This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3635151

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2025.xxxxxxx

Locally Enhanced Multi-view Discriminant Analysis: Preserving Neighborhood Structure for Improved Cross-View Classification

ELAHEH MOTAMEDI[1], (Student Member, IEEE), and MILAD SIAMI[2] (Senior Member, IEEE)
[1]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA (e-mail: motamedi.e@northeastern.edu)
[2]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA (e-mail: m.siami@northeastern.edu)

ABSTRACT Multi-view discriminant analysis (MvDA) achieves cross-view recognition by maximizing global class separation, but fails when class boundaries are closely positioned– a critical limitation for real-world applications like distinguishing similar facial expressions or robotic trajectories. We propose Locally Enhanced Multi-view Discriminant Analysis (LE-MvDA), which achieves up to 97.7% accuracy on RoboMNIST (6.3% improvement over state-of-the-art) by fundamentally reimagining how discriminative subspaces preserve local structure. Unlike MvDA's homogeneous treatment of data, LE-MvDA introduces a supervised affinity matrix with adaptive local scaling factors ($\sigma_k = \|x_k - x_k^{(K)}\|$) that dynamically adjusts to neighborhood density, ensuring samples from the same class remain cohesive while maximizing inter-class margins. This class-aware locality preservation is integrated directly into the discriminative objective through a unified generalized eigenvalue problem, eliminating the traditional trade-off between discrimination and structure preservation. Extensive experiments demonstrate LE-MvDA's superiority: 92.8% on Multi-PIE (3.4% improvement), 91.9% on ORL (9.9% improvement), and competitive performance against deep learning models (97.7% vs 98.1% for TimeSformer) while requiring 100× fewer parameters (30K vs 3M+ for TimeSformer) and achieving 40× faster inference (0.04ms vs 1.64ms for TimeSformer). LE-MvDA is particularly valuable for resource-constrained deployments requiring both high accuracy and interpretability. Its explicit projection matrices enable direct feature importance analysis, contrasting with the black-box nature of deep learning alternatives.Nonetheless, deep learning remains advantageous for large-scale datasets or tasks requiring end-to-end representation learning.

INDEX TERMS Affinity matrix, cross-view recognition, dimensionality reduction, LE-MvDA, locality preserving projection, multi-view classification, multi-view discriminant analysis, multi-view subspace learning, neighborhood preservation, robot action recognition.

I. INTRODUCTION

Multi-view learning faces a fundamental challenge when existing methods like Multi-view Discriminant Analysis (MvDA) fail to handle closely positioned class boundaries in feature space, a common scenario in real-world applications [1]. Such difficulties arise in tasks like distinguishing between similar facial expressions (e.g., neutral vs. slight smile) or robotic trajectories for drawing similar digits (e.g., 0 vs. 8), where patterns share curved shapes but differ in subtle execution details. These cases expose the critical limitation of global optimization approaches that treat all inter-class relationships uniformly. Multi-view learning [2]–[5] has thus become a central research domain, motivated by the complementary nature of multiple perspectives: while individual views capture partial characteristics, integrating them yields a more comprehensive representation. However, this richness comes at the cost of higher dimensionality, leading to the 'curse of dimensionality'—reduced efficiency, increased model complexity, and sparsity. Dimensionality reduction techniques address this by projecting multi-view data into compact, discriminative subspaces.

Classical single-view dimensionality reduction methods—including principal component analysis (PCA) [6], linear discriminant analysis (LDA) [7], and locality preserving projections (LPP) [8]—established foundations by maximizing variance, discriminative separation, or local structure preservation. Recent extensions of these techniques using two-dimensional formulations [9], [10] have further improved recognition performance. Extending these ideas, multi-view subspace learning (MvSL) methods—from early approaches like Canonical Correlation Analysis (CCA) [11] and Multi-view CCA (MCCA) [12] to recent advances [13]–[17]—align diverse views into shared subspaces.

Supervised variants—including Generalized Multi-view Analysis (GMA) [18] and Multi-view Discriminant Analysis (MvDA) [2]—further exploit label information for classification, with MvDA emerging as a leading approach due to its joint optimization of view alignment and class separability. Yet, its global centroid-based separation proves inadequate for classes that are inherently close or overlapping, as it ignores crucial fine-grained neighborhood structures. Concurrently, while LPP preserves local geometry, its unsupervised formulation leaves inter-class overlaps unresolved. To address these complementary shortcomings, we propose Locally Enhanced Multi-view Discriminant Analysis (LE-MvDA). Unlike conventional approaches that separately address discriminative learning and local structure preservation, LE-MvDA introduces an adaptive affinity matrix with local scaling factors ($\sigma_k = \|x_k - x_k^{(K)}\|$ [19]) that dynamically adjusts to neighborhood density within each class. This unified formulation, solved through a single generalized eigenvalue problem, preserves intrinsic neighborhood structures in the learned subspace while maintaining strong discriminative power. By strategically maximizing between-class variations across both inter- and intra-view perspectives while concurrently minimizing within-class variations, LE-MvDA eliminates the traditional trade-off between local and global objectives, achieving clearer class boundaries and improved classification performance. Our main contributions can be summarized as follows:

- **Adaptive Local Discrimination:** We introduce a supervised affinity matrix with adaptive local scaling, achieving up to 9.9% accuracy improvement over the strongest baseline by better separating closely positioned classes, while naturally handling heterogeneous class distributions.
- **Unified Mathematical Framework:** LE-MvDA embeds locality preservation into the discriminative objective within a single generalized eigenvalue problem, simultaneously optimizing neighborhood preservation and class separation in a unified formulation.
- **Superior Performance Across Domains:** LE-MvDA consistently outperforms classical multi-view methods, achieving up to 9.9% accuracy gain on benchmark datasets. It also achieves accuracy within 0.4% of TimeSformer while requiring 100× fewer parameters, underscoring its efficiency.
- **Interpretability and Efficiency:** Unlike black-box deep models, LE-MvDA offers transparent projection matrices for feature analysis and delivers 0.04ms inference latency, over 40× faster than TimeSformer, enabling real-time deployment on resource-constrained devices.

LE-MvDA offers a computationally efficient and interpretable alternative to deep learning, ideal for small- to medium-scale datasets and resource-constrained deployments. To establish LE-MvDA's effectiveness against deep learning approaches, we benchmark against representative architectures including Convolutional Neural Networks (CNNs) [20], [21] and Transformers [22], as well as state-of-the-art video understanding models (SlowFast [23], TimeSformer [24]). This comparison highlights the trade-offs between end-to-end learning and modular approaches, demonstrating LE-MvDA's competitive performance while maintaining significant advantages in computational efficiency and interpretability for multi-view temporal data analysis.

It is worth noting that recent advances in efficient deep learning architectures—such as DPNet [25] and EfficientFormer [26]—demonstrate ongoing efforts to balance accuracy and computational efficiency within the deep learning paradigm. While such optimizations represent valuable improvements within deep learning frameworks, LE-MvDA offers a fundamentally different approach through shallow subspace learning, providing distinct advantages in mathematical interpretability and extreme computational efficiency that complement rather than compete with deep learning efficiency innovations.

The remainder of this paper is organized as follows: Section II reviews related work, Section III presents LE-MvDA's formulation, Section IV provides experimental validation, Section V discusses limitations and future work, and Section VI concludes the paper. Implementation is available at: https://github.com/SiamiLab/LE-MvDA.git.

II. BACKGROUND ON MULTI-VIEW LEARNING AND PRELIMINARIES

This section establishes the theoretical foundations for understanding our proposed approach. We introduce the mathematical notation used throughout this paper, then review three fundamental methods that form the basis of our work: FDA [27] for supervised dimensionality reduction, LPP [8] for neighborhood structure preservation, and MvDA [2] for multi-view discriminative learning. Understanding these methods—their formulations and limitations—provides essential context for our proposed LE-MvDA framework, which integrates discriminative power with local structure preservation in multi-view settings.

A. NOTATIONS

In this paper, scalars are represented by lowercase italic letters (e.g., x), vectors by lowercase boldface letters (e.g., $\mathbf{x}$), matrices by uppercase boldface letters (e.g., $\mathbf{X}$), and sets by uppercase calligraphic letters (e.g., $\mathcal{X}$). Furthermore, $\mathrm{tr}(\cdot)$ denotes the trace operator, $\det(\cdot)$ represents the determinant, and the transpose of a matrix or vector is indicated by the superscript $^\top$.

