionally perturbed inputs, improving invariance to environmental noise while preserving geometric structure through log–exp mappings. The Log Determinant(LogDet) divergence-based sparsity term further suppresses redundant correlations in SPD matrices – capabilities absent in SPDNet and SPD-SSLLE – while the multi-class N-pair metric loss enhances global class separability on the manifold. Together, these components form a structurally distinct and functionally more robust SPD learning framework tailored for fault diagnosis under noise and redundancy conditions.

3. The proposed method

In this section, the SPD manifold deep metric learning model for feature extraction of bearing signal data and the corresponding fault diagnosis are introduced in detail.

3.1. The SPD manifold modeling of signal data

Let $X = \left(x_1, x_2, \dots, x_N\right) \in R^{N\times 1}$ be the original bearing signal data, which is a time series. In most fault diagnosis scenarios, these signals are typically collected using accelerometers due to their high sensitivity to mechanical vibrations. However, accelerometer-based measurements are susceptible to sensor placement, structural resonance, and environmental noise, which may limit their ability to capture complete system dynamics. Therefore, directly using raw one-dimensional time-series signals for fault diagnosis has certain drawbacks, such as weak feature representation and poor robustness. Therefore, We begin by applying the *phase space reconstruction* method (Takens, 2006) to map the raw univariate time series data into a higher-dimensional space, and then structure a SPD matrix to represent the data (Liu et al. (2022)). The overall procedure for transforming the raw vibration signal into an SPD matrix – including phase space reconstruction and covariance computation – is summarized in Fig. 2, which provides a visual overview of the steps described below.

The process of converting the time series $X = \left(x_1, x_2, \dots, x_N\right)$ into a matrix $m \in R^{d_1\times d}$ can be written as follows:

$$m_i = \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(d-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(d-1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_{d_1} & x_{d_1+\tau} & \cdots & x_{d_1+(d-1)\tau} \end{bmatrix} = \begin{bmatrix} m^i_{11} & m^i_{12} & \cdots & m^i_{1d} \\ m^i_{21} & m^i_{22} & \cdots & m^i_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ m^i_{d_1 1} & m^i_{d_1 2} & \cdots & m^i_{d_1 d} \end{bmatrix} \in R^{d_1\times d} \tag{4}$$

where τ represents the time delay parameter, d is the embedding dimension(In our experiment, τ and d were set to 3 and 10, respectively, based on the average mutual information (AMI) method (Fraser and Swinney, 1986) and Cao's method (Cao, 1997)).

To provide a clear theoretical basis for these selections, we briefly summarize the rationale behind these two methods. The AMI method measures the nonlinear dependency between $X(t)$ and $X(t+\tau)$. When τ is too small, the variables remain highly correlated and contain redundant information; when it is too large, their dependency diminishes, making it difficult to reveal the underlying system dynamics. Therefore, the first local minimum of the AMI curve is typically adopted as the optimal delay, as it minimizes redundancy while preserving informative structure. The embedding dimension d is then estimated using Cao's method, which examines how the neighborhood relationships of reconstructed vectors change as the embedding dimension increases from d to $d+1$. If increasing d substantially alters the relative distances among neighboring vectors, the current dimension is considered insufficient. Once these changes become negligible and the reconstructed series behaves differently from a purely random sequence, the underlying dynamics are regarded as adequately unfolded, and the corresponding embedding dimension is selected.

Furthermore, d_1 is the number of reconstructed phase points, calculated as $d_1 = N - (d-1)\tau$. Then we compute:

$$p_i = m_i^T m_i \in R^{d\times d}$$

p_i is an SPD matrix corresponding to the original signal data X. As shown in Fig. 2, the raw time series is embedded into the reconstructed phase space to form matrix m, from which the SPD matrix $p = m^T m$ is computed.

Each SPD matrix p effectively encodes the statistical characteristics of the reconstructed phase space. Specifically, the diagonal elements p_{ii} represent the total energy of all state vectors in the ith dimension of the phase space, while the off-diagonal elements p_{ij} $(i \neq j)$ correspond to the uncentered covariance between the ith and jth dimensions. The diagonal entries thus reflect the global statistical behavior of the system's dynamic energy across different delay scales τ_i (self-correlation), and the off-diagonal entries characterize the dynamic coupling strength between different delay scales τ_i and τ_j (cross-correlation). Therefore, the SPD matrix p provides a mathematically tractable description of nonlinear characteristics in the bearing vibration signal.

3.2. SPD multi-class N-pair loss

Multi-class N-pair loss (Sohn, 2016) is an improved contrastive loss function designed to enhance the model's classification performance by considering the relative relationships between positive and negative samples. The effect of this loss on sample distribution is illustrated in Fig. 3, which visually shows how N-pair optimization brings within-class samples closer while pushing apart between-class samples on the SPD manifold. Let $p \in P$ represent the input SPD matrix, p^+ and p^- represent the positive and negative examples of p, where p^+ and p are from the same class, and p^- is from a different class. In this method, each sample p is compared not only with a single negative sample p^-, but also with $N-1$ negative samples from different classes. The loss function minimizes the distance between the positive sample p^+ and the sample p, while maximizing the distance between the negative sample p^- and p. This method further strengthens the distinction between-class samples by leveraging multiple negative examples, allowing the model to maximize the distance of between-class samples and minimize the distance of within-class samples in the embedding space, thereby improving the model's classification performance.

Following multi-class N-pair loss, we adopt a mini-batch construction where each class contributes one anchor–positive pair. Concretely, in each mini-batch we select N distinct fault classes and randomly sample two SPD matrices from each class. This yields $P = \{(p_i, p_i^+)\}_{i=1}^N$ where p_i and p_i^+ are two SPD matrices from the same class i. For a given anchor p_i, its within-class positive is p_i^+, while the positives from all other classes $\{p_j^+\}_{j\neq i}$ are reused as negatives.

As shown in Fig. 3: The SPD multi-class N-pair loss interacts with all negative classes during each update. Each anchor point has corresponding $N-1$ negative examples for loss calculation. It not only brings the positive samples closer but also pushes the $N-1$ negative samples further away, thereby significantly enhancing the discriminability between different categories.

Since we are optimizing the distance between samples using a multi-class N-pair loss on the SPD manifold, the Riemannian metric on the SPD manifold needs to be used when calculating the distance between samples. The performance of the LEM is comparable to that of the AIRM, but LEM has a significantly lower computational cost (Arsigny et al., 2007). Therefore, we choose to use the logarithmic Euclidean metric. For two SPD matrices p_i and p_j, the logarithmic Euclidean distance is defined as: $D\left(p_i, p_j\right) = \left\|\log\left(p_i\right) - \log\left(p_j\right)\right\|_F^2$. Thus, the SPD multi-class N-pair loss is given by:

$$L_1\left(\theta_1; \left\{\left(p_i, p_i^+\right)\right\}_{i=1}^N\right) = \frac{1}{N}\sum_{i=1}^{N}\log\left(1 + \sum_{j\neq i}\exp\left(D\left(p_i, p_i^+\right) - D\left(p_i, p_j^+\right)\right)\right) \tag{5}$$

where p_j^+ represents the positive examples from the other $N-1$ classes, excluding the ith class, and is used as the negative examples for the ith class in this context, θ_1 denotes the learnable encoder BiMap layer. For completeness, Appendix B of the supplementary material provides the full gradient derivation and numerical stability considerations of the SPD multi-class N-pair loss used in Eq. (5).

From a computational point of view, for a mini-batch containing N classes (and thus N anchor–positive pairs), the SPD multi-class N-pair

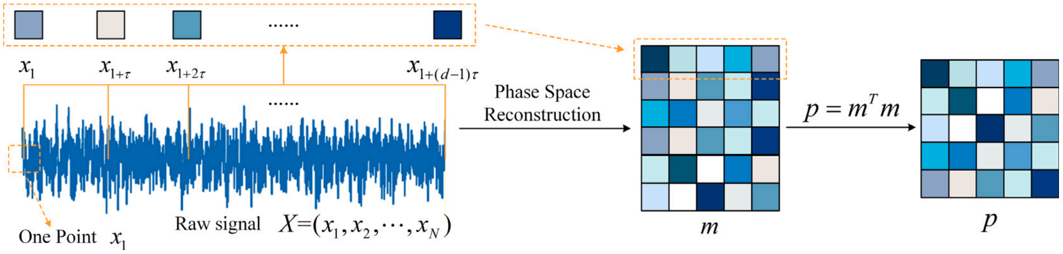


Fig. 2. Construction of SPD Matrix via Phase Space Reconstruction: The original one-dimensional signal is embedded into a higher-dimensional phase space using the phase space reconstruction method. Each reconstructed vector consists of signal value x_i sampled at intervals defined by a time delay τ, forming the matrix m. The final symmetric positive definite (SPD) matrix p is obtained by computing $p = m^T m$.

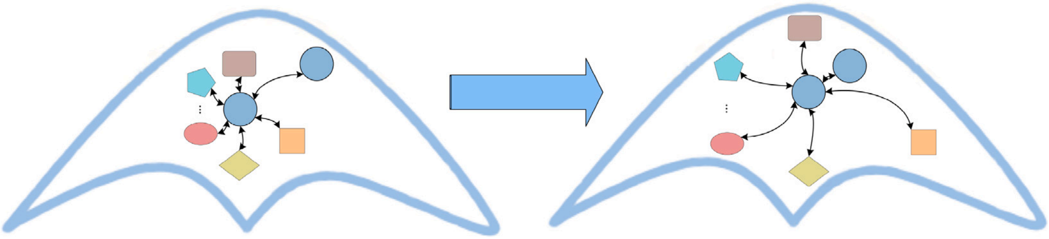


Fig. 3. Our proposed metric learning method can effectively optimize the sample distribution on SPD manifolds. In the above figure, the colors and shapes of homogeneous samples are the same. In the left figure, the distance between heterogeneous samples is relatively close. After optimizing the metric learning term (as shown in the right figure), homogeneous samples are brought closer together to form tighter clusters, while the distance between heterogeneous samples is widened, significantly improving the discrimination ability. The distance between samples is the geodesic distance, not the Euclidean distance, and this corresponds to the LEM on the SPD manifold.

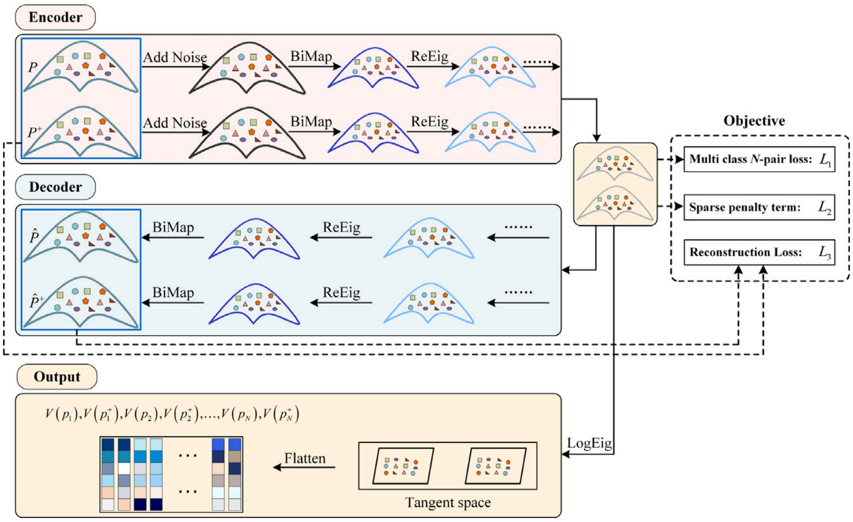

Encoder
P
P^+
Add Noise
BiMap
ReEig
......
Objective
Multi class N-pair loss: L_1
Sparse penalty term: L_2
Reconstruction Loss: L_3
Decoder
$\hat{P}$
$\hat{P}^+$
BiMap
ReEig
......
Output
$V(p_1),V(p_1^+),V(p_2),V(p_2^+),\ldots,V(p_N),V(p_N^+)$
Flatten
Tangent space
LogEig

Fig. 4. The proposed network consists of an encoder, a decoder, and an output module. Clean SPD matrices are first corrupted with noise and passed through the encoder to obtain latent representations. The decoder reconstructs SPD matrices from these representations, and the reconstruction error between clean and reconstructed SPD matrices is minimized. A sparsity penalty is imposed on the latent representations to suppress redundancy and improve generalization, while a multi-class N-pair loss pulls samples from the same class together and pushes samples from different classes apart. Finally, the latent representations are mapped to the tangent space by a LogEig layer and flattened to form the output feature vectors.

loss in Eq. (5) involves $O(N^2)$ pairwise interactions $D(p_i, p_j^+)$. In our setting, however, N is bounded by the number of fault types in each dataset, so this quadratic factor remains small in practice. Moreover, the overall runtime of the N-pair term is typically dominated by the BiMap and ReEig layers rather than the pairwise interaction cost itself.