Additionally, to enhance the clarity of our presentation, we define common notations for MvSL-based methods. Let $\mathbf{x}_k \in \mathbb{R}^{d\times 1}$ for $k = 1, 2, \ldots, n$ represent d-dimensional samples, where $\mathbf{y}_k \in \{1, 2, \ldots, c\}$ denotes the associated class labels. Here, n represents the total number of samples, and c indicates the number of distinct classes. The number of samples in class i is denoted by n_i, satisfying the condition:

$$\sum_{i=1}^{c} n_i = n.$$

The matrix $\mathbf{X}$, composed of all the samples, is defined as:

$$\mathbf{X} = (\mathbf{x}_1 \,|\, \mathbf{x}_2 \,|\, \cdots \,|\, \mathbf{x}_n) \in \mathbb{R}^{d \times n}.$$

We aim to project the high-dimensional data $\mathbf{x}_k$ into a lower-dimensional space, resulting in representations $\mathbf{z}_k \in \mathbb{R}^{r \times 1}$, with $1 \leq r \leq d$, where r is the dimensionality of the target embedding space. Typically, d is large, while r is small. We assume a linear dimensionality reduction framework, where the transformation matrix $\mathbf{T} \in \mathbb{R}^{d \times r}$ maps the data to the lower-dimensional space:

$$\mathbf{z}_k = \mathbf{T}^\top \mathbf{x}_k.$$

To formally establish the multi-view subspace learning framework used in this study, we define the structure and notation of samples across views. Specifically, we consider the set of samples from the jth view as

$$\mathcal{X}_j = \{\mathbf{x}_{ijk} \,|\, i = 1, \ldots, c;\; k = 1, \ldots, n_{ij}\}.$$

where $\mathbf{x}_{ijk} \in \mathbb{R}^d$ represents the kth sample from the jth view of the ith class. Here, c denotes the number of classes, and n_{ij} is the number of samples from the jth view corresponding to the ith class. The samples from the v views are then projected into the common space using the v linear transformations, expressed as $\mathcal{Z} = \{\mathbf{z}_{ijk} = \mathbf{T}_j^\top \mathbf{x}_{ijk} \,|\, i = 1, \ldots, c; j = 1, \ldots, v; k = 1, \ldots, n_{ij}\}$.
To further clarify, we define the mean of the low-dimensional embeddings within each class. Specifically, let $\mu_i = \frac{1}{n_i} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} \mathbf{z}_{ijk}$ represent the mean of the low-dimensional embeddings within the i^{th} class, where n_i denotes the total number of samples in that class. Similarly, the overall mean of all low-dimensional embeddings, denoted by μ, is given by

$$\mu = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} \mathbf{z}_{ijk}.$$

To clarify, these definitions of the mean are specific to the context of low-dimensional embeddings within the MvDA and LE-MvDA methods. Any alternative definitions used elsewhere in this paper are explicitly specified. For ease of reference, key notations are summarized in Table 1.

B. FDA

One of the most widely used techniques for dimensionality reduction is FDA [27]. In this section, we briefly outline the definition of FDA. Let $\mathbf{S}^B$ and $\mathbf{S}^W$ represent the within-class scatter matrix and the between-class scatter matrix, respectively:

$$\mathbf{S}^W = \sum_{i=1}^{c} \sum_{k:y_k=i} (\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^\top,$$

$$\mathbf{S}^B = \sum_{i=1}^{c} n_i(\mu_i - \mu)(\mu_i - \mu)^\top.$$

where $\sum_{k:y_k=i}$ indicates the sum over all k such that $y_k = i$, μ_i represents the mean of the samples in class i, and μ denotes the overall mean of all samples:

$$\mu_i = \frac{1}{n_i} \sum_{k:y_k=i} \mathbf{x}_k,$$

$$\mu = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k = \frac{1}{n} \sum_{i=1}^{c} n_i \mu_i.$$

TABLE 1. Descriptions of Notations.

Notations	Descriptions
v	Number of views
c	Number of classes
n	Total samples from all views and classes
n_i	Number of samples for all views in one class
n_{ij}	Number of samples per class per view
d, r	Original and reduced data dimensions
K	Number of neighbors for affinity matrix construction
k_{clf}	Number of neighbors used in kNN classifier
$\mathbf{T} \in \mathbb{R}^{d \times r}$	Projection matrix
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Affinity matrix
$\mathbf{x}_k \in \mathbb{R}^{d \times 1}$	Data point in original space
$\mathbf{X} \in \mathbb{R}^{d \times n}$	Data matrix in original space
$\mathbf{z}_k \in \mathbb{R}^{r \times 1}$	Data point in target space
$\mathbf{Z} \in \mathbb{R}^{r \times n}$	Data matrix in target space
$\mathbf{y}_k \in \mathbb{R}^{c \times 1}$	Class labels

We assume that $\mathbf{S}^W$ has full rank. The FDA transformation matrix $\mathbf{T}_{FDA}$ is defined as follows:

$$\mathbf{T}_{FDA} = \underset{\mathbf{T} \in \mathbb{R}^{d \times r}}{\operatorname{argmax}} \left[\frac{\det\left(\mathbf{T}^\top \mathbf{S}^B \mathbf{T}\right)}{\det\left(\mathbf{T}^\top \mathbf{S}^W \mathbf{T}\right)} \right].$$

FDA aims to find a transformation matrix $\mathbf{T}$ that maximizes the between-class scatter while minimizing the within-class scatter. In the formulation above, it is implicitly assumed that $\mathbf{T}^\top \mathbf{S}^W \mathbf{T}$ is invertible. Consequently, the optimization is subject to the constraint:

$$\operatorname{rank}(\mathbf{T}) = r.$$

Let $\{\varphi_m\}_{m=1}^{d}$ represent the generalized eigenvectors corresponding to the generalized eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$, as determined by the following generalized eigenvalue problem:

$$\mathbf{S}^B \varphi = \lambda \mathbf{S}^W \varphi.$$

The analytical solution to the above maximization problem for T_{FDA} is given by

$$\mathbf{T}_{FDA} = (\varphi_1 | \varphi_2 | \cdots | \varphi_r).$$

C. LPP

PCA and LDA are designed to preserve the global structure of data. However, in many practical scenarios, maintaining the local structure is more critical. LPP [8] is an algorithm that addresses this by learning a subspace that preserves local data relationships. It focuses on preserve the intrinsic geometry of the data, ensuring that samples close to each other in the original space remain close in the projected space. The LPP transformation matrix $\mathbf{T}_{LPP}$ is defined as follows:

$$\mathbf{T}_{LPP} = \underset{\mathbf{T}\in\mathbb{R}^{d\times r}}{\text{argmin}} \left(\frac{1}{2} \sum_{k,l=1}^{n} \mathbf{A}_{k,l} \|\mathbf{T}^\top \mathbf{x}_k - \mathbf{T}^\top \mathbf{x}_l\|^2 \right), \quad \text{subject to} \quad \mathbf{T}^\top \mathbf{X}\mathbf{D}\mathbf{X}^\top \mathbf{T} = \mathbf{I}_r. \tag{1}$$

Here, $\mathbf{D}$ is an n-dimensional diagonal matrix, with the k-th diagonal element defined as $\mathbf{D}_{k,l}$. Let $\mathbf{A}$ be an affinity matrix, an $n \times n$ matrix where the (k,l)-th element $\mathbf{A}_{k,l}$ quantifies the affinity between samples $\mathbf{x}_k$ and $\mathbf{x}_l$. We assume that $\mathbf{A}_{k,l} \in [0,1]$, where $\mathbf{A}_{k,l}$ is large when $\mathbf{x}_k$ and $\mathbf{x}_l$ are "close" and small when they are "far apart." There are various methods to define the affinity matrix $\mathbf{A}$, including approaches based on the Heat Kernel, Euclidean Distance, Nearest Neighbor, and Local Scaling.

$$\mathbf{D}_{k,k} \equiv \sum_{l=1}^{n} A_{k,l}.$$

Equation 1 indicates that LPP aims to find a transformation matrix $\mathbf{T}$ that ensures data pairs close together in the original space $\mathbb{R}^d$ remain close in the embedding space. The constraint in Equation 1 is introduced to avoid degenerate solutions. Let $\{\psi_k\}_{m=1}^{d}$ represent the generalized eigenvectors associated with the generalized eigenvalues $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_d$, satisfying the following generalized eigenvalue problem:

$$\mathbf{X}\mathbf{L}\mathbf{X}^\top \psi = \gamma \mathbf{X}\mathbf{D}\mathbf{X}^\top \psi,$$

where

$$\mathbf{L} = \mathbf{D} - \mathbf{A}.$$

In spectral graph theory [28], $\mathbf{L}$ is referred to as the graph-Laplacian matrix, with $\mathbf{A}$ being the adjacency matrix of the graph. The solution can be expressed as

$$\mathbf{T}_{LPP} = (\psi_d \,|\, \psi_{d-1} \,|\, \cdots \,|\, \psi_{d-r+1}).$$

D. MVDA

MvDA [2], an extension of LDA designed for multi-view problems, was developed to jointly capture view correlation, intra-view discriminability, and inter-view discriminability. MvDA aims to determine a set of linear transformations, $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_v$, that project samples from v distinct views into a shared discriminative space. The optimal projection matrices $\mathbf{T}_v$ are obtained by maximizing the ratio of between-class scatter to within-class scatter, ensuring that projected data in the lower-dimensional space achieves maximum class separability across all views.

In this shared space, MvDA's objective is to maximize the between-class scatter, $\mathbf{S}_{\mathcal{Z}}^{B}$, while minimizing the within-class scatter, $\mathbf{S}_{\mathcal{Z}}^{W}$. This objective is effectively formulated as a generalized Rayleigh quotient, framing the problem as an optimization task. Letting $\mathbf{T}_v$ represent the projection matrix for view v, the MvDA objective can be expressed as:

$$\mathbf{T}_{MvDA} = \underset{\mathbf{T}_1,\ldots,\mathbf{T}_v}{\arg\max} \frac{\text{tr}(\mathbf{S}_{\mathcal{Z}}^{B})}{\text{tr}(\mathbf{S}_{\mathcal{Z}}^{W})}. \tag{2}$$

where the within-class scatter matrix $\mathbf{S}_{\mathcal{Z}}^{W}$ accounts for variability within each class across views, and the between-class scatter matrix $\mathbf{S}_{\mathcal{Z}}^{B}$ captures the separability between different classes. Specifically, these matrices are defined as:

$$\mathbf{S}_{\mathcal{Z}}^{W} = \sum_{i=1}^{c} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} \left(\mathbf{z}_{ijk} - \mu_i\right) \left(\mathbf{z}_{ijk} - \mu_i\right)^T, \tag{3}$$

$$\mathbf{S}_{\mathcal{Z}}^{B} = \sum_{i=1}^{c} n_i \left(\mu_i - \mu\right) \left(\mu_i - \mu\right)^T. \tag{4}$$

Here, $\mathbf{z}_{ijk}$ denotes the projected samples, μ_i the class mean for class i, and μ the global mean across all classes, enabling MvDA to achieve an optimal discriminative structure within the reduced-dimensional space.

III. LOCALLY ENHANCED MULTI-VIEW DISCRIMINANT ANALYSIS

In this section, we present LE-MvDA, a supervised multi-view classification approach that enhances traditional discriminant analysis for high-dimensional data. Building on the MvDA method, LE-MvDA incorporates local data structures to improve discriminative power. The optimization objective to determine the projection matrices is formulated as

$$\mathbf{T}_{LE-MvDA} = \underset{\mathbf{T}_1,\ldots,\mathbf{T}_v}{\arg\max} \frac{\text{tr}(\mathbf{T}^T \tilde{\mathbf{S}}_{\mathcal{Z}}^{B} \mathbf{T})}{\text{tr}(\mathbf{T}^T \tilde{\mathbf{S}}_{\mathcal{Z}}^{W} \mathbf{T})}. \tag{5}$$

where $\tilde{\mathbf{S}}_{\mathcal{Z}}^{B}$ and $\tilde{\mathbf{S}}_{\mathcal{Z}}^{W}$ denote the local between-class and within-class scatter matrices, respectively. Formally, the local within-class scatter matrix for the low-dimensional embeddings in the common space can be formulated as shown below. For a detailed derivation, please refer to Appendix :

$$\tilde{\mathbf{S}}_{\mathcal{Z}}^{W} = \left[\mathbf{T_1}^\top \; \mathbf{T_2}^\top \; \ldots \; \mathbf{T_v}^\top\right] \begin{pmatrix} S_{11} & S_{12} & \ldots & S_{1v} \\ S_{21} & S_{22} & \ldots & S_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ S_{v1} & S_{v2} & \ldots & S_{vv} \end{pmatrix} \begin{bmatrix} \mathbf{T_1} \\ \mathbf{T_2} \\ \vdots \\ \mathbf{T_v} \end{bmatrix},$$

$$= \sum_{j,r} \mathbf{T_j}^\top \mathbf{S}_{jr} \mathbf{T_r}.$$

with $\mathbf{T} = \left[\mathbf{T_1}^\top, \mathbf{T_2}^\top, \ldots, \mathbf{T_v}^\top\right]^\top$ and $\mathbf{S}_{jr}$ defined as follows:

$$\mathbf{S}_{jr} = \begin{cases} \sum_{i=1}^{c} \sum_{k:y_k=i} \left(\sum_{l:y_l=i} \mathbf{A}_{k,l}^{T}\right) \mathbf{x}_{ijk}\mathbf{x}_{ijk}^\top \\ \quad - \sum_{i=1}^{c} \sum_{k,l:y_k,y_l=i} \mathbf{A}_{k,l}^{T} \mathbf{x}_{ijl}\mathbf{x}_{irk}^\top, & j = r \\ \\ - \sum_{i=1}^{c} \sum_{k,l:y_k,y_l=i} \mathbf{A}_{k,l}^{T} \mathbf{x}_{ijl}\mathbf{x}_{irk}^\top. & j \neq r \end{cases} \tag{6}$$

Algorithm 1 The algorithm of LE-MvDA for dimensionality reduction and Classification

```
Require: $X \in \mathbb{R}^{d \times n}$: Data matrix of training samples,
    $\mathbf{y}_k \in \{1, 2, \ldots, c\}$: class labels,
    $A \in \mathbb{R}^{n \times n}$: Affinity matrix,
    $\mathbf{n}_{ij}$: Number of samples for class $i$ and view $j$,
    $c$: Number of classes,
    $v$: Number of views,
    $k_{\text{PCA}}$: Number of principal components,
    $K$: Number of neighbors for affinity matrix construction,
    $k_{\text{clf}}$: Number of neighbors used in final kNN classifier,
    $d, r$ : Original and reduced data dimensions
    $\lambda$: Regularization parameter.
Ensure: $T \in \mathbb{R}^{d \times r}$: Dimensionality reduction matrix
    Compute sample totals:
        $$n_i = \sum_{j=1}^{v} n_{ij}, \quad n = \sum_{i=1}^{c} n_i$$
    Center the data, compute covariance matrix, extract top $k_{\text{PCA}}$ eigenvectors, and project to PCA subspace
    Compute the mean matrix $\mu_{ij}$ for each class $i$ and view $j$
    Compute local scaling factors $\sigma_k, \sigma_l$ for affinity matrix construction based on $K$-nearest neighbors, as defined in Equation (8).
    Compute affinity values via a Gaussian kernel with local scaling using Equation (7)
    Compute the within-class scatter matrix $\mathbf{S}_{jr}$ using Equation (6)
    Compute the between-class scatter matrix $\mathbf{D}_{jr}$ by Equation (9)
    Reshape $\mathbf{S}_{jr}$ and $\mathbf{D}_{jr}$ from 4D tensors to 2D matrices $\mathbf{S}_{jr}^{2D}$ and $\mathbf{D}_{jr}^{2D}$
    Regularize the within-class scatter matrix:
        $$\mathbf{S}_{jr}^{2D} \leftarrow \mathbf{S}_{jr}^{2D} + \lambda I$$
    Solve the generalized eigenvalue problem to obtain eigenvectors $\mathbf{T}$ and eigenvalues $\mathbf{D}$:
        $$\mathbf{D}_{jr}^{2D}\mathbf{w} = \lambda \mathbf{S}_{jr}^{2D}\mathbf{w}$$
    Sort eigenvalues in descending order and retain the top $r$ eigenvectors to form the projection matrix $T$.
    Project the data onto the learned LE-MvDA subspace:
        $$Z_{\text{train}} = \mathbf{T}^\top X_{\text{train}}, \quad Z_{\text{test}} = \mathbf{T}^\top X_{\text{test}}$$
    Perform cross-validation to select the optimal number of neighbors $k_{\text{cls}}$ for k-Nearest Neighbors (kNN) classification
```

The weight matrix $\mathbf{A}_{k,l}^{T}$ encodes the affinity between samples and is given by:

$$\mathbf{A}_{k,l}^{T} = \begin{cases} \dfrac{\exp\left(-\frac{\|\mathbf{x}_k - \mathbf{x}_l\|^2}{\sigma_k \sigma_l}\right)}{n_i} & \text{if } y_k = y_l = i, \\ 0 & \text{if } y_k \neq y_l. \end{cases} \tag{7}$$

where σ_k and σ_l are local scaling parameters, following the self-tuning method proposed in [19]. Specifically, each local scale is computed based on the distance to the K-th nearest neighbor:

$$\sigma_k = \|\mathbf{x}_k - \mathbf{x}_k^{(K)}\|, \quad \sigma_l = \|\mathbf{x}_l - \mathbf{x}_l^{(K)}\|. \tag{8}$$

where $\mathbf{x}_k^{(K)}$ and $\mathbf{x}_l^{(K)}$ denote the K-th nearest neighbors of $\mathbf{x}_k$ and $\mathbf{x}_l$, respectively.