3.3. Theoretical analysis of SPD manifold metric learning

For practical fault diagnosis, it is crucial that the deep learning model can be trained stably and yields reliable separations between fault classes. This subsection introduces the theoretical results that guarantee these properties for the proposed SPD manifold framework. In real industrial systems, a diagnosis model must (i) have an optimization landscape with a single well-defined global minimizer and no spurious local minima, so that gradient-based training is predictable and stable, (ii) be insensitive to changes of coordinate system or sensor arrangement, and (iii) adopt a loss that decreases as class separability improves, so that optimization directly yields better fault classification. The following theorems formalize these properties. Geodesic convexity and strong convexity of the multi-class N-pair loss on the SPD manifold guarantee the existence and uniqueness of a global minimizer and ensure that Riemannian gradient descent enjoys Euclidean-like linear convergence. The similarity invariance result shows that the loss is unchanged under congruent (similarity) transformations, reflecting robustness of the learned metric to different but equivalent signal representations (e.g., sensor placement or preprocessing changes). Finally, the margin-based bound establishes a direct link between the geodesic margin of different fault classes and the value of the N-pair loss, clarifying that enlarging inter-class geodesic margins on the SPD manifold leads to an exponential decrease of the optimization objective and thus more discriminative fault separation.

Theorem 3.1 (*Geodesic Convexity of N-Pair Loss on the SPD Manifold*). *Let S^d_{++} denote the set of $d \times d$ SPD matrices endowed with the log-Euclidean metric. Given pairs $\{(\mathrm{p}_i, p_i^+)\}_{i=1}^N \subset S^d_{++}$, define features $x_i = \phi(\log p_i)$ and $x_i^+ = \phi(\log p_i^+)$ via a differentiable mapping $\phi : \mathrm{Sym}(d) \to \mathbb{R}^m$. The multi-class N-pair loss*

$$\mathcal{L}(\{p_i, p_i^+\}) = -\sum_{i=1}^{N} \log \frac{\exp\big(\langle x_i, x_i^+\rangle\big)}{\sum_{j=1}^{N} \exp\big(\langle x_i, x_j^+\rangle\big)}$$

is geodesically convex on S^d_{++}. The detailed derivation of this equivalent inner-product form of the N-pair loss is provided in Appendix A of the supplementary material. Moreover, if ϕ is linear, then $\mathcal{L}$ is geodesically strongly convex and admits a unique global minimizer.

Proof. (1) Under the log-Euclidean metric, S^d_{++} is a complete, simply connected and flat Hadamard manifold (Sra and Hosseini, 2013) with distance $d\,(S,T) = \|\log(S) - \log(T)\|_F^2$. The exponential and logarithm maps provide an isometry between S^d_{++} and the vector space $\mathrm{Sym}(d)$.

(2) In this vector space, the N-pair loss is a composition of the convex log-sum-exp function with an affine map; such compositions are convex.

(3) Since the logarithmic map is a linear isometry, the function $\mathcal{L}(\exp(\cdot))$ on the SPD manifold is equivalent to the composition of a convex function and an affine mapping in Euclidean space. Consequently, $\mathcal{L}$ is geodesically convex, i.e.,

$$\mathcal{L}(\gamma(t)) \;\le\; (1-t)\,\mathcal{L}(P_1) + t\,\mathcal{L}(P_2),$$

for any geodesic curve $\gamma(t)$ connecting P_1 and P_2 (Boyd and Vandenberghe, 2004).

(4) If the mapping ϕ is linear and invertible, then the Hessian of $\mathcal{L}$ is positive definite in the associated Euclidean vector space. As a result, $\mathcal{L}$ is geodesically strongly convex on the SPD manifold. Since the SPD manifold is a Hadamard manifold and strongly convex functions on Hadamard manifolds admit a unique minimizer, $\mathcal{L}$ therefore has a unique minimizer on the SPD manifold. □

Corollary 3.1 (*Linear convergence of Riemannian gradient descent*). *If the mapping ϕ is linear and the N-pair loss $\mathcal{L}$ is geodesically μ-strongly convex with a Lipschitz continuous gradient, then Riemannian gradient descent with constant step size $\eta = 1/L_g$ converges linearly to the unique minimizer on S^d_{++}. Specifically, on flat manifolds with curvature $\kappa = 0$, Theorem 15 in Zhang and Sra (2016) shows that*

$$\mathcal{L}(x_t) - \mathcal{L}(x^*) \;\le\; (1-\epsilon)^{t-2}\,\frac{L_g D^2}{2}, \qquad \epsilon = \min\left\{\frac{1}{\zeta(\kappa, D)},\ \frac{\mu}{L_g}\right\},$$

where L_g is the Lipschitz constant of the Riemannian gradient and D is the distance from the initial iteration to the optimal solution. κ is a lower bound on the sectional curvature of the manifold, and $\zeta(\kappa, D)$ is the geometric factor defined in Zhang and Sra (2016). In our case, the SPD manifold S^d_{++} is flat with curvature $\kappa = 0$, so $\zeta(\kappa, D)$ reduces to 1, and hence $\epsilon = \min\{\mu/L_g, 1\}$, which coincides with the Euclidean convergence bound.

Proof. By Theorem 3.1, the N-pair loss $\mathcal{L}$ is geodesically μ-strongly convex on S^d_{++}. Therefore, in our setting the loss $\mathcal{L}$ has an L_g-Lipschitz continuous Riemannian gradient, and the curvature of the SPD manifold satisfies $\kappa = 0$. Hence, the stated linear convergence rate follows directly from Theorem 15 in Zhang and Sra (2016). □

Corollary 3.2 (*Similarity Invariance*). *For any invertible matrix $A \in \mathrm{GL}(d)$, define the similarity transformation $T_A : S^d_{++} \to S^d_{++}$. The multi-class N-pair loss satisfies*

$$\mathcal{L}\big(\{T_A(P_i),\, T_A(P_i^+)\}\big) = \mathcal{L}\big(\{P_i,\, P_i^+\}\big).$$

Proof. (1) According to the Log-Euclidean framework (Arsigny et al., 2007), the distance under the log-Euclidean metric satisfies

$$d(A \cdot S,\ A \cdot T) = d(S, T),$$

for any invertible matrix $A \in \mathrm{GL}(d)$, which means that the SPD manifold remains isometric under similarity transformations.

(2) For the inner-product terms, we have

$$\langle \phi(\log T_A(P)),\ \phi(\log T_A(P^+))\rangle = \langle A\log(P)A^\top,\ A\log(P^+)A^\top\rangle.$$

Since the Frobenius inner product satisfies

$$\langle AMA^\top,\ ANA^\top\rangle = \mathrm{tr}\big((AMA^\top)^\top(ANA^\top)\big) = \mathrm{tr}(AM^\top NA^\top) = \mathrm{tr}(M^\top N),$$

the similarity transformation does not change the value of the inner product.

(3) Therefore, all exponential terms of the form $\exp(\langle\cdot,\cdot\rangle)$ in the N-pair loss remain unchanged, and the entire loss $\mathcal{L}$ is invariant under similarity transformations. This demonstrates that the proposed model possesses coordinate invariance and robustness. □

Corollary 3.3 (*Margin-based Bound*). *Assume that for each anchor i and every negative sample $j \neq i$, there exists a geodesic margin $\gamma > 0$ such that*

$$D\left(p_i, p_i^+\right) + \gamma \le D\left(p_i, p_j\right). \tag{6}$$

Then each term of the multi-class N-pair loss satisfies

$$\log\Big(1 + \sum_{j \neq i} \exp\big(D(p_i, p_i^+) - D(p_i, p_j)\big)\Big) \;\le\; \log\Big(1 + (N-1)e^{-\gamma}\Big), \tag{7}$$

and the full loss obeys the bound

$$\mathcal{L}_{\mathrm{N}} \;\le\; N \cdot \log\Big(1 + (N-1)e^{-\gamma}\Big). \tag{8}$$

Consequently, as the geodesic margin γ increases, the N-pair loss decays exponentially, meaning that the optimization objective explicitly encourages learning representations with large geodesic margins on the SPD manifold.

Proof. The assumption (6) implies that for every negative sample $j \neq i$,

$$D(p_i, p_i^+) - D(p_i, p_j) \;\le\; -\gamma.$$

Thus each exponential term in Eq. (5) satisfies

$$\exp\big(D(p_i, p_i^+) - D(p_i, p_j)\big) \;\le\; e^{-\gamma}.$$

Summing over the $N-1$ negative samples yields inequality (7). Finally, summing inequality (7) over all $i = 1, \ldots, N$ gives the inequality (8). □

3.4. SPD sparse denoising autoencoder

Fig. 4 is a schematic diagram of the network structure proposed in this paper. The proposed network consists of an *encoder*, a *decoder*, and an output module. The encoding and decoding module consists of the BiMap (bilinear map) layer and the ReEig (eigenvalue rectification) layer from the original SPDNet, which are used to get a hidden layer representation of the input SPD data while maintaining the Riemannian structure of the underlying SPD manifold.

Let N be the number of categories of the samples. Then, during the single training of the proposed network, the batch data needs to contain anchor samples p_i and positive samples p_i^+ of each class, with a total of $2N$ samples. Let $P = \left[p_1, p_2, \dots, p_N\right]$, $\tilde{P} = \left[\tilde{p}_1^+, \tilde{p}_2^+, \dots, \tilde{p}_N^+\right]$ are anchor sample and the corresponding positive sample sets.

3.4.1. Noise simulation

In real-world scenarios, mechanical equipment systems contain a large amount of noise. In order to make the model adaptable to real industrial environments, we added noise to the original input SPD data, making the proposed method robust to noise interference. In 2023, Moysidis et al. (2023) modeled the noise in real-world environments as Gaussian noise, which is a practical and statistical based method. From this, we model noise in industrial environments as Gaussian noise in this paper. Specifically, for a clean SPD matrix p_i, we first apply the matrix logarithm operation to project it onto the tangent space (i.e., the space of symmetric matrices), where additive Gaussian noise can be naturally introduced while preserving the geometric consistency of the SPD manifold.

$$u_i = \log\left(p_i\right) = U \log\left(\Sigma\right) U^T \tag{9}$$

where $p_i = U \Sigma U^T$ denotes the eigenvalue decomposition. Next, we generate a zero matrix N of the same size as p_i, with its diagonal elements N_{ii} sampled from a Gaussian distribution $\mathcal{N}\left(0, \sigma^2\right)$, and its upper triangular elements $N_{ij}\,(i < j)$ sampled from a Gaussian distribution $\mathcal{N}\left(0, \rho^2\right)$. The lower triangular part is symmetrically assigned as $N_{ji} = N_{ij} (i < j)$, forming a symmetric Gaussian noise matrix. We then add the noise matrix to the symmetric matrix u_i in the tangent space and project the result back onto the SPD manifold using matrix exponentiation, yielding the noisy SPD matrix:

$$\tilde{p}_i = \exp\left(u_i + \gamma N\right) \tag{10}$$

where γ is the noise intensity factor that controls the magnitude of the disturbance. The parameters for the diagonal Gaussian noise σ^2, the off-diagonal Gaussian noise ρ^2, and the noise intensity factor γ are set to 0.1, 0.001, and 0.1, respectively. This technique effectively simulates the process of introducing noise into the SPD matrix. Experiments on real-world datasets in Section 4 clearly demonstrate that the proposed method significantly mitigates the impact of noise on fault detection in industrial production. In addition, we perform a sensitivity analysis of the noise parameters σ^2, ρ^2, and γ; the corresponding accuracy–parameter curves and robustness bandwidths are reported in Appendix D of the supplementary material.