This adaptive local scaling ensures that the affinity weights better reflect the underlying geometry of the data. This approach automatically adapts to local density variations, with smaller σ values in dense regions and larger σ values in sparse regions, making the affinity construction robust across different data distributions.

$$\tilde{\mathbf{S}}_{\mathcal{Z}}^{B} = \begin{bmatrix} \mathbf{T_1}^\top \; \mathbf{T_2}^\top \; \ldots \; \mathbf{T_v}^\top \end{bmatrix} \begin{pmatrix} D_{11} & D_{12} & \ldots & D_{1v} \\ D_{21} & D_{22} & \ldots & D_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ D_{v1} & D_{v2} & \ldots & D_{vv} \end{pmatrix} \begin{bmatrix} \mathbf{T_1} \\ \mathbf{T_2} \\ \vdots \\ \mathbf{T_v} \end{bmatrix},$$

$$= \sum_{j,r} \mathbf{T_j}^\top \mathbf{D}_{jr} \mathbf{T_r}.$$

where D_{jr} is defined below. Here, $\mu_{ij}^{(x)} = \frac{1}{n_{ij}} \sum_{k=1}^{n_{ij}} x_{ijk}$ denotes the mean of the samples in the original feature space corresponding to the i-th class and the j-th view.

$$D_{jr} = \left(\sum_{i=1}^{c} \frac{n_{ij} n_{ir}}{n_i} \mu_{ij}^{(x)} \mu_{ir}^{(x)\top} \right) - \frac{1}{n} \left(\sum_{i=1}^{c} n_{ij} \mu_{ij}^{(x)} \right) \left(\sum_{i=1}^{c} n_{ir} \mu_{ir}^{(x)} \right)^\top. \tag{9}$$

Algorithm 1 summarizes the complete LE-MvDA dimensionality reduction and classification process.

As noted in [29], the objective function in (5) is structured as a trace ratio, which prevents the existence of a closed-form solution. To facilitate a more tractable approach, we reformulate it as a ratio trace, as shown below:

$$\mathbf{T}_{LE-MvDA} = \arg \max_{\mathbf{T}_1, \ldots, \mathbf{T}_v} \operatorname{tr} \left(\frac{\mathbf{T}^T \tilde{\mathbf{S}}_{\mathcal{Z}}^{B} \mathbf{T}}{\mathbf{T}^T \tilde{\mathbf{S}}_{\mathcal{Z}}^{W} \mathbf{T}} \right). \tag{10}$$

which can be addressed analytically via generalized eigenvalue decomposition.

1) Computational Complexity

The overall training cost of LE-MvDA can be summarized as follows:

- **Affinity matrix construction:** $O(n^2 d)$ for pairwise distance computation and $O(n^2 \log k)$ for k-nearest neighbor search.
- **Within-class scatter matrices:** $O(n^2 d^2)$ due to pairwise operations and matrix multiplications.
- **Generalized eigenvalue problem:** $O(d^3)$ for eigendecomposition, assuming a small number of views v.
- **Between-class scatter:** $O(cv^2 d^2)$, which is negligible compared to the above terms, where c is the number of classes.

Total training complexity: $O(n^2d^2 + d^3)$, which is asymptotically comparable to MvDA's $O(n^2d^2)$, with an additional $O(d^3)$ term for eigen-decomposition. In practice, the $O(n^2d^2)$ term dominates for large n.

A. LIMITATIONS OF EXISTING APPROACHES AND LE-MVDA'S THEORETICAL DISTINCTIONS

While several attempts have been made to incorporate local structures into multi-view learning, existing approaches face fundamental limitations. Locality-preserving variants of Canonical Correlation Analysis focus on correlation maximization rather than discriminant analysis, limiting their applicability to supervised classification tasks. Local multi-view clustering methods typically address unsupervised learning scenarios and fail to leverage class label information for enhanced discrimination. Among supervised multi-view methods, MvDA represents a significant advancement by combining multi-view learning with discriminant analysis principles. However, MvDA faces limitations in certain scenarios where it may not identify the most class-separable space. This challenge arises from its focus on maximizing class separation from the global mean, which can overlook the distinctions between closely related class pairs [1]. Although MvDA has proven effective in still image recognition tasks, its application to video-based recognition remains unexplored. Additionally, MvDA does not explicitly enforce separation of class centers, potentially reducing its discriminative capability [1]. To overcome the identified challenges, LE-MvDA introduces a supervised locality preservation mechanism specifically designed for multi-view discriminant analysis through its class-aware affinity construction and unified optimization formulation. The weight matrix $\mathbf{A}^T_{k,l}$, defined in Equation (7), plays a central role in LE-MvDA by encoding the class-aware local affinity between samples. This matrix is constructed to preserve local neighborhood information only within the same class, thereby enhancing intra-class compactness without affecting inter-class separability. Unlike unsupervised locality-preserving methods (e.g., LPP or standard graph Laplacians), our affinity matrix is explicitly supervised: entries are nonzero only when both samples $\mathbf{x}_k$ and $\mathbf{x}_l$ belong to the same class. Furthermore, to avoid fixed-scale kernel sensitivity and to better adapt to local density variations, the similarity between samples is scaled using local scale parameters σ_k and σ_l computed via the self-tuning method in [19]. Each σ reflects the distance to the K-th nearest neighbor of a sample, enabling the affinity structure to automatically adapt to varying sample distributions. This local scaling significantly improves robustness in heterogeneous or high-dimensional feature spaces. By integrating a class-aware, adaptively scaled affinity matrix directly into the multi-view discriminant objective, LE-MvDA unifies local geometric structure and discriminative learning within a single optimization framework—a key advancement over prior methods that address these aspects independently. This integration gives rise to what we term "discriminative local neighborhoods," where the simultaneous optimization of both objectives leads to synergistic, rather than additive, gains in classification performance.

IV. EXPERIMENTAL EVALUATION AND BENCHMARKING

In this section, we compare the classification accuracy of LE-MvDA against several established methods, including MCCA, MvDA, pc-MvDA, and MvDA-vc. The proposed LE-MvDA is evaluated on three real-world datasets using a system equipped with 32 GB of RAM, an Intel Core i9-12900H CPU (2.50 GHz, 14 cores), and an NVIDIA GeForce RTX 3080 GPU.

The remainder of this section introduces the datasets used in our study, outlines the experimental setup and evaluation framework, and presents detailed results comparing LE-MvDA with the baseline methods.

A. DATASETS

1) Multi-PIE

The CMU Multi-PIE dataset [30] provides over $750,000$ face images of 337 individuals under varying viewpoints, lighting conditions, and expressions. It includes 13 head-level camera views spaced at 15° intervals from -90° to 90°, along with two overhead views simulating surveillance angles. For our study, we selected a subset of $62,400$ images from 240 subjects, each captured in 13 poses with 20 samples per pose, focusing on neutral expressions for consistency. All images were cropped to 56×46 pixels. Figure 1 illustrates examples of two individuals from different viewpoints.

2) ORL

The ORL dataset [31] comprises 400 images of 40 subjects, with 10 images per subject captured from different angles over two years under varying lighting conditions, facial expressions, and facial features. Each image has a resolution of 92×112 pixels with 256 grayscale levels. Following the feature extraction strategy in [32], we created three views for each image: the original grayscale image, features extracted using Fast Fourier Transform (FFT) [33], and features extracted using Canny edge detection [34]. This multi-view configuration enables a fair comparison of LE-MvDA with other multi-view learning methods. Figure 2 shows examples of two subjects from the original view, while Figure 3 illustrates all three feature representations.

3) RoboMNIST

RoboMNIST [35] is a multimodal dataset designed for multi-robot activity recognition (MRAR), integrating synchronized data from three cameras (video), three WiFi sniffers (CSI), and three microphones (audio). It features two Franka Emika robotic arms performing 10 distinct activities—drawing digits 0 through 9—at three velocity levels (high, medium, low), resulting in 60 unique combinations with 32 repetitions each. Recordings are 15 seconds long at 30 Hz, providing RGB video from three camera angles at 2560×720 resolution. Figure 4 illustrates the end effector's positions for each activity, providing a visual representation of the different trajectories

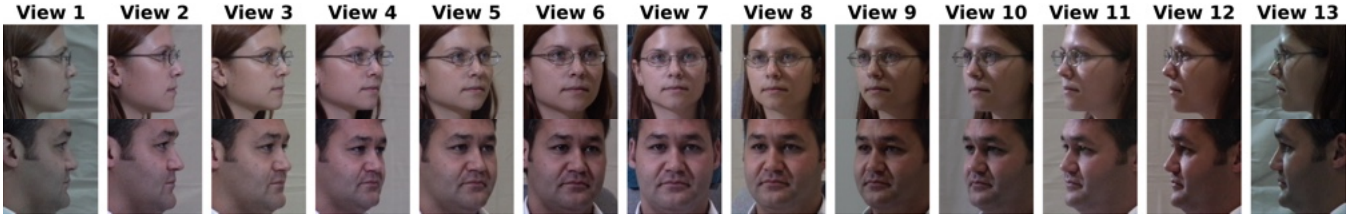


FIGURE 1. Example from the Multi-PIE dataset: Two individuals shown with neutral facial expressions from different viewpoints.