3.4.2. The sparse penalty term

For the sparsity penalty term in SPD space, we first use LogDet divergence (Davis et al., 2007) to define the similarity between two SPD matrices A and B:

$$D_{ld}\left(A, B\right) = tr\left(AB^{-1}\right) - \log\det\left(AB^{-1}\right) - d \tag{11}$$

where d is the number of rows or columns of A, and $\log\det\left(AB^{-1}\right)$ represents taking the logarithm of the determinant of the product matrix of A and B^{-1}. If the LogDet divergence values of two SPD matrices are smaller, that means they are more similar. For the SPD matrix $P = \left[p_1, p_2, \dots, p_N\right]$ obtained through the encoder, the corresponding sparse penalty term calculation formula is as follows:

$$L_2\left(P\right) = \frac{1}{N}\sum_{i=1}^{N} -\left(tr\left(p_i S^{-1}\right) - \log\det\left(p_i S^{-1}\right) - d\right) \tag{12}$$

where S represents a sparse SPD matrix with target sparsity, and its definition process is as follows.

First, initialize a negative identity matrix as the Cholesky factor. Next, generate a sparse lower triangular matrix LTM, where the density of non-zero elements is controlled by the parameter α (A larger α results in a sparser matrix, and we set α to 0.5). To verify the robustness of this parameter, a small sensitivity analysis was conducted by varying $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. As summarized in Table 1, the classification accuracy remains stable within the range 0.3–0.7, with the best performance achieved at $\alpha = 0.5$, while only extreme values lead to noticeable degradation. This indicates that the proposed sparse penalty term is effective yet insensitive to the choice of α.

Table 1
Sensitivity analysis of the sparsity control parameter α.

Parameter α	0.1	0.3	0.5 (default)	0.7	0.9
Accuracy (%)	95.86	96.46	**99.44**	98.47	93.90

The values of the non-zero elements are sampled from a uniform distribution $\mathcal{U}\left(0.1, 0.9\right)$ to introduce randomness. To further randomize the sparse distribution and disrupt the row and column order, the matrix LTM is randomly permuted in both rows and columns:

$$LTM' = P \cdot LTM \cdot P^T \tag{13}$$

where P is a permutation matrix generated by a random permutation. The permuted matrix is then overlaid onto the initial Cholesky factor, resulting in a new sparse lower triangular matrix: $chol = -I + LTM'$. Finally, the sparse SPD matrix is obtained through the Cholesky inverse operation: $S = chol^T \cdot chol$.

To ensure the applicability of the Euclidean method to symmetric positive definite matrices derived from the encoder, a LogEig layer is introduced following the final BiMap layer. This layer serves to project the SPD matrix onto the tangent space, transform it into a vectorized representation within this space, and utilize it as the output of the proposed network. Given an SPD matrix p_i, $V\left(p_i\right)$ denotes its vectorized form after processing through the LogEig layer.

3.4.3. Decoding stage

The decoder is structurally symmetric to the encoder, and its output matches the input size of the encoder. The reconstruction loss between the reconstructed SPD matrix obtained through the decoder and the input clean SPD matrix is defined as follows:

$$L_3\left(\theta_1, \theta_2, \phi; P\right) = \frac{1}{N}\sum_{i=1}^{N} \left\|p_i - \hat{p}_i\right\|_F^2 + \frac{1}{N}\sum_{i=1}^{N} \left\|p_i^+ - \hat{p}_i^+\right\|_F^2 \tag{14}$$

where θ_2 denotes the learnable decoder BiMap layer, $\hat{p}_i = \phi_{\left(\theta_1, \theta_2\right)}\left(\tilde{p}_i\right)$ represents the SPD matrix reconstructed by the proposed network, and $\phi_{\left(\theta_1, \theta_2\right)}$ represents the nonlinear Riemannian network embedding from the noisy SPD matrix to the reconstructed SPD matrix. To compute the reconstruction loss, we employ the Euclidean metric (EuM) instead of LEM. Because EuM avoids the need to compute the inverse of P during optimization (Wang et al., 2023). Moreover, as rigorously shown in Appendix E of the supplementary material, for any pair of SPD matrices the squared Euclidean and squared log–Euclidean discrepancies are locally equivalent up to multiplicative constants; therefore, minimizing the Euclidean reconstruction loss is (up to constants) equivalent to minimizing the corresponding LEM-based loss.

Through the encoding and decoding mechanism, we can obtain the hidden layer representation of the input SPD matrix. For the SPD matrix output by the encoder, we can use metric learning from the previous section to improve the model's discriminative ability. In addition, we also applied sparse penalty terms on the SPD manifold to encourage the BiMap layer of the encoder to learn a projection method that suppresses redundant dimensions in the input SPD matrix, thereby indirectly applying sparsity to the output SPD matrix. The output SPD matrix is a sparse SPD matrix, indicating that the correlation between features is concentrated in a limited number of dimensions. These dimensions may capture key discriminative patterns, thereby enhancing the interpretability of the model. Additionally, sparse SPD matrices tend to encapsulate information more efficiently, minimizing the influence of irrelevant dimensions and improving the model's generalization capability, especially when data is limited.

3.5. Objective function

The loss function of the proposed network in this paper is expressed as:

$$L\left(\theta_3, \phi; P\right) = L_1\left(\theta_1; P\right) + \lambda_1 L_2\left(\theta_1; P\right) + \lambda_2 L_3\left(\theta_1, \theta_2; \phi; P\right) \tag{15}$$

where $\theta_3 = \{\theta_1, \theta_2\}$, λ_1 is the weight parameter for the sparsity penalty term, and λ_2 is the weight parameter for the reconstruction loss. By using the SPD multi-class N-pair loss, the inter-class distance of SPD data is maximized while minimizing the intra-class distance, effectively mapping the original data features to a more discriminative feature space. Additionally, the use of a sparse penalty term removes irrelevant features, allowing the model to focus on meaningful signal features and reducing the risk of overfitting. Furthermore, the reconstruction loss between the clean input SPD matrix and the SPD matrix reconstructed from noisy data helps eliminate environmental interference, thereby enhancing the model's generalization capability.

To adaptively control the relative importance of each loss term, we introduce a dynamic adjustment strategy for the hyperparameters λ_1 and λ_2. This is achieved using the Nesterov Accelerated Gradient (NAG) (Nesterov, 1983) method, which enables the model to adjust λ_1 and λ_2 throughout training, rather than relying on manually tuned fixed values. In all experiments, we initialize λ_1 and λ_2 to 0.1 and 0.01, respectively. Since the reconstruction loss L_3 typically has a larger numerical scale than the other terms, a smaller initial value is used for λ_2 so that all loss components have comparable magnitudes at the beginning of training.

Considering the overall objective function:

$$L = L_1 + \lambda_1 L_2 + \lambda_2 L_3, \tag{16}$$

we treat λ_1 and λ_2 as trainable variables. After each iteration of network parameter updates (i.e., for each mini-batch), we fix the model parameters and update λ_1 and λ_2 by minimizing the total loss L using the following NAG update rule:

$$\begin{aligned} v_t^{(\lambda)} &= \mu v_{t-1}^{(\lambda)} - \eta \nabla_\lambda L(\lambda_t + \mu v_{t-1}^{(\lambda)}), \\ \lambda_{t+1} &= \lambda_t + v_t^{(\lambda)}, \end{aligned} \tag{17}$$

where $\lambda \in \{\lambda_1, \lambda_2\}$, $v_t^{(\lambda)}$ is the momentum term, μ is the momentum coefficient, and η is the learning rate for the hyperparameters. In our implementation, we set $\mu = 0.9$ and $\eta = 1 \times 10^{-3}$. The gradient $\nabla_\lambda L(\cdot)$ is computed with respect to the total loss.

To preserve the semantic meaning of the sparsity and reconstruction penalties and avoid numerical instability, we further impose non-negativity and interval constraints on λ_1 and λ_2. After each NAG update, the trade-off parameters are projected onto predefined positive ranges:

$$\lambda_{k,t+1} = \min\left(\max(\tilde{\lambda}_{k,t+1}, \lambda_k^{\min}), \lambda_k^{\max}\right), \quad k \in \{1, 2\}, \tag{18}$$

where we set $\lambda_1^{\min} = 1 \times 10^{-3}$, $\lambda_2^{\min} = 1 \times 10^{-4}$ and $\lambda_1^{\max} = \lambda_2^{\max} = 1$. This projection prevents the weights from becoming negative or excessively large, which would otherwise distort the contribution of the corresponding loss terms. The evolution of λ_1 and λ_2 over training epochs, illustrating the stability of this dynamic weighting scheme, is provided in Appendix C of the supplementary material.

The overall training procedure alternates between the following steps:

- Update all network parameters based on the total loss L for the current mini-batch.
- Fix network parameters and apply NAG to update λ_1 and λ_2, followed by the projection step.
- Repeat the above steps for each training iteration.

Table 2
Classification performance (%) of different methods on the CWRU dataset. Accuracy is reported under different training sample sizes α, while Precision and F1 are computed at $\alpha = 70$.

Method	Accuracy vs. training sample size α					Metrics at $\alpha = 70$	
	30	40	50	60	70	Precision	F1
DAE-SVM	14.52	25.97	16.33	17.08	31.39	30.75	31.07
MD-2DCNN	30.95	40.83	45.83	53.75	54.44	53.50	53.97
BiGRU	53.45	60.69	61.66	63.12	67.50	66.92	67.21
1DCNN	79.16	81.52	85.33	91.25	92.77	90.74	91.74
MCNN-LSTM	82.38	85.69	92.50	93.54	96.17	96.23	96.13
EVGG	80.59	81.13	84.53	85.71	87.58	88.18	87.88
AM-CNN	83.04	86.36	90.74	93.89	96.83	95.27	96.04
ICNN	86.01	89.21	92.65	94.76	97.66	97.35	97.50
DRSN	91.90	93.05	93.16	95.62	97.08	97.31	97.08
GTFENet	95.71	95.97	97.83	97.91	99.16	99.28	99.25
SPD-SSLLE	77.86	85.28	87.00	90.00	90.83	91.76	91.10
SPDNet	89.06	89.34	93.40	93.95	94.83	96.01	94.70
Ours	**96.43**	**97.36**	**98.83**	**98.12**	**99.44**	**99.38**	**99.34**

4. Experiments

In this section, we evaluate the proposed network on several public bearing datasets, including the Case Western Reserve University (CWRU) dataset (Smith and Randall, 2015), the University of Ottawa (Ottawa) dataset (Huang and Baddour, 2018), the Paderborn University (PU) dataset (Lessmeier et al., 2016), and the SCA (Svenska Cellulosa Aktiebolaget) dataset (Lundström and O'Nils, 2023). Unlike the first three, the SCA dataset was collected in a real industrial environment rather than under laboratory conditions, allowing us to evaluate algorithm performance in a more realistic setting. This dataset contains vibration signals of 11 bearings collected from pulp mills between 2019 and the end of 2022. Each bearing has a different model, source machine, and fault type. More detailed information can be found in Lundström and O'Nils (2023).

We compared the proposed algorithm with several well-known Euclidean methods and SPD manifold methods. Specifically, the SPD manifold methods include SPDNet and SPD-SSLLE (Semi-Supervised Local Linear Embedding), while the Euclidean methods encompass Denoising Autoencoder (Vincent et al., 2008), 1DCNN (Kim, 2014), MD(Metric Based)-2DCNN (Huang et al., 2022), MultiCNN-LSTM (Chen et al., 2021), DRSN(Deep Residual Shrinkage Network) (Zhao et al., 2019), EVGG(Evidential VGG) (Zhou et al., 2023), AM-CNN(Anti-noise Multi-scale CNN) (Jin et al., 2022), ICNN (CNN+VMD+CWT) (Gu et al., 2022), GTFENet(Gram Time–Frequency Enhanced Network) (Jia et al., 2023), and others. For SPD-SSLLE, we use grid search to find the optimal parameters on each dataset, while the other deep learning methods use the parameters recommended by the original authors.

4.1. Implementation details

We first model the original vibration signals into SPD matrices and then split them into training and testing sets. Noise is added to the training set and input into the proposed network shown in Fig. 4 to learn a more discriminative embedding. The proposed network is trained using the stochastic gradient descent (SGD) algorithm with a learning rate of 0.1. As the proposed network converges, both training and testing sets are input to obtain embedding vectors. An SVM classifier is trained using the training set embeddings, and the test set embeddings are classified to evaluate the method's effectiveness.

4.2. The performance comparison experiments

4.2.1. CWRU dataset experiment

The CWRU dataset includes ten fault modes: one normal signal and three fault locations(ball, inner race, and outer race) × three fault diameter sizes (7 inches, 14 inches, and 21 inches). Abbreviated as NR, B1, B2, B3, IR1, IR2, IR3, OR1, OR2, and OR3. We assessed the performance of various methods with different sample sizes, using training sample proportions of 30%, 40%, 50%, 60%, and 70%. The classification accuracies at different training proportions, together with Precision and F1-score at $\alpha = 70$, are summarized in Table 2.

As shown in Table 2, our method demonstrates excellent performance across all training sample sizes on the CWRU dataset. Notably, it achieves a high accuracy of 96.43% with only 30% samples, significantly outperforming other methods (e.g., surpassing the second-best DRSN by over 4.5%). At 70% samples, it reaches 99.44%, exceeding all strong baselines including ICNN (97.66%) and DRSN (97.08%).

Since the experiments in this paper are multi-class classification tasks with balanced sample sizes across all categories, the variations in Negative Predictive Value (NPV) and specificity are minimal compared to accuracy. Most methods exhibit very similar values for these two metrics, and therefore, NPV and specificity are not included in the reported experimental results. To better reflect the overall classification performance, we focus on metrics that provide more discriminative insights.

To this end, Table 2 also reports Precision and F1-score at $\alpha = 70$ for all methods. Our method achieves the highest scores in all three metrics (Accuracy: 99.44%, Precision: 99.38%, F1-score: 99.34%), with its F1-score being 2.26 percentage points higher than that of DRSN. These results highlight our model's strong detection capability, robustness, and reliability for complex fault diagnosis.

To clearly compare the classification behaviors of different methods, Fig. 5 shows the confusion matrices obtained under the 70% training setting. As illustrated in the figure, samples B1, B2, B3, IR2, and IR3 exhibit higher misclassification tendencies. For example, DRSN mistakenly classified B3 and IR2 samples, while SPD-SSLLE frequently misclassified B2, B3, and IR2—particularly IR2 as B3, and vice versa. Similarly, SPDNet misclassified some IR2 samples as B3 and IR3 samples as OR3. In contrast, our method achieves nearly 100% classification accuracy across all sample categories.

To further assess the effectiveness of our model in feature learning, we examine the distribution of features before and after representation learning. Specifically, we use t-SNE (Maaten and Hinton, 2008) to project both the raw input features and the deep features extracted by the model into a 2D space. As illustrated in Fig. 6, the raw features exhibit significant overlap among different classes, making them hard to distinguish. In contrast, the learned features form compact and well-separated clusters, indicating that our model is capable of learning highly discriminative representations that facilitate accurate fault classification.

4.2.2. *Ottawa dataset experiment*

We conducted a three-class classification experiment using normal, inner race fault, and outer race fault signals from the Ottawa dataset. Unlike the ten-class CWRU dataset, Ottawa's stronger separability makes the three-class task simpler, with most methods achieving good accuracy. The classification accuracies at different training sample sizes, together with Precision and F1-score at $\alpha = 70$, are summarized in Table 3.

As shown in Table 3, the proposed method consistently achieves the highest classification accuracy across all training sample sizes (α), and its performance is robust to changes in sample size (for example, the accuracy is 97.46% when $\alpha = 30$ and 99.63% when $\alpha = 70$, a difference of only 2.17%). Moreover, the Precision and F1-score at $\alpha = 70$ further confirm the superiority of our method: it attains the best results in all three metrics (Accuracy: 99.63%, Precision: 98.61%, F1-score: 99.12%). These results indicate not only high accuracy but also strong generalization and balanced performance across classes.

Table 3
Classification performance (%) of different methods on the Ottawa dataset. Accuracy is reported under different training sample sizes α, while Precision and F1 are computed at $\alpha = 70$.

Method	Accuracy vs. training sample size α					Metrics at $\alpha = 70$	
	30	40	50	60	70	Precision	F1
DAE-SVM	31.59	29.63	31.33	31.67	37.04	36.22	36.63
MD-2DCNN	55.71	61.29	62.44	67.50	78.88	79.73	79.30
BiGRU	71.26	73.14	75.55	80.83	81.48	80.18	80.82
1DCNN	86.34	88.33	89.77	88.05	86.66	87.53	87.09
MCNN-LSTM	90.95	91.29	92.66	93.33	95.18	93.99	94.58
EVGG	94.75	94.91	96.33	96.83	98.27	97.42	97.84
AM-CNN	87.93	89.82	90.66	91.22	93.95	91.52	92.72
ICNN	91.49	93.18	93.67	95.39	96.43	95.49	95.96
DRSN	95.39	96.85	98.44	98.61	99.62	98.27	98.94
GTFENet	97.63	98.33	98.88	99.16	99.25	98.32	98.78
SPD-SSLLE	90.63	95.74	94.44	96.39	97.41	96.29	96.85
SPDNet	93.75	95.11	96.20	96.59	99.21	98.62	98.91
Ours	**97.46**	**98.89**	**99.56**	**99.44**	**99.63**	**98.61**	**99.12**

Table 4
Classification performance (%) of different methods on the PU dataset. Accuracy is reported under different training sample sizes α, while Precision and F1 are computed at $\alpha = 70$.

Method	Accuracy vs. training sample size α					Metrics at $\alpha = 70$	
	30	40	50	60	70	Precision	F1
DAE-SVM	13.10	15.74	17.50	17.01	20.37	22.93	21.57
MD-2DCNN	25.99	27.31	35.27	36.80	44.70	41.92	43.27
BiGRU	29.56	31.25	35.55	36.45	44.90	45.13	45.01
1DCNN	85.51	88.19	90.55	92.01	86.57	82.86	84.67
MCNN-LSTM	69.24	70.37	78.33	84.37	84.72	85.43	85.07
EVGG	85.81	87.88	88.27	90.67	93.39	92.36	92.87
AM-CNN	85.31	87.74	88.42	89.47	91.22	90.88	91.05
ICNN	89.08	90.91	92.49	93.65	95.33	94.28	94.80
DRSN	94.04	94.90	95.00	95.83	96.77	94.22	95.48
GTFENet	93.84	94.67	95.83	96.87	97.23	96.94	97.08
SPD-SSLLE	77.14	80.09	83.61	85.42	86.11	87.51	86.80
SPDNet	80.00	83.41	86.64	88.54	93.22	91.44	92.32
Ours	**93.45**	**95.83**	**96.39**	**97.22**	**97.69**	**97.26**	**97.47**

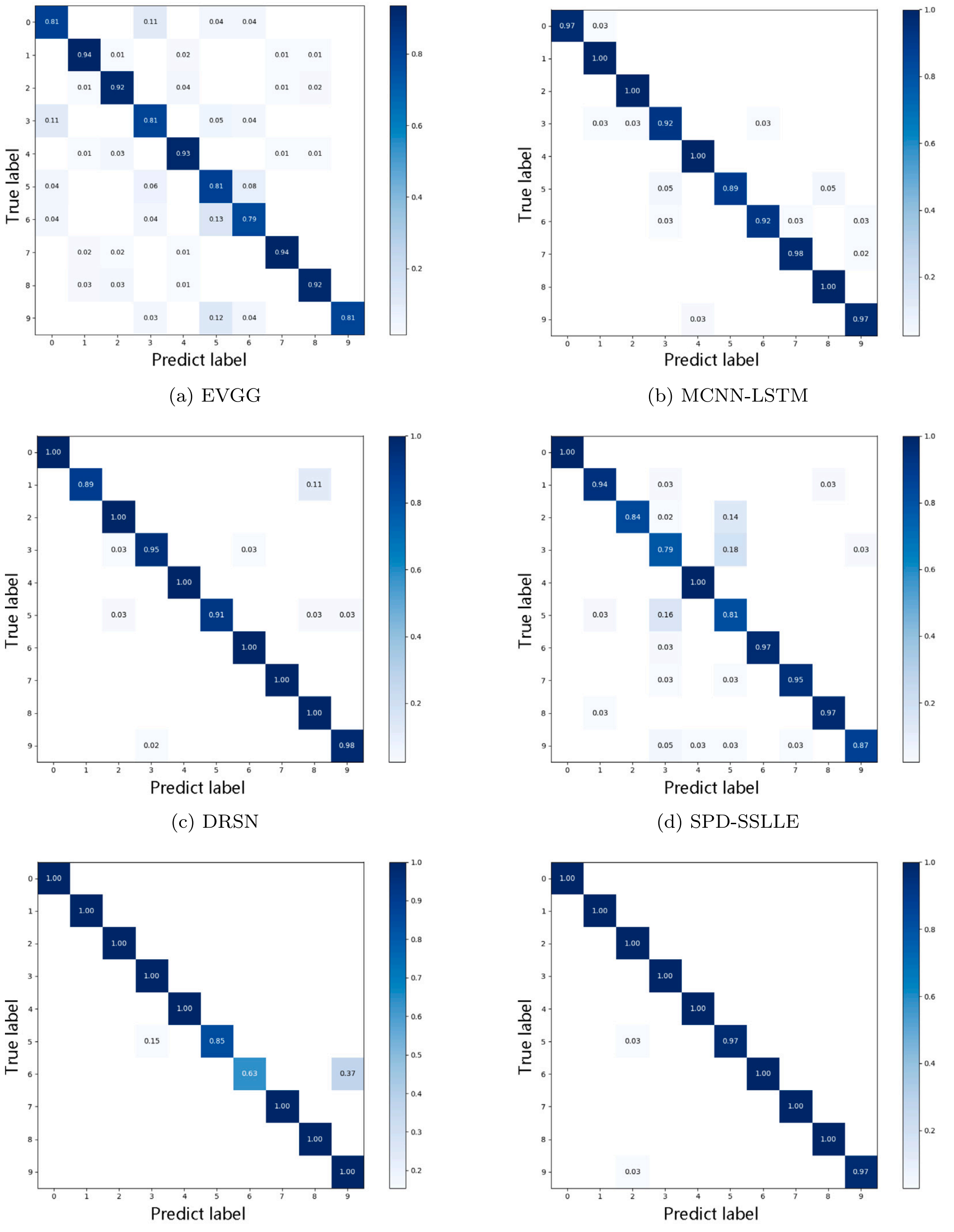
4.2.3. *PU dataset experiment*

The PU dataset has 9 fault modes: normal signal, outer race level 1 EDM damage, outer race level 2 manual engraving damage, outer race level 1 manual engraving damage, outer race level 1 drilling damage, outer race level 2 drilling damage, inner race level 1 EDM damage, inner race level 1 manual engraving damage, and inner race level 2 manual engraving damage. Experimental results under different training sample proportions (30%, 40%, 50%, 60%, and 70%) are summarized in Table 4.