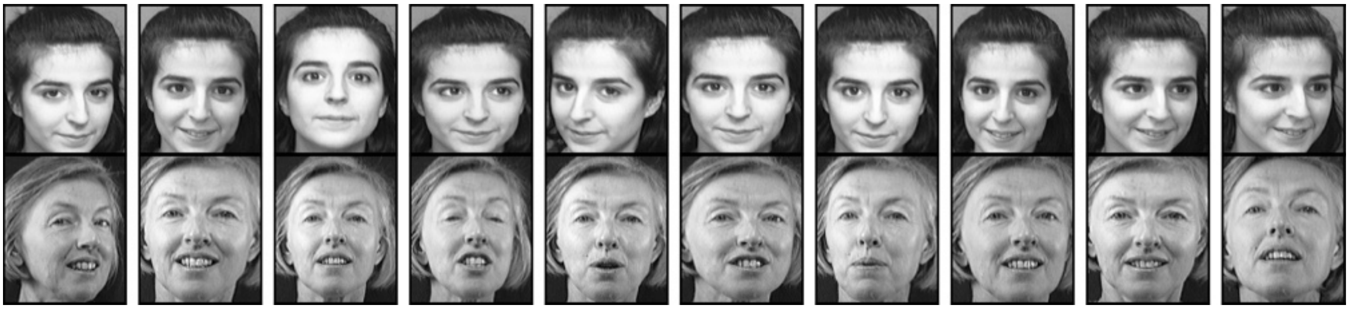


FIGURE 2. Example from the ORL Dataset: Two individuals shown with neutral facial expressions.

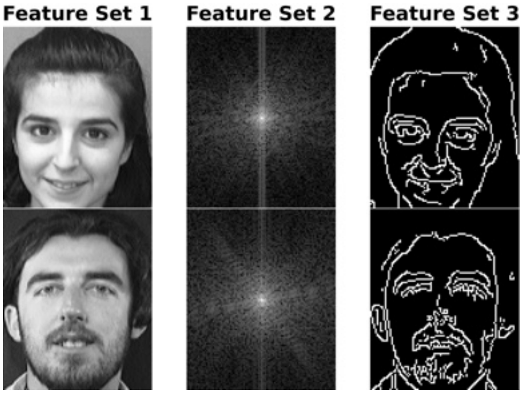


FIGURE 3. Representative samples from the ORL dataset. The first column shows the original grayscale images (Feature Set 1), the second column presents frequency-domain features extracted using FFT (Feature Set 2), and the third column depicts edge-based features obtained with the Canny edge detector (Feature Set 3).

across the activities.

For this study, we selected a subset comprising one robotic arm, three cameras positioned to capture different viewpoints, operation at low velocity, and 30 repetitions per class. This configuration results in a data tensor with dimensions $450 \times 10 \times 3 \times 30$, where 450 corresponds to the number of frames per sample (30 frames per second over 15 seconds), 10 represents the number of activity classes, 3 denotes the number of camera views, and 30 indicates the number of repetitions per class.

While this study focuses on single-robot scenarios, the dataset's multi-robot architecture demonstrates LE-MvDA's potential scalability to complex coordination tasks. In multi-robot settings, LE-MvDA's tensor-based formulation can naturally accommodate multiple robots as additional 'views,' where each robot's sensor data (end-effector positions) contributes to coordinated action recognition. The class-aware affinity matrix construction can capture synchronized behaviors across robot teams, enabling discrimination between different coordinated actions.

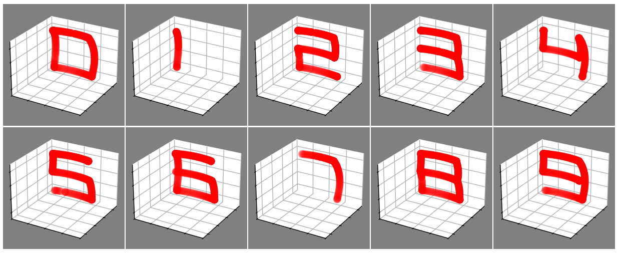


FIGURE 4. Example from the RoboMNIST dataset: The robotic arm draws numbers 0 through 9 on a vertical imaginary plane, representing 10 distinct activity classes. The plot shows the end effector trajectories for these numbers, with initial and final positioning phases omitted for clarity, and the robotic arm and background removed.

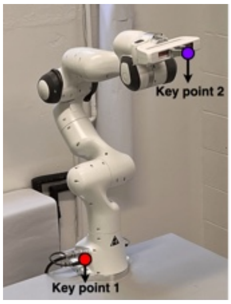


FIGURE 5. Pose detection example from the RoboMNIST dataset showing two key points annotated on the robotic arm: Key Point 1 (base of the arm, highlighted in red) and Key Point 2 (end effector, highlighted in purple).

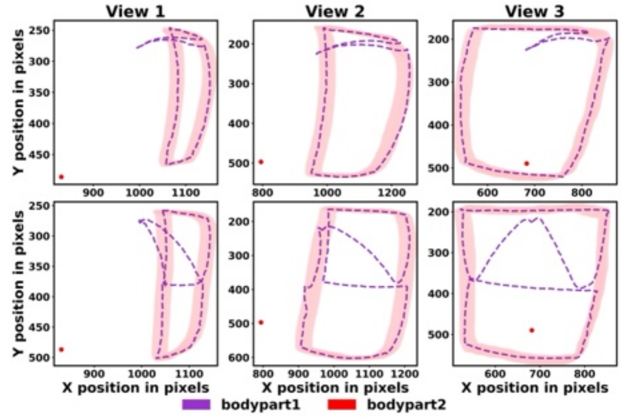


FIGURE 6. Visualization of the robotic arm's end-effector trajectories, extracted via pose detection, for two classes (class 0: top row; class 8: bottom row) across three camera viewpoints (columns). Differences in curvature and thickness reflect perspective variations, highlighting the challenges of multi-view representation. The x- and y-axes denote pixel positions, and the class numbers are highlighted for clarity.

B. EXPERIMENTAL SETUP AND VALIDATION FRAMEWORK

This section details the experiments conducted on three real-world datasets to assess the effectiveness of the proposed LE-MvDA in multi-view classification tasks. We benchmark LE-MvDA against four established baseline methods: MCCA, MvDA, pc-MvDA, and MvDA-vc. These baselines were chosen as they represent both foundational and enhanced variants of multi-view discriminant analysis. MCCA serves as an unsupervised correlation-based baseline, while MvDA provides a supervised framework that maximizes inter-class separability across views. pc-MvDA and MvDA-vc build upon MvDA by incorporating pairwise class-center constraints and view-consistency regularization, respectively.

To mitigate the effects of high dimensionality and address the Small Sample Size (SSS) problem [36], we applied PCA as a preprocessing step for all methods. This ensures a fair and consistent dimensionality reduction strategy across experiments.

Building on this setup, we implement a two-phase framework for multi-view classification. The first phase involves learning a shared subspace from multi-view training data, while the second phase uses the learned representation to classify unseen test samples.

1) Learning Phase

Given v views, the learning phase involves three key steps:

Feature Extraction:

- **Multi-PIE and ORL Datasets:** For the Multi-PIE dataset, no additional feature extraction techniques were required, as the raw pixel values provided sufficient information for analysis. Similarly, in the case of the ORL dataset, the raw pixel values were used for the original view, while FFT and Canny edge detection were applied to generate features for the other two views.
- **RoboMNIST Dataset:** In the RoboMNIST dataset, training videos were processed using DeepLabCut [37] for pose detection. We extracted 20 frames per video and annotated two key points: the base of the robotic arm and the end effector, as shown in Figure 5. DeepLabCut employs a ResNet-50 [38] model pre-trained on ImageNet [39] and fine-tuned for our task. The resulting trajectories illustrate the end effector's movement over a 15-second period, captured at 450 timestamps from three distinct viewpoints. Figure 6 shows the pose detection results for two distinct classes (0 and 8), which represent an example of closely positioned class boundaries where the curved movement patterns share similar shapes but differ in subtle execution details, with data from three camera views (columns labeled as View 1, View 2, and View 3). Each subplot visualizes the trajectory of the end effector, which is extracted using deep learning-based pose detection and serves as a feature vector capturing the movement dynamics across frames for further analysis. The x-axis represents the horizontal position in pixels, while the y-axis indicates the vertical position in pixels. The color coding shown in the legend below the plots differentiates between bodypart1 (purple) and bodypart2 (red) in the detection process, providing insights into how different body parts contribute to movement dynamics across the three camera views. The numbers 0 and 8 are highlighted in red for visual clarity. The full pipeline, where the trajectory of the end-effector's poses is constructed and used as a feature vector, is illustrated in Figure 7.

Common Space Construction: In this step, a common feature space is built where training samples from the same class are projected close to one another, even if captured from different viewpoints. This is achieved by deriving v linear transformation matrices, T, from the reduced training set using each of the aforementioned methods. These matrices are then applied to project the training samples from each view into a low-dimensional subspace.