As shown in Table 4, our method demonstrates superior classification accuracy on the PU dataset, achieving 95.83% at $\alpha = 40$ and 97.69% at $\alpha = 70$, both surpassing all competing methods at the corresponding sample sizes. Moreover, Table 4 also reports Precision and F1-score at $\alpha = 70$, where our method attains the best performance across all three metrics (Accuracy: 97.69%, Precision: 97.26%, F1-score: 97.47%), outperforming the next-best method by 0.46 and 0.39 percentage points in Accuracy and F1-score, respectively. These results confirm our network's high accuracy and balanced classification performance across fault categories.

4.2.4. *SCA dataset experiment*

The SCA dataset includes 8 fault modes: Inner Race 1 (IR1), Outer Race 1 (OR1), Inner Race 2 (IR2), Ball (B), Inner Race 3 (IR3), Inner Race 4 (IR4), Outer Race 2 (OR2), and Outer Race 3 (OR3). Since the SCA dataset is derived from industrial machines in real-world settings, it plays a crucial role in evaluating the true performance of the algorithm. The classification accuracies for various training sample



(a) EVGG

(b) MCNN-LSTM

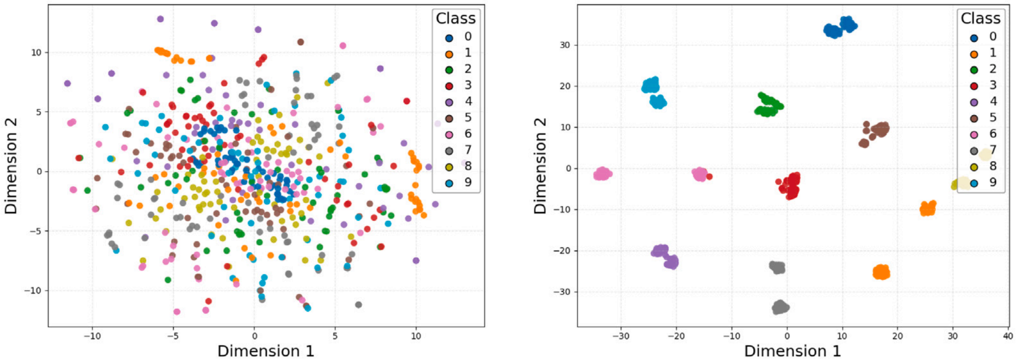
(c) DRSN

(d) SPD-SSLLE

(e) SPDNet

(f) Ours

Fig. 5. Confusion matrix on the CWRU dataset with 70% training sample size. The ten categories shown in the figure are as follows: 0 (NR), 1 (B1), 2 (B2), 3 (B3), 4 (IR1), 5 (IR2), 6 (IR3), 7 (OR1), 8 (OR2), and 9 (OR3). Similar confusion matrices can also be obtained on other datasets, but due to space limitations, they are not presented one by one.



(a) Original input features.

(b) Features extracted by the proposed method.

Fig. 6. t-SNE visualization results of CWRU dataset.

Table 5
Classification performance (%) of different methods on the SCA dataset. Accuracy is reported under different training sample sizes α, while Precision and F1 are computed at $\alpha = 70$.

Method	Accuracy vs. training sample size α					Metrics at $\alpha = 70$	
	30	40	50	60	70	Precision	F1
DAE-SVM	48.57	51.67	52.00	53.12	56.67	53.99	55.30
MD-2DCNN	55.00	67.08	70.00	73.75	78.33	76.98	77.65
BiGRU	56.78	62.91	65.50	67.50	68.33	65.89	67.09
1DCNN	70.35	72.50	73.50	78.12	79.25	76.84	78.03
MCNN-LSTM	80.36	85.42	87.00	91.25	94.17	91.11	90.99
EVGG	81.35	84.14	86.81	90.55	93.60	92.39	92.99
AM-CNN	85.47	88.05	89.44	92.76	94.68	92.27	93.46
ICNN	84.69	86.24	86.93	89.37	90.74	89.86	90.30
DRSN	82.50	84.58	86.00	88.75	90.83	92.21	90.05
GTFENet	93.92	94.58	95.50	96.87	97.50	97.92	97.47
SPD-SSLLE	78.57	81.25	82.50	83.12	85.00	85.60	83.65
SPDNet	85.15	87.94	90.62	93.75	94.79	96.35	95.98
Ours	**97.50**	**96.25**	**97.50**	**98.12**	**98.33**	**98.15**	**98.00**

Table 6
Classification accuracy of different methods on the CWRU dataset under 5 different SNR scenarios.

	−4 dB	0 dB	4 dB	7 dB	10 dB
DAE-SVM	18.64	21.04	24.72	28.59	30.14
BiGRU	29.31	38.77	47.68	58.15	61.44
1DCNN	46.76	63.66	71.37	78.92	82.78
MCNN-LSTM	57.41	66.31	69.14	85.25	89.53
EVGG	76.26	79.31	81.38	83.99	84.06
AM-CNN	84.58	88.77	89.41	89.85	91.72
ICNN	79.75	82.08	86.19	91.73	93.44
DRSN	82.86	86.02	88.56	91.97	93.78
GTFENet	86.73	89.41	91.79	93.68	95.44
SPD-SSLLE	61.44	72.39	76.17	81.92	84.26
SPDNet	64.15	68.16	74.35	83.96	88.03
Ours	**93.46**	**94.86**	**95.69**	**97.56**	**98.23**

sizes, together with Precision and F1-score at $\alpha = 70$, are summarized in Table 5.

As demonstrated in Table 5, our method consistently outperforms all others in classification accuracy, regardless of the training sample size. Even with only 30% of the training data, it reaches 97.50% accuracy, outperforming the second-best method by 3.58% and the third-best method by 12.03%. Moreover, when the training size decreases from 70% to 30%, the accuracy of our model drops by only 0.83%, demonstrating strong robustness to limited training data. At $\alpha = 70$, our model also achieves the highest precision (98.15%) and F1-score (98.00%), with its F1-score exceeding that of SPDNet by 2.02%. It is worth noting that the SCA dataset is collected from real industrial environments, involving diverse bearing types, machine setups, and complex conditions such as variable speeds and fault locations. Despite these challenges, our model maintains superior performance, indicating strong robustness and generalization ability, and showing great potential for practical deployment in industrial fault diagnosis.

4.3. Noise robustness evaluation

To evaluate the robustness of different models under noisy environments, we conducted classification experiments on both the CWRU and SCA datasets. Following common practices in fault diagnosis research, noise was added directly to the clean signals to emulate varying signal-to-noise ratio (SNR) levels (Zhou et al., 2023; Jin et al., 2022; Zhao et al., 2019; Jia et al., 2023). The SNR in decibels is calculated as

$$SNR_{dB} = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right), \tag{19}$$

where P_{signal} and P_{noise} denote the effective power of the signal and the injected noise, respectively.

Gaussian noise on CWRU. On the CWRU dataset, we first consider additive Gaussian white noise, which is commonly used to model broadband background interference in rotating machinery. In our experiments, SNR values ranged from −4 dB to 10 dB, and the classification accuracy of each model under these conditions is reported to assess their noise tolerance.

As shown in Table 6, our method achieves the highest accuracy under all SNR conditions on the CWRU dataset. Even at −4 dB, it maintains 93.46% accuracy, significantly outperforming all baselines. Accuracy increases with higher SNR, reaching 98.23% at 10 dB. In contrast, models such as BiGRU, 1DCNN, and MCNN-LSTM perform poorly under strong noise, indicating limited robustness. Compared to top-performing baselines, our method consistently outperforms DRSN by 10.6% and 4.45% under the same SNR conditions. These results highlight the superior noise resistance and stability of our model on CWRU.

Impulsive Non-Gaussian and mixed noise on SCA. To further investigate robustness under more realistic in-situ noise conditions, we conduct additional experiments on the SCA dataset with non-Gaussian and mixed noise. In structural health monitoring, vibration measurements are often contaminated by impact/impulsive disturbances, $1/f$-type noise, and periodic interference near the mains/line frequency. To emulate such environments, we inject these noise components at the signal level before constructing SPD representations.

Table 7
Classification accuracy of different methods on the SCA dataset under impulsive non-Gaussian and mixed-noise scenarios with varying SNR levels.

Method	Impulsive non-Gaussian noise					Mixed noise				
	−4 dB	0 dB	4 dB	7 dB	10 dB	−4 dB	0 dB	4 dB	7 dB	10 dB
EVGG	81.51	82.16	85.07	88.46	89.84	72.96	75.39	78.97	81.57	82.72
AM-CNN	86.77	87.47	89.12	90.71	92.79	83.32	84.41	86.33	89.75	90.78
ICNN	74.27	76.90	78.89	84.11	86.14	71.21	74.26	79.39	83.39	85.88
DRSN	78.36	81.36	83.07	86.24	86.49	73.87	77.03	80.27	83.91	85.47
GTFENet	83.34	86.88	87.86	91.69	92.32	81.55	84.65	88.94	90.49	92.33
SPDNet	61.87	69.40	75.17	83.31	85.29	61.85	63.24	70.71	78.83	85.47
Ours	**93.08**	**95.41**	**96.14**	**96.49**	**97.40**	**92.55**	**94.52**	**96.22**	**96.57**	**97.85**

Let $x(t)$ denote the original vibration signal and $\tilde{x}(t)$ the noisy signal. Noise is added according to

$$\tilde{x}(t) = x(t) + n(t), \tag{20}$$

where $n(t)$ represents the injected noise. We consider two noise scenarios on the SCA dataset: (i) *impulsive non-Gaussian noise*, where a small proportion of time indices are randomly selected and large-amplitude spikes are superimposed to mimic impact-like transients, leading to a strongly heavy-tailed distribution; and (ii) *mixed noise*, obtained by combining the impulsive component with additional low-frequency $1/f$ and mains/line-frequency harmonic components to simulate harsh field conditions with simultaneous impulsive, low-frequency, and periodic disturbances.

For each noise type, the overall noise level is controlled by the SNR in decibels, computed with the same formula as in the CWRU experiments. We consider SNR values ranging from −4 dB to 10 dB to cover both moderate and severe noise conditions. At each SNR level, noise is injected into the SCA signals, and all methods are evaluated using the same train/test protocol as in the clean-data setting. The classification accuracies on the SCA dataset under impulsive non-Gaussian noise and mixed-noise conditions at different SNR levels are summarized in Table 7.

The results in Table 7 further confirm the robustness of the proposed method under realistic non-Gaussian and mixed-noise conditions. Under impulsive non-Gaussian noise, our approach consistently achieves the highest accuracy at all SNR levels, maintaining 93.08% even at −4 dB and reaching 97.40% at 10 dB, whereas the best Euclidean baselines (AM-CNN and GTFENet) drop to 86.77% and 83.34% at −4 dB, respectively. A similar trend is observed in the mixed-noise scenario: the proposed method retains 92.55% accuracy at −4 dB and exceeds 96% for SNR ≥ 4 dB, while GTFENet and AM-CNN decrease to 81.55% and 83.32% at −4 dB. Across both noise types, the accuracy of competing Euclidean networks degrades more sharply as the SNR decreases, and SPDNet exhibits the weakest robustness, especially under severe mixed noise. These results indicate that the SPD-manifold representation combined with the proposed deep metric learning scheme provides substantially improved robustness against impulsive and mixed disturbances compared with both Euclidean and existing SPD-based baselines.

4.4. Model complexity analysis

To further evaluate the computational efficiency and scalability of the proposed method, we compare the number of Floating-Point Operations (FLOPs) and model parameters under standardized input conditions with several state-of-the-art deep learning approaches. The results are presented in Table 8. Implementation Note: All FLOPs values in Table 8 are calculated based on a single input sample, which consists of a one-dimensional time series with 2048 data points and corresponds to one of 10 possible fault categories. For methods that require non-temporal inputs (e.g., spectrograms or multi-scale features), preprocessing steps strictly follow the procedures outlined in their original implementations. For our method, the FLOPs required to transform the time series into a SPD matrix via phase space reconstruction are also included in the total computation.

Our SPD-based manifold learning framework achieves remarkable efficiency through an ultra-lightweight design. Instead of stacking deep convolutional or residual blocks with many parameters, it uses algebraically efficient SPD operations: BiMap layers, which apply learnable semi-orthogonal projections, and ReEig layers, which perform nonlinear eigenvalue corrections. These layers are implemented as matrix multiplications and eigendecompositions on SPD matrices, resulting in a compact model with low computational cost.

Our model reduces the parameter count by approximately 5 × compared to 1DCNN, and reduces FLOPs by over 50×compared to EVGG. To illustrate why conventional architectures tend to suffer from parameter bloat, consider two typical components: a convolutional layer with 32 input and 64 output channels (3 × 3 kernel) introduces 18,496 parameters, while a ResNet18-style residual block with two such layers totals 73,856 parameters. In contrast, a BiMap layer that transforms a 32 × 32 SPD matrix into a 64 × 64 SPD matrix requires only 512 parameters, originating from a single projection matrix $W \in R^{32\times64}$.

To clarify why traditional deep neural networks are computationally intensive, we compare a BiMap layer from SPD manifold learning with a standard convolution layer. The BiMap layer transforms a 32 × 32 input ($n = 32$) into a 64 × 64 output ($m = 64$) using bilinear mapping $Y = W^{\top}XW$, requiring $2nm(n+m) = 393{,}216$ FLOPs. Its complexity depends only on input and output dimensions: $\mathcal{O}(n^2m)$.

By contrast, a convolution layer with 3 × 3 kernels ($k = 3$), 3 input channels ($c_{\text{in}} = 3$), 32 output channels ($c_{\text{out}} = 32$), and 64 × 64 spatial resolution ($h = w = 64$), requires $2k^2c_{\text{in}}c_{\text{out}}hw = 7{,}077{,}888$ FLOPs. Its complexity, $\mathcal{O}(k^2c_{\text{in}}c_{\text{out}}hw)$, grows rapidly with kernel size, spatial dimensions, and channel count.

Notably, convolutional networks typically expand and compress channels repeatedly (e.g., 32→64→128), which further increases the cost. Both layers in our comparison use standard configurations, yet convolution incurs much higher computational load due to its reliance on spatial and multi-channel processing.

Moreover, the geometric properties of the SPD manifold inherently support scalability in high-throughput systems. For systems with a small number of sensors, the SPD matrix constructed via phase space reconstruction has a dimensionality of $d \times d$, where d is the embedding dimension of the reconstructed space. In contrast, for systems with a large number of sensors, the SPD matrix is derived from the covariance matrix of the raw signals and has a dimensionality of $N \times N$, where N denotes the number of sensors. This distinctive modeling strategy enables the theoretical applicability of SPD matrices to large-scale sensor arrays in industrial monitoring scenarios. Thanks to the extremely low computational complexity of the proposed method, it can not only effectively accommodate the increasing input scale, but also enable near real-time processing of high-throughput data.

Furthermore, the feasibility of deploying this method in real-time industrial monitoring systems, especially on edge devices, is a crucial consideration. Given the low computational cost of the proposed method, with only 6561 model parameters and 0.448M FLOPs per sample, it is well-suited for deployment on resource-constrained edge devices such as embedded systems and IoT devices used in industrial environments. The lightweight design of the model, which relies on efficient SPD manifold operations rather than traditional convolutional layers, allows for real-time processing of continuous sensor streams with minimal resource usage. Moreover, the model's compatibility with parallel hardware such as GPUs and FPGAs further enhances its scalability, making it suitable for large-scale deployment across numerous monitoring stations. This scalability, combined with the model's robustness to noise and minimal latency, ensures that it can be effectively

Table 8
Comparison of FLOPs and parameter counts for a single input sample.

Method	FLOPs	Parameter count
1DCNN	38.27 M	32 330
MCNN-LSTM	103.6 M	95 740
EVGG	22.7 M	246 170
AM-CNN	5956.53 M	1971209
ICNN	85.67 M	187 034
DRSN	345.1 M	5245258
GTFENet	250.13 M	671 434
Ours	0.448 M	6561

Table 9
Comparison of EuM and LEM on CWRU dataset.

Metric	Accuracy	Precision	F1	Time
EuM	99.44	99.38	99.34	1.28 s
LEM	99.62	98.49	99.41	1.93 s

integrated into existing embedded systems for continuous, real-time fault detection and diagnostics in industrial settings.

4.5. Interpretability

To understand how the SPD manifold deep network learns interpretable features, we analyze the structural evolution of the SPD matrices within the model. As described in Section 3.1, each SPD matrix P encodes the statistical properties of the reconstructed phase space. Critically, the diagonal elements P_{ii} represent the signal energy (autocorrelation) at their respective time-delay scales τ_i, while the off-diagonal elements $P_{ij}\,(i \neq j)$ characterize the dynamic coupling (cross-correlation) between different scales τ_i and τ_j. Therefore, tracking how the relative prominence of the diagonal versus off-diagonal elements changes across network layers provides direct insight into how the network transforms and refines the feature representation.

To visually illustrate how these structural characteristics evolve through the network, Fig. 7 presents the SPD matrices at different BiMap layers for two example samples. As shown in Fig. 7, subfigures (a) and (e) represent the SPD matrices constructed directly from two raw OR1 samples. Subfigures (b), (c), and (d), as well as (f), (g), and (h), illustrate the corresponding outputs after the first, second, and third BiMap layers, respectively. It can be observed that, as the network goes deeper, the contrast between the diagonal and off-diagonal elements becomes increasingly pronounced. This indicates that the sum of diagonal elements – representing the total energy at each time-delay dimension – gradually increases through each BiMap layer. In other words, the energy of the entire SPD matrix becomes more concentrated along the diagonal as the network deepens. This pattern highlights that the SPD manifold-based deep network progressively enhances the representation of temporal self-correlations while suppressing cross-scale interactions, thereby making the learned features more discriminative and interpretable.

4.6. Ablation studies

4.6.1. EuM vs. LEM reconstruction loss

To further investigate the impact of using EuM vs. LEM in the reconstruction loss on both classification performance and training efficiency, we conducted a comparison experiment on the CWRU dataset. As shown in Table 9, using LEM slightly enhances classification accuracy. Fig. 8 displays the loss convergence curves for EuM vs. LEM on the CWRU dataset. Both curves follow a similar trend, with the loss steadily decreasing over epochs, indicating that both EuM and LEM effectively reduce reconstruction loss. However, due to the significantly larger size of the original SPD matrices and the model-reconstructed SPD matrices compared to the hidden layer's SPD matrices, using LEM in the reconstruction loss incurs a higher computational cost. Specifically, although the matrix logarithmic operations involved in LEM are conceptually simple, their computational cost becomes more significant when applied repeatedly to large SPD matrices. The increased complexity of these operations during backpropagation results in a higher overall computational burden. Therefore, while the training time per epoch is approximately 1.28 s when using EuM, it rises to about 1.93 s per epoch with LEM. Therefore, we opted to use EuM in the reconstruction loss, as it provides a better trade-off between computational efficiency and classification performance.

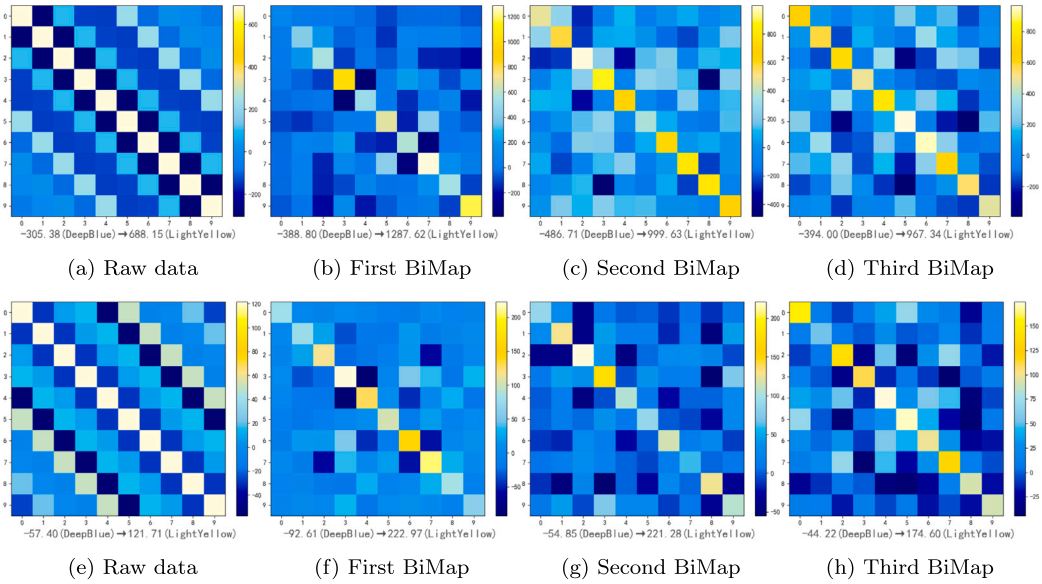
4.6.2. Effect of network components

To evaluate the contribution of each component in the proposed network, we conducted an ablation study testing the effects of SPD Denoising (SD), multi-class N-pair loss (NL), and sparse penalty (SP). We tested each component individually to understand its impact: SPD Denoising reduces noise and captures key features; N-pair loss improves the model's ability to distinguish between classes by optimizing sample distances; and the sparse penalty encourages the network to focus on important features by penalizing irrelevant activations. The full model integrates all three components, and we compared their effects using classification accuracy, confidence intervals, and statistical significance based on paired t-tests. The experiments were performed on the CWRU dataset with varying training sample sizes (α = 30, 40, 50, 60, 70). The classification accuracy with 95% confidence intervals and the corresponding p-values for statistical significance are reported in Table 10 and Table 11, respectively. The 95% confidence intervals for classification accuracy were calculated based on the results of five independent runs of the network. In each run, the network was trained on a different random split of the CWRU dataset, and the classification accuracy was averaged across these runs. Similarly, the paired t-test was performed on the results of these five independent runs, ensuring that the p-values reflect statistical significance across different training data splits.

As shown in Table 10, all three components – SPD denoising (SD), multi-class N-pair loss (NL), and sparse penalty (SP) – contribute positively to the model's performance. When only SD is used (baseline), the accuracy at $\alpha = 30$ is 74.07 ± 1.29%. Adding SP increases the accuracy at $\alpha = 30$ to 86.61 ± 1.05%, while introducing NL produces an even larger improvement, reaching 92.36 ± 2.36%. The full model that integrates SD, NL, and SP achieves the highest accuracy of 95.85 ± 0.89% at $\alpha = 30$, demonstrating that the three components provide complementary benefits when combined.

Table 11 reports the corresponding p-values from paired t-tests. All comparisons with the baseline (SD only) show extremely significant improvements across all training sizes. Because the baseline accuracy is substantially lower (74.07%–86.75%), the performance gaps are large, resulting in very small p-values for A1→A4, A1→A5, and A1→A7. The full model also shows significant gains over its two-component variants (SD+SP and SD+NL) under most settings. At $\alpha = 30$, for example, the improvements over SD+NL and SD+SP remain statistically significant, with p-values of 0.004283 and 0.0000033, respectively. As the amount of training data increases, the performance gap between strong configurations naturally narrows. For instance, the difference between SD+NL and the full model at $\alpha = 70$ is only 0.16 percentage points, yielding a non-significant p-value of 0.312458. This behavior is expected because both A4 and A7 already achieve near-saturated performance at large training sizes, and the remaining accuracy difference lies within normal statistical variation, which naturally leads to higher p-values.