Building the Classifier: Once the v transformations are learned, the features projected into the common space from each view are used to train a single predictive model, ϕ, such as a k-Nearest Neighbors (k-NN) classifier. The classifier is constructed based on the projected training data, where each sample is labeled according to its corresponding class. To optimize performance, the number of neighbors (K) is tuned by varying K between 1 and 10, using majority voting among the K nearest neighbors.

2) Prediction Phase

The prediction phase consists of two stages:

Feature Extraction: For the testing sample $\mathbf{x}_{ijk}$ from the j-th view, features are extracted and subsequently projected into a reduced-dimensional subspace using the corresponding transformation matrix $\mathbf{T}_j$. The projected feature vector is expressed as $\mathbf{z}_{ijk} = \mathbf{T}_j^\top \mathbf{x}_{ijk}$.

Class Prediction: The classifier $\phi(\mathbf{z}_{ijk})$ processes the projected feature $\mathbf{z}_{ijk}$, and assigns the appropriate action label based on the classifier's output.

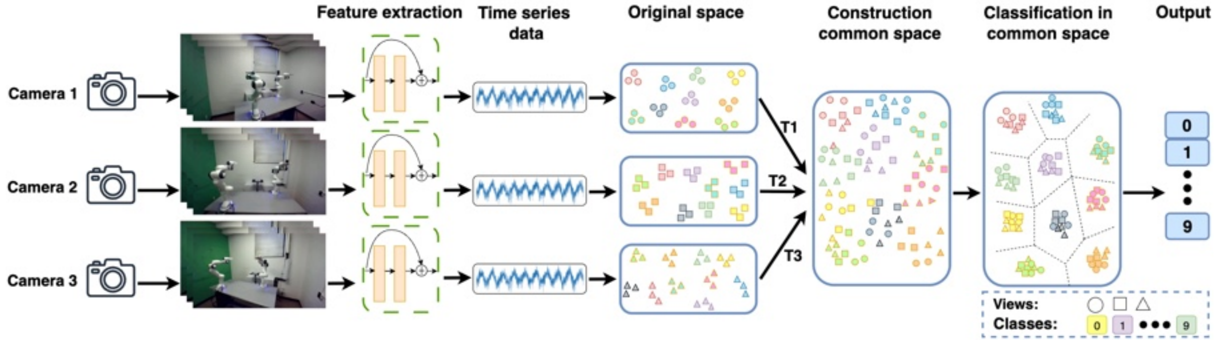
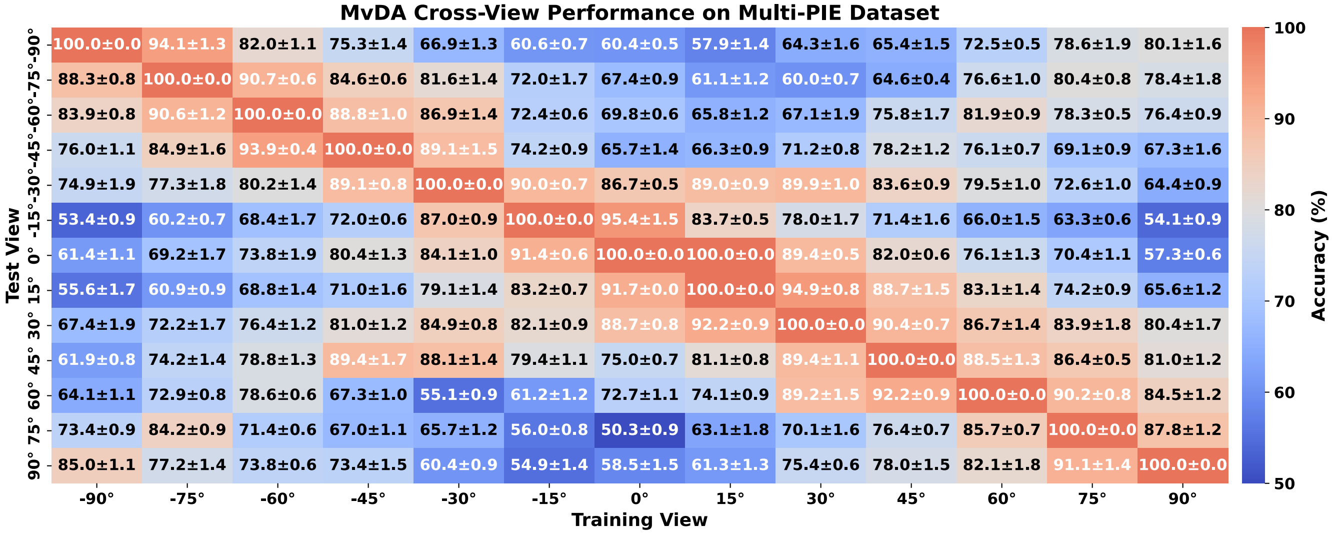


FIGURE 7. Multi-view Robot Activity Classification Pipeline. The framework consists of three main stages: (1) *Pose estimation* is performed using DeepLabCut with a ResNet-50 backbone, extracting keypoints from videos captured by three synchronized cameras observing robotic actions. The extracted features are converted into view-specific *time-series data*, each encoding the motion dynamics from a different viewpoint. (2) These time-series sequences are mapped from their original feature spaces into a *shared subspace* using linear transformation matrices T_1, T_2, and T_3, one per view. (3) Classification is then performed in the common subspace, where samples from different views are projected close together based on class similarity. Shapes indicate different views (circle, square, triangle), and colors represent distinct activity classes (0–9), enabling consistent recognition across views.



Test View \ Training View	-90°	-75°	-60°	-45°	-30°	-15°	0°	15°	30°	45°	60°	75°	90°
-90°	100.0±0.0	94.1±1.3	82.0±1.1	75.3±1.4	66.9±1.3	60.6±0.7	60.4±0.5	57.9±1.4	64.3±1.6	65.4±1.5	72.5±0.5	78.6±1.9	80.1±1.6
-75°	88.3±0.8	100.0±0.0	90.7±0.6	84.6±0.6	81.6±1.4	72.0±1.7	67.4±0.9	61.1±1.2	60.0±0.7	64.6±0.4	76.6±1.0	80.4±0.8	78.4±1.8
-60°	83.9±0.8	90.6±1.2	100.0±0.0	88.8±1.0	86.9±1.4	72.4±0.6	69.8±0.6	65.8±1.2	67.1±1.9	75.8±1.7	81.9±0.9	78.3±0.5	76.4±0.9
-45°	76.0±1.1	84.9±1.6	93.9±0.4	100.0±0.0	89.1±1.5	74.2±0.9	65.7±1.4	66.3±0.9	71.2±0.8	78.2±1.2	76.1±0.7	69.1±0.9	67.3±1.6
-30°	74.9±1.9	77.3±1.8	80.2±1.4	89.1±0.8	100.0±0.0	90.0±0.7	86.7±0.5	89.0±0.9	89.9±1.0	83.6±0.9	79.5±1.0	72.6±1.0	64.4±0.9
-15°	53.4±0.9	60.2±0.7	68.4±1.7	72.0±0.6	87.0±0.9	100.0±0.0	95.4±1.5	83.7±0.5	78.0±1.7	71.4±1.6	66.0±1.5	63.3±0.6	54.1±0.9
0°	61.4±1.1	69.2±1.7	73.8±1.9	80.4±1.3	84.1±1.0	91.4±0.6	100.0±0.0	100.0±0.0	89.4±0.5	82.0±0.6	76.1±1.3	70.4±1.1	57.3±0.6
15°	55.6±1.7	60.9±0.9	68.8±1.4	71.0±1.6	79.1±1.4	83.2±0.7	91.7±0.0	100.0±0.0	94.9±0.8	88.7±1.5	83.1±1.4	74.2±0.9	65.6±1.2
30°	67.4±1.9	72.2±1.7	76.4±1.2	81.0±1.2	84.9±0.8	82.1±0.9	88.7±0.8	92.2±0.9	100.0±0.0	90.4±0.7	86.7±1.4	83.9±1.8	80.4±1.7
45°	61.9±0.8	74.2±1.4	78.8±1.3	89.4±1.7	88.1±1.4	79.4±1.1	75.0±0.7	81.1±0.8	89.4±1.1	100.0±0.0	88.5±1.3	86.4±0.5	81.0±1.2
60°	64.1±1.1	72.9±0.8	78.6±0.6	67.3±1.0	55.1±0.9	61.2±1.2	72.7±1.1	74.1±0.9	89.2±1.5	92.2±0.9	100.0±0.0	90.2±0.8	84.5±1.2
75°	73.4±0.9	84.2±0.9	71.4±0.6	67.0±1.1	65.7±1.2	56.0±0.8	50.3±0.9	63.1±1.8	70.1±1.6	76.4±0.7	85.7±0.7	100.0±0.0	87.8±1.2
90°	85.0±1.1	77.2±1.4	73.8±0.6	73.4±1.5	60.4±0.9	54.9±1.4	58.5±1.5	61.3±1.3	75.4±0.6	78.0±1.5	82.1±1.8	91.1±1.4	100.0±0.0