These results demonstrate that (1) each module independently improves performance, (2) coupling two modules provides further benefits, and (3) the full triad consistently delivers the best overall results. The stable 95% confidence intervals across all settings, along with



(a) Raw data (b) First BiMap (c) Second BiMap (d) Third BiMap

(e) Raw data (f) First BiMap (g) Second BiMap (h) Third BiMap

Fig. 7. Evolution of SPD matrices for two OR1 samples. Each row corresponds to one sample, showing its SPD matrix at four stages: (a, e) raw data, (b, f) after the first BiMap layer, (c, g) after the second BiMap layer, and (d, h) after the third BiMap layer. As the SPD representations evolve through the network, a clear shift in the matrix structure is observed. In particular, the energy becomes increasingly concentrated along the diagonal, indicating stronger temporal self-correlations and suppressed cross-scale interactions.

Table 10
Classification accuracy (mean ± 95% confidence interval, in %) of ablation experiments across different training sample sizes α on the CWRU dataset.

ID	SD	NL	SP	30	40	50	60	70
A1	Yes	No	No	74.07 ± 1.29	77.82 ± 1.21	79.71 ± 1.19	83.89 ± 1.10	86.75 ± 1.43
A2	No	Yes	No	90.96 ± 1.19	91.55 ± 1.35	93.63 ± 1.30	94.88 ± 0.91	96.61 ± 1.02
A3	No	No	Yes	82.57 ± 1.39	86.22 ± 1.28	88.86 ± 0.91	91.82 ± 0.70	94.05 ± 1.06
A4	Yes	Yes	No	92.36 ± 2.36	93.97 ± 1.10	96.19 ± 0.80	97.28 ± 0.92	99.07 ± 0.23
A5	Yes	No	Yes	86.61 ± 1.05	89.18 ± 1.20	92.24 ± 1.13	97.53 ± 0.97	97.47 ± 0.67
A6	No	Yes	Yes	94.24 ± 0.97	95.26 ± 0.73	97.19 ± 0.57	98.28 ± 0.45	99.15 ± 0.34
A7	Yes	Yes	Yes	95.85 ± 0.89	97.55 ± 0.73	98.13 ± 0.71	98.53 ± 0.90	99.23 ± 0.57

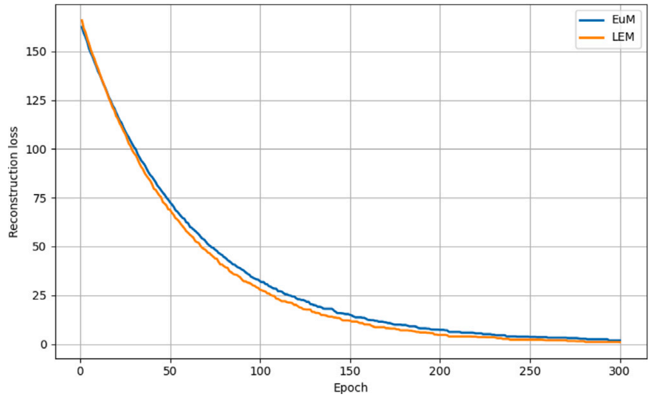


Fig. 8. Reconstruction Loss convergence curves for EuM and LEM on the CWRU dataset.

Table 11
p-values of paired t-tests comparing key ablation configurations.

Comparison	30	40	50	60	70
A1 vs A4	0.0000023	0.0000001	0.0000001	0.0000003	0.0000017
A1 vs A5	0.0000007	0.0000008	0.000002	0.0000001	0.0000007
A1 vs A7	0.0000001	0.0000002	0.0000007	0.0000008	0.0000007
A4 vs A7	0.004283	0.003874	0.021347	0.047612	0.312458
A5 vs A7	0.0000033	0.0000027	0.0000092	0.041387	0.006473

the observed statistical significance patterns, indicate that the proposed components enhance not only accuracy but also the model's generalization capability without inducing overfitting.

4.6.3. Effect of classifier choice (SVM vs. Linear Head)

To investigate the impact of the classifier choice, we compare the proposed SPD feature extractor followed by an SVM with an end-to-end variant that replaces the SVM by a Linear Head trained with cross-entropy loss. The results on the CWRU dataset under different training ratios are summarized in Table 12. Across all five training ratios (30%–70%), the SVM consistently outperforms the Linear Head. The performance gap is most pronounced in the low-data regime: with only 30% of the data for training, the SVM achieves 95.85 ± 0.89% accuracy, whereas the Linear Head attains 87.82 ± 1.20%; at a 40% ratio, the accuracies are 97.55 ± 0.73% and 91.63 ± 0.98%, respectively. As the training ratio increases, the gap gradually narrows, but the SVM still maintains a slight advantage even at 70% of the data (99.23 ± 0.57% vs. 98.35 ± 0.61%). Paired t-tests yield p-values in the range $[8.0 \times 10^{-7}, 8.3 \times 10^{-4}]$, indicating that the improvements brought by the SVM are statistically significant under all training ratios.

These observations suggest that the margin-maximization property of SVM leads to more robust and discriminative SPD embeddings, particularly in small-sample settings, whereas jointly optimizing a Linear Head with cross-entropy introduces an additional objective and extra hyperparameters that can slightly weaken the class separability of the learned representations. From an optimization perspective, the SPD multi-class N-pair loss explicitly enforce large inter-class margins and compact intra-class clusters in the SPD embedding space, while the

Table 12
Comparison of SVM and linear head classifiers under different training ratios on the CWRU dataset. Classification accuracy is reported as mean±std (%), and the last row lists the p-values of paired t-tests between SVM and Linear Head.

Method	Training ratio (%)				
	30	40	50	60	70
SVM	95.85 ± 0.89	97.55 ± 0.73	98.13 ± 0.71	98.53 ± 0.90	99.23 ± 0.57
Linear head	87.82 ± 1.20	91.63 ± 0.98	94.10 ± 1.18	96.70 ± 1.09	98.35 ± 0.61
p-value	0.0000017	0.0000008	0.0000943	0.0001580	0.0008296

cross-entropy loss attached to the Linear Head only constrains the final decision boundaries and does not directly regularize pairwise relations between embeddings. When these heterogeneous objectives are optimized simultaneously, the gradients induced by cross-entropy may partially conflict with those of the metric-learning losses, leading to embeddings that are sufficient for the Linear Head but less well structured in terms of global class separability, especially under reduced training data.

4.7. Analysis

From this study, the following conclusions can be drawn:

1. The proposed SPD manifold-based deep metric learning method demonstrates excellent classification performance on various bearing fault datasets. It achieves 96.43% accuracy on the CWRU dataset with only 30% training data, and 98.33% on the challenging SCA dataset collected with variable speeds and strong noise. These results reflect its strong generalization ability in both controlled and real-world industrial environments.
2. As demonstrated in Section 4.3, the proposed method maintains high reliability under severe noise conditions. At −4 dB SNR, simulating harsh industrial noise, it achieves 93.46% accuracy—substantially outperforming noise-sensitive models such as MCNN-LSTM (57.41%) and 1DCNN (46.76%). Even at moderate noise levels (4 dB SNR), it reaches 95.69% accuracy, showing a 6.28%–9.5% improvement over AM-CNN, ICNN, and DRSN. This robustness can be attributed to the SPD sparse denoising autoencoder, which effectively suppresses noise disturbances through reconstruction.
3. As detailed in Section 4.4, the proposed method offers higher computational efficiency than conventional deep networks while using minimal resources. With only 6561 model parameters (just 3.5% of ICNN's 187K) and 0.448M FLOPs per sample (1.97% of EVGG's 22.7M), it replaces computationally intensive convolutional layers with efficient SPD manifold operations. The semi-orthogonal projection of the BiMap layer and the ReEig operation contribute to the lightweight and scalable deployment on edge devices for real-time monitoring.

5. Conclusion

This paper presents a bearing fault diagnosis method based on SPD manifold deep metric learning, which mainly includes two steps. First, a Riemannian sparse denoising autoencoder is developed, which takes noisy SPD matrices as input and generates embedded representations and reconstructed SPD matrices. By optimizing the loss function to minimize reconstruction error, multi-class N-pair loss, and sparse penalty terms, the model enhances the clustering of similar samples and the separation of dissimilar samples, while improving robustness to vibration signals. Then, SVM is applied in the embedding space for classification. Experimental results confirm that modeling signal data on Riemannian manifolds yields superior performance compared to conventional Euclidean approaches. Moreover, the proposed model achieves better results than other Riemannian manifold-based methods, highlighting its effectiveness and robustness.

While this approach demonstrates promising results, it also has several limitations inherent to supervised learning models. One key limitation is the dependence on large, balanced, and accurately labeled datasets, which can be challenging to obtain in real-world industrial applications, where fault data may be scarce, noisy, or costly to collect. Additionally, supervised models often struggle with generalization across varying operating conditions, unseen fault types, and different machines, making them less adaptable to diverse scenarios. To address these limitations, future work could explore the integration of semi-supervised or unsupervised domain adaptation strategies.

Recent studies have explored the application of these domain adaptation algorithms in fault diagnosis (Li et al., 2025a; Zhao et al., 2025). Furthermore, some studies have explored the potential of unsupervised learning methods for damage detection, such as the combination of deep autoencoders and one-class support vector machines (Wang and Cha, 2021). Wang and Cha (2022) present and develop several unsupervised novelty detection methods, analyzing their performance in structural damage detection. These studies highlight the potential of both unsupervised and semi-supervised methods. Specifically, combining semi-supervised domain adaptation methods could help address the scarcity of fault data, while unsupervised domain adaptation techniques could enhance the model's generalization across different operating conditions and machines, especially when the model encounters unseen fault types. By focusing on these future directions, we aim to improve the robustness and versatility of the model, enabling it to perform efficiently across a wider range of real-world scenarios.

CRediT authorship contribution statement

Junshi Cheng: Writing – original draft, Software, Methodology, Formal analysis, Data curation. **Ruisheng Ran:** Supervision, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Bin Fang:** Writing – review & editing, Funding acquisition, Formal analysis. **Benchao Li:** Writing – review & editing, Formal analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Bin Fang reports financial support was provided by the National Natural Science Foundation of China. Ruisheng Ran reports financial support was provided by Chongqing Municipal Education Commission. Ruisheng Ran reports financial support was provided by Ministry of Education of the People's Republic of China. Bin Fang reports financial support was provided by the Natural Science Foundation of Chongqing. Bin Fang reports financial support was provided by the Key Special Project for Technological Innovation and Application Development in Chongqing. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China 62376042, the Key Project of Chongqing Municipal Education Commission KJZD-K202100505, the Project of the

Ministry of Education in China 20YJAZH084, the Natural Science Foundation of Chongqing CSTB2024NSCQ-KJFZZDX0036, the Key Special Project for Technological Innovation and Application Development in Chongqing CSTB2023TIAD-KPX0050 & CSTB2022TIAD-KPX0176.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2026.113821.

Data availability

Data will be made available on request.

References

Althubaiti, A., Elasha, F., Teixeira, J.A., 2022. Fault diagnosis and health management of bearings in rotating equipment based on vibration analysis–a review. J. Vibroeng. 24 (1), 46–74. http://dx.doi.org/10.21595/jve.2021.22100.

Arsigny, V., Fillard, P., Pennec, X., Ayache, N., 2007. Geometric means in a novel vector space structure on symmetric positive-definite matrices. SIAM J. Matrix Anal. Appl. 29 (1), 328–347. http://dx.doi.org/10.1137/050637996.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press, URL https://bida.uclv.edu.cu/bitstream/handle/123456789/17293/bv_cvxbook.pdf?sequence=1&isAllowed=y.

Cao, L., 1997. Practical method for determining the minimum embedding dimension of a scalar time series. Phys. D: Nonlinear Phenom. 110 (1–2), 43–50. http://dx.doi.org/10.1016/S0167-2789(97)00118-8.

Chen, X., Zhang, B., Gao, D., 2021. Bearing fault diagnosis base on multi-scale CNN and LSTM model. J. Intell. Manuf. 32 (4), 971–987. http://dx.doi.org/10.1007/s10845-020-01600-2.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27. http://dx.doi.org/10.1109/TIT.1967.1053964.

Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S., 2007. Information-theoretic metric learning. In: Proceedings of the 24th International Conference on Machine Learning. pp. 209–216. http://dx.doi.org/10.1145/1273496.1273523.

Deng, W., Shang, S., Zhang, L., Lin, Y., Huang, C., Zhao, H., Ran, X., Zhou, X., Chen, H., 2025. Multi-strategy quantum differential evolution algorithm with cooperative co-evolution and hybrid search for capacitated vehicle routing. IEEE Trans. Intell. Transp. Syst. 26 (11), 18460–18470. http://dx.doi.org/10.1109/TITS.2025.3594782.

Diepeveen, W., 2024. Pulling back symmetric Riemannian geometry for data analysis. http://dx.doi.org/10.48550/arXiv.2403.06612, arXiv preprint arXiv:2403.06612.

Entezami, A., Sarmadi, H., Behkamal, B., Mariani, S., 2025. Early warning of structural damage via manifold learning-aided data clustering and non-parametric probabilistic anomaly detection. Mech. Syst. Signal Process. 224, 111984. http://dx.doi.org/10.1016/j.ymssp.2024.111984.

Eren, L., 2017. Bearing fault detection by one-dimensional convolutional neural networks. Math. Probl. Eng. 2017 (1), 8617315. http://dx.doi.org/10.1109/TIT.1967.1053964.

Fraser, A.M., Swinney, H.L., 1986. Independent coordinates for strange attractors from mutual information. Phys. Rev. A 33 (2), 1134. http://dx.doi.org/10.1103/PhysRevA.33.1134.

Gu, J., Peng, Y., Lu, H., Chang, X., Chen, G., 2022. A novel fault diagnosis method of rotating machinery via VMD, CWT and improved CNN. Measurement 200, 111635. http://dx.doi.org/10.1016/j.measurement.2022.111635.

Gu, Y.-K., Zhou, X.-Q., Yu, D.-P., Shen, Y.-J., 2018. Fault diagnosis method of rolling bearing using principal component analysis and support vector machine. J. Mech. Sci. Technol. 32, 5079–5088. http://dx.doi.org/10.1007/s12206-018-1004-0.

Huang, H., Baddour, N., 2018. Bearing vibration data collected under time-varying rotational speed conditions. Data Brief 21, 1745–1749. http://dx.doi.org/10.1016/j.dib.2018.11.019.

Huang, Z., Van Gool, L., 2017. A riemannian network for spd matrix learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, (1), http://dx.doi.org/10.1609/aaai.v31i1.10866.

Huang, K., Wu, S., Sun, B., Yang, C., Gui, W., 2022. Metric learning-based fault diagnosis and anomaly detection for industrial data with intraclass variance. IEEE Trans. Neural Netw. Learn. Syst. 35 (1), 547–558. http://dx.doi.org/10.1109/TNNLS.2022.3175888.

Jia, L., Chow, T.W., Yuan, Y., 2023. GTFE-net: A gramian time frequency enhancement CNN for bearing fault diagnosis. Eng. Appl. Artif. Intell. 119, 105794. http://dx.doi.org/10.1016/j.engappai.2022.105794.

Jin, Y., Qin, C., Zhang, Z., Tao, J., Liu, C., 2022. A multi-scale convolutional neural network for bearing compound fault diagnosis under various noise conditions. Sci. China Technol. Sci. 65 (11), 2551–2563. http://dx.doi.org/10.1007/s11431-022-2109-4.

Kim, Y., 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Lessmeier, C., Kimotho, J.K., Zimmer, D., Sextro, W., 2016. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In: PHM Society European Conference, vol. 3, (1), http://dx.doi.org/10.36001/phme.2016.v3i1.1577.

Li, W., Chen, Z., He, G., 2020. A novel weighted adversarial transfer network for partial domain fault diagnosis of machinery. IEEE Trans. Ind. Inform. 17 (3), 1753–1762. http://dx.doi.org/10.1109/TII.2020.2994621.

Li, J., Deng, W., Dang, X., Zhao, H., 2025a. Cross-domain adaptation fault diagnosis with maximum classifier discrepancy and deep feature alignment under variable working conditions. IEEE Trans. Reliab. 74 (3), 4106–4115. http://dx.doi.org/10.1109/TR.2025.3551155.

Li, J., Deng, W., Ding, J., Zhao, H., 2025b. IBN-MixStyle network with dynamic weighted invariant risk minimization for domain-generalized bearing fault diagnosis. IEEE Trans. Consum. Electron. http://dx.doi.org/10.1109/TCE.2025.3607134, 1–1.

Li, H., Lian, X., Guo, C., Zhao, P., 2015. Investigation on early fault classification for rolling element bearing based on the optimal frequency band determination. J. Intell. Manuf. 26 (1), 189–198. http://dx.doi.org/10.1007/s10845-013-0772-8.

Li, X., Zhong, X., Shao, H., Han, T., Shen, C., 2021. Multi-sensor gearbox fault diagnosis by using feature-fusion covariance matrix and multi-Riemannian kernel ridge regression. Reliab. Eng. Syst. Saf. 216, 108018. http://dx.doi.org/10.1016/j.ress.2021.108018.

Liu, Y., Hu, Z., Zhang, Y., 2022. Symmetric positive definite manifold learning and its application in fault diagnosis. Neural Netw. 147, 163–174. http://dx.doi.org/10.1016/j.neunet.2021.12.013.

Lundström, A., O'Nils, M., 2023. Factory-based vibration data for bearing-fault detection. Data 8 (7), 115. http://dx.doi.org/10.3390/data8070115.

Maaten, L.v.d., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605, URL https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

Moysidis, D.A., Karatzinis, G.D., Boutalis, Y.S., Karnavas, Y.L., 2023. A study of noise effect in electrical machines bearing fault detection and diagnosis considering different representative feature models. Machines 11 (11), 1029. http://dx.doi.org/10.3390/machines11111029.

Nesterov, Y., 1983. A method for unconstrained convex minimization problem with the rate of convergence O (1/k2). In: Dokl. Akad. Nauk. SSSR, vol. 269, (3), p. 543, URL https://cir.nii.ac.jp/crid/1370576118744597902.

Peng, Z., Li, J., Hao, H., 2022. Structural damage detection via phase space based manifold learning under changing environmental and operational conditions. Eng. Struct. 263, 114420. http://dx.doi.org/10.1016/j.engstruct.2022.114420.

Pennec, X., Fillard, P., Ayache, N., 2006. A Riemannian framework for tensor computing. Int. J. Comput. Vis. 66, 41–66. http://dx.doi.org/10.1007/s11263-005-3222-z.

Sarmadi, H., Entezami, A., Behkamal, B., De Michele, C., 2022. Partially online damage detection using long-term modal data under severe environmental effects by unsupervised feature selection and local metric learning. J. Civ. Struct. Health Monit. 12 (5), 1043–1066. http://dx.doi.org/10.1007/s13349-022-00596-y.

Shuang, L., Meng, L., 2007. Bearing fault diagnosis based on PCA and SVM. In: 2007 International Conference on Mechatronics and Automation. IEEE, pp. 3503–3507. http://dx.doi.org/10.1109/ICMA.2007.4304127.

Smith, W.A., Randall, R.B., 2015. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. Mech. Syst. Signal Process. 64, 100–131. http://dx.doi.org/10.1016/j.ymssp.2015.04.021.

Sohn, K., 2016. Improved deep metric learning with multi-class N-pair loss objective. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (Eds.), In: Advances in Neural Information Processing Systems, vol. 29, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf.

Sra, S., Hosseini, R., 2013. Geometric optimisation on positive definite matrices for elliptically contoured distributions. In: Advances in Neural Information Processing Systems, vol. 26, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2013/file/3948ead63a9f2944218de038d8934305-Paper.pdf.

Suh, Y.-J., Kim, B.H., 2021. Riemannian embedding banks for common spatial patterns with EEG-based SPD neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (1), pp. 854–862. http://dx.doi.org/10.1609/aaai.v35i1.16168.

Takens, F., 2006. Detecting strange attractors in turbulence. In: Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held At the University of Warwick 1979/80. Springer, pp. 366–381, URL https://link.springer.com/content/pdf/10.1007/BFb0091924.pdf.

Thorsen, O.V., Dalva, M., 2002. Failure identification and analysis for high-voltage induction motors in the petrochemical industry. IEEE Trans. Ind. Appl. 35 (4), 810–818. http://dx.doi.org/10.1109/28.777188.

Torzoni, M., Manzoni, A., Mariani, S., 2022. Structural health monitoring of civil structures: A diagnostic framework powered by deep metric learning. Comput. Struct. 271, 106858. http://dx.doi.org/10.1016/j.compstruc.2022.106858.

Van, M., Kang, H.-J., 2015. Bearing defect classification based on individual wavelet local fisher discriminant analysis with particle swarm optimization. IEEE Trans. Ind. Inform. 12 (1), 124–135. http://dx.doi.org/10.1109/TII.2015.2500098.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. pp. 1096–1103, https://dl.acm.org/doi/abs/10.1145/1390156.1390294.

Wang, Z., Cha, Y.-J., 2021. Unsupervised deep learning approach using a deep autoencoder with a one-class support vector machine to detect damage. Struct. Health Monit. 20 (1), 406–425. http://dx.doi.org/10.1177/1475921720934051.

Wang, Z., Cha, Y.-J., 2022. Unsupervised machine and deep learning methods for structural damage detection: a comparative study. Eng. Rep. 7 (1), e12551. http://dx.doi.org/10.1002/eng2.12551.

Wang, R., Wu, X.-J., Xu, T., Hu, C., Kittler, J., 2023. U-spdnet: An SPD manifold learning-based neural network for visual classification. Neural Netw. 161, 382–396. http://dx.doi.org/10.1016/j.neunet.2022.11.030.

Ye, S., Zhang, F., Gao, F., Zhou, Z., Yang, Y., 2022. Fault diagnosis for multilevel converters based on an affine-invariant riemannian metric autoencoder. IEEE Trans. Ind. Inform. 19 (3), 2619–2628. http://dx.doi.org/10.1109/TII.2022.3186992.

Zhang, H., Sra, S., 2016. First-order methods for geodesically convex optimization. In: Conference on Learning Theory. PMLR, pp. 1617–1638, URL https://proceedings.mlr.press/v49/zhang16b.html.

Zhao, H., Liu, C., Dang, X., Xu, J., Deng, W., 2025. Few-shot cross-domain fault diagnosis of transportation motor bearings using MAML-GA. IEEE Trans. Transp. Electrification http://dx.doi.org/10.1109/TTE.2025.3625779, 1–1.

Zhao, M., Zhong, S., Fu, X., Tang, B., Pecht, M., 2019. Deep residual shrinkage networks for fault diagnosis. IEEE Trans. Ind. Inform. 16 (7), 4681–4690. http://dx.doi.org/10.1109/TII.2019.2943898.

Zhou, H., Chen, W., Cheng, L., Williams, D., De Silva, C.W., Xia, M., 2023. Reliable and intelligent fault diagnosis with evidential VGG neural networks. IEEE Trans. Instrum. Meas. 72, 1–12. http://dx.doi.org/10.1109/TIM.2023.3250308.

Zhu, Z., Lei, Y., Qi, G., Chai, Y., Mazur, N., An, Y., Huang, X., 2023. A review of the application of deep learning in intelligent fault diagnosis of rotating machinery. Measurement 206, 112346. http://dx.doi.org/10.1016/j.measurement.2022.112346.