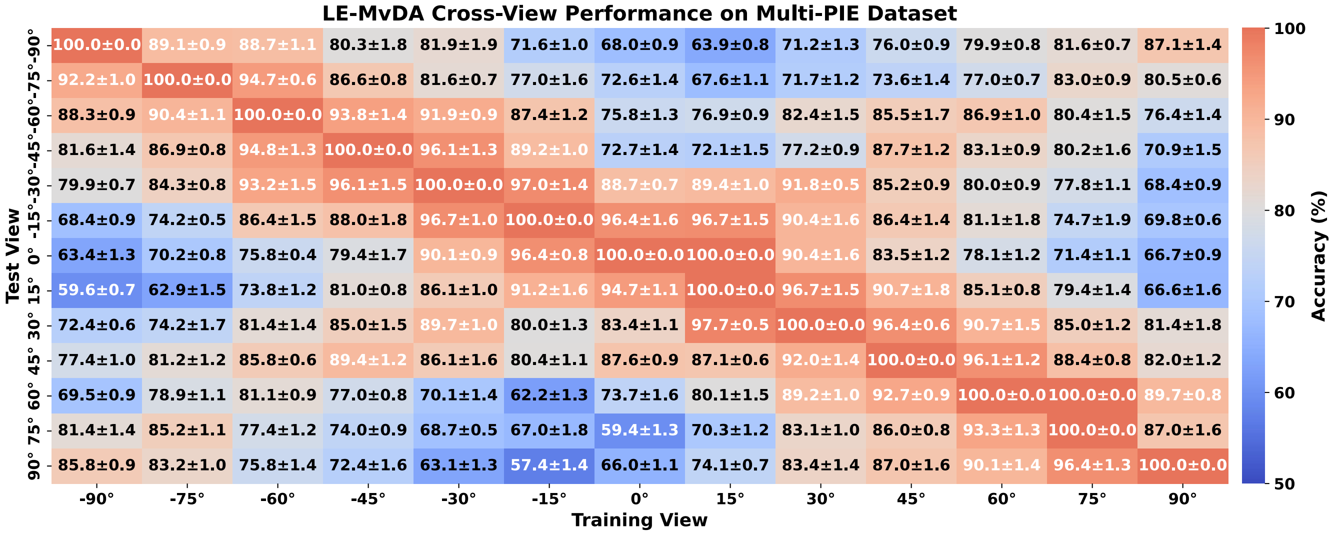
FIGURE 8. Heatmap visualization of cross-view recognition accuracy of MvDA on the Multi-PIE dataset, reported as mean ± standard deviation over five folds. Each cell indicates the rank-1 classification accuracy (%) across all pairwise combinations of camera views.

C. EXPERIMENTAL RESULTS

1) Experimental results on Multi-PIE dataset

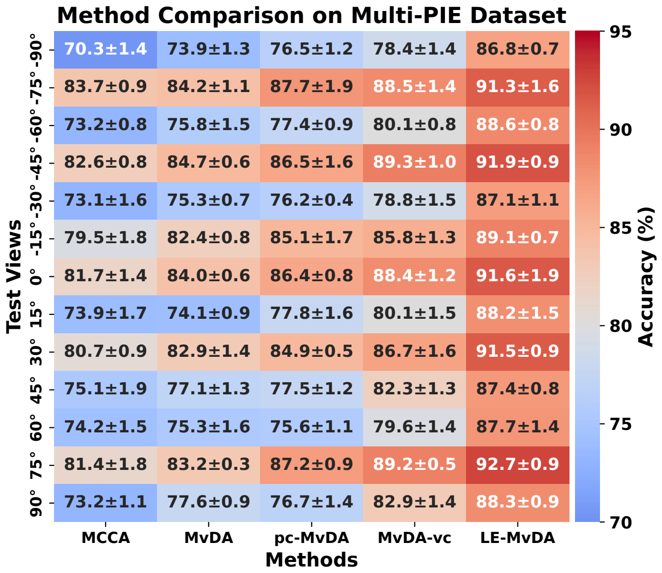
Face recognition across different poses is evaluated using both pairwise and multi-view approaches on the Multi-PIE dataset, inspired by the general evaluation methodology presented in [2]. In the pairwise approach, the model is trained on one view and tested on a different view. Since the Multi-PIE dataset includes 13 distinct views, this results in $13 \times 12 = 156$ pairwise evaluations, with rank-1 recognition accuracy for each pair visualized in the heatmaps shown in Figures 8 and 9. As illustrated in these heatmaps, each cell reports the percentage of correct rank-1 recognition for its respective test and train view combination. Diagonal entries represent cases where the model is trained and tested on the same view, showing the highest performance (red colors indicating high accuracy). Off-diagonal values reflect cross-view performance, where the model is tested on a view different from the one it was trained on. From Figure 8, we observe that MvDA exhibits a clear dependency on viewpoint, with better performance when the training and testing views are closer in angle, as evidenced by the warmer (red) regions near the diagonal. Performance decreases significantly as the angular difference increases, shown by the cooler (blue) regions further from the diagonal. In contrast, Figure 9 demonstrates that LE-MvDA shows stronger cross-view generalization than MvDA, with consistently warmer colors across the off-diagonal entries. For instance, training on -90° and testing on -75° yields 92.2% accuracy for LE-MvDA, compared to 88.3% for MvDA. The heatmap visualization clearly illustrates LE-MvDA's superior ability to handle cross-view variations, with higher recognition rates across various angular shifts, making it a more robust solution for multi-view face recognition.

Additionally, we evaluated LE-MvDA in a multi-view scenario, where for each experiment, samples from all views except one are used for training, and the excluded view serves as the test set. The results, presented in Figure 10, show



Test View \ Training View	-90°	-75°	-60°	-45°	-30°	-15°	0°	15°	30°	45°	60°	75°	90°
-90°	100.0±0.0	89.1±0.9	88.7±1.1	80.3±1.8	81.9±1.9	71.6±1.0	68.0±0.9	63.9±0.8	71.2±1.3	76.0±0.9	79.9±0.8	81.6±0.7	87.1±1.4
-75°	92.2±1.0	100.0±0.0	94.7±0.6	86.6±0.8	81.6±0.7	77.0±1.6	72.6±1.4	67.6±1.1	71.7±1.2	73.6±1.4	77.0±0.7	83.0±0.9	80.5±0.6
-60°	88.3±0.9	90.4±1.1	100.0±0.0	93.8±1.4	91.9±0.9	87.4±1.2	75.8±1.3	76.9±0.9	82.4±1.5	85.5±1.7	86.9±1.0	80.4±1.5	76.4±1.4
-45°	81.6±1.4	86.9±0.8	94.8±1.3	100.0±0.0	96.1±1.3	89.2±1.0	72.7±1.4	72.1±1.5	77.2±0.9	87.7±1.2	83.1±0.9	80.2±1.6	70.9±1.5
-30°	79.9±0.7	84.3±0.8	93.2±1.5	96.1±1.5	100.0±0.0	97.0±1.4	88.7±0.7	89.4±1.0	91.8±0.5	85.2±0.9	80.0±0.9	77.8±1.1	68.4±0.9
-15°	68.4±0.9	74.2±0.5	86.4±1.5	88.0±1.8	96.7±1.0	100.0±0.0	96.4±1.6	96.7±1.5	90.4±1.6	86.4±1.4	81.1±1.8	74.7±1.9	69.8±0.6
0°	63.4±1.3	70.2±0.8	75.8±0.4	79.4±1.7	90.1±0.9	96.4±0.8	100.0±0.0	100.0±0.0	90.4±1.6	83.5±1.2	78.1±1.2	71.4±1.1	66.7±0.9
15°	59.6±0.7	62.9±1.5	73.8±1.2	81.0±0.8	86.1±1.0	91.2±1.6	94.7±1.1	100.0±0.0	96.7±1.5	90.7±1.8	85.1±0.8	79.4±1.4	66.6±1.6
30°	72.4±0.6	74.2±1.7	81.4±1.4	85.0±1.5	89.7±1.0	80.0±1.3	83.4±1.1	97.7±0.5	100.0±0.0	96.4±0.6	90.7±1.5	85.0±1.2	81.4±1.8
45°	77.4±1.0	81.2±1.2	85.8±0.6	89.4±1.2	86.1±1.6	80.4±1.1	87.6±0.9	87.1±0.6	92.0±1.4	100.0±0.0	96.1±1.2	88.4±0.8	82.0±1.2
60°	69.5±0.9	78.9±1.1	81.1±0.9	77.0±0.8	70.1±1.4	62.2±1.3	73.7±1.6	80.1±1.5	89.2±1.0	92.7±0.9	100.0±0.0	100.0±0.0	89.7±0.8
75°	81.4±1.4	85.2±1.1	77.4±1.2	74.0±0.9	68.7±0.5	67.0±1.8	59.4±1.3	70.3±1.2	83.1±1.0	86.0±0.8	93.3±1.3	100.0±0.0	87.0±1.6
90°	85.8±0.9	83.2±1.0	75.8±1.4	72.4±1.6	63.1±1.3	57.4±1.4	66.0±1.1	74.1±0.7	83.4±1.4	87.0±1.6	90.1±1.4	96.4±1.3	100.0±0.0

FIGURE 9. Heatmap visualization of cross-view recognition accuracy of LE-MvDA on the Multi-PIE dataset, reported as mean ± standard deviation over five folds. Each cell indicates the rank-1 classification accuracy (%) across all pairwise combinations of camera views.



Test Views	MCCA	MvDA	pc-MvDA	MvDA-vc	LE-MvDA
-90°	70.3±1.4	73.9±1.3	76.5±1.2	78.4±1.4	86.8±0.7
-75°	83.7±0.9	84.2±1.1	87.7±1.9	88.5±1.4	91.3±1.6
-60°	73.2±0.8	75.8±1.5	77.4±0.9	80.1±0.8	88.6±0.8
-45°	82.6±0.8	84.7±0.6	86.5±1.6	89.3±1.0	91.9±0.9
-30°	73.1±1.6	75.3±0.7	76.2±0.4	78.8±1.5	87.1±1.1
-15°	79.5±1.8	82.4±0.8	85.1±1.7	85.8±1.3	89.1±0.7
0°	81.7±1.4	84.0±0.6	86.4±0.8	88.4±1.2	91.6±1.9
15°	73.9±1.7	74.1±0.9	77.8±1.6	80.1±1.5	88.2±1.5
30°	80.7±0.9	82.9±1.4	84.9±0.5	86.7±1.6	91.5±0.9
45°	75.1±1.9	77.1±1.3	77.5±1.2	82.3±1.3	87.4±0.8
60°	74.2±1.5	75.3±1.6	75.6±1.1	79.6±1.4	87.7±1.4
75°	81.4±1.8	83.2±0.3	87.2±0.9	89.2±0.5	92.7±0.9
90°	73.2±1.1	77.6±0.9	76.7±1.4	82.9±1.4	88.3±0.9

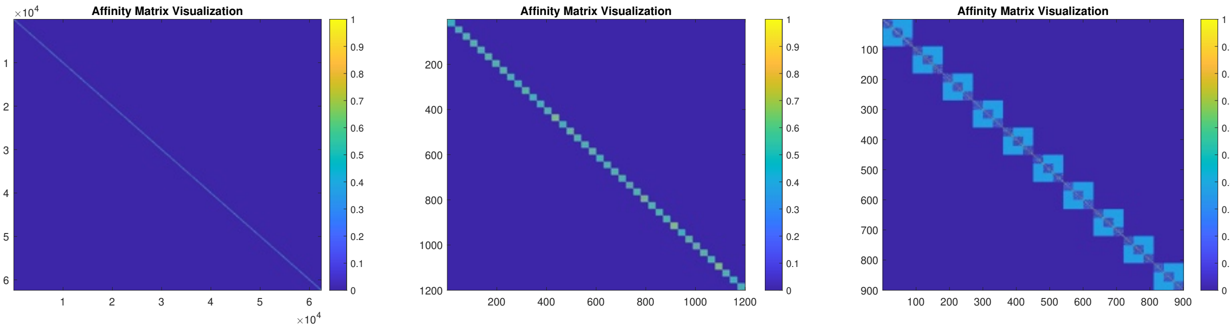
FIGURE 10. Heatmap visualization of classification accuracy for LE-MvDA and baseline methods across various testing view angles on the Multi-PIE dataset. Each column represents a method, and each row corresponds to a test view. For each row, the model is trained on all views except the one being tested, enabling evaluation of cross-view generalization. Results are reported as mean ± standard deviation over five folds.

that MvDA-vc outperforms other methods. However, our proposed LE-MvDA method achieves up to an 8.5% improvement over MvDA-vc, demonstrating its ability to find a more discriminative common space for feature representation. In Figure 11(a), the affinity matrix for the Multi-PIE dataset (comprising 240 classes, 13 views, and 20 samples per class, totaling $62,400$ samples) exhibits a prominent diagonal structure segmented into smaller blocks. Each block along the diagonal corresponds to samples from the same class and view, showing high similarity due to strong intra-class and intra-view coherence. These bright yellow blocks indicate high affinity values (close to 1), suggesting that samples within each block are highly similar in the feature space. Conversely, the off-diagonal regions appear mostly dark blue, representing low similarity between samples from different classes or views. This sharp contrast between high and low affinity regions highlights the ability of the locality-preserving construction to capture both global and local neighborhood structures. The affinity matrix is constructed within the LE-MvDA framework using local scaling [19], which adapts the affinity values based on neighborhood density, thereby enhancing the matrix's ability to reflect both global and local data structures. This construction plays a key role in guiding the discriminative projection learning process.

As shown in Table 8, both MvDA-vc and LE-MvDA outperform the other methods when evaluated on the entire dataset using a conventional 70% training and 30% testing split, rather than in pairwise or view-specific settings. For creating the training and testing splits, 70% of the images for each subject and pose (i.e., 14 samples per subject per pose, totaling $240 \times 13 \times 14$ images) were randomly selected for the training set, while the remaining 30% (i.e., 6 samples per subject per pose, totaling $240 \times 13 \times 6$ images) were reserved for the test set. Notably, LE-MvDA achieves a 3.4% improvement over MvDA-vc, underscoring its effectiveness in learning robust representations across the full multi-view distribution.

2) Experimental results on ORL dataset

To further assess the effectiveness of multi-view learning under both pairwise and multi-representation scenarios, we conducted experiments on the ORL dataset. Several baseline methods were evaluated alongside our proposed approach to highlight its advantages and quantify performance improve-



(a) (b) (c)

FIGURE 11. Affinity matrices of three datasets, illustrating the block-diagonal structure that indicates high intra-class and intra-view similarity, while off-diagonal elements reflect weaker affinities across different classes and views. (a) Multi-PIE dataset, comprising 240 classes, 13 views, and 20 samples per class, for a total of $62,400$ samples. (b) ORL dataset, comprising 40 classes, 3 views, and 20 samples per class, totaling $1,200$ samples. (c) RoboMNIST dataset, containing 10 classes, 3 views, and 30 samples per class, for a total of 900 samples.

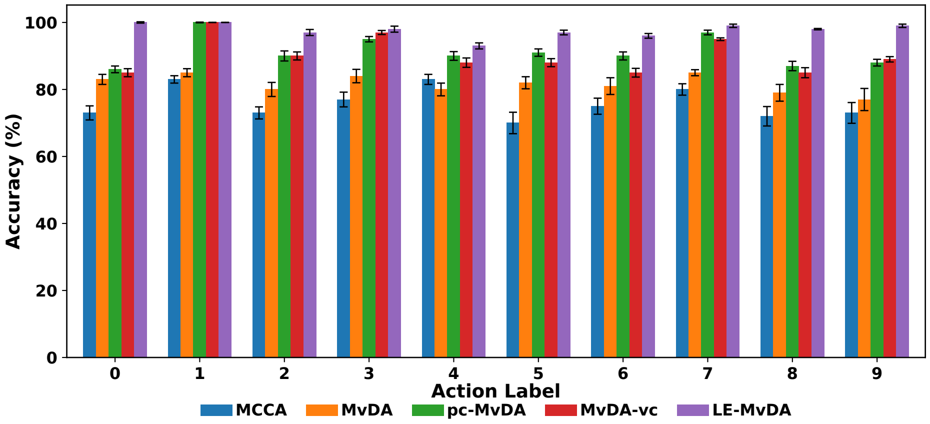


FIGURE 12. Classification accuracy for each action label on the RoboMNIST dataset, comparing the proposed LE-MvDA with baseline methods. Error bars denote the standard deviation over 10 repetitions, reflecting performance consistency. The x-axis indicates action labels, and the y-axis shows accuracy (%).

TABLE 2. Rank-1 recognition accuracy (%) of MvDA on the ORL dataset, evaluated across all pairwise combinations of feature representations. Results are reported as mean ± standard deviation over five folds.

Test	Train		
	Original view	FFT	Canny Edges
Original view	$100.0 \pm 0.0\%$	$60.5 \pm 1.1\%$	$64.9 \pm 1.9\%$
FFT	$59.5 \pm 1.4\%$	$100.0 \pm 0.0\%$	$57.0 \pm 1.3\%$
Canny Edges	$56.5 \pm 1.2\%$	$55.0 \pm 1.8\%$	$100.0 \pm 0.0\%$

TABLE 3. Rank-1 recognition accuracy (%) of LE-MvDA on the ORL dataset, evaluated across all pairwise combinations of feature representations. Results are reported as mean ± standard deviation over five folds.

Test	Train		
	Original view	FFT	Canny Edges
Original view	$100.0 \pm 0.0\%$	$68.7 \pm 0.8\%$	$67.9 \pm 1.0\%$
FFT	$75.0 \pm 1.1\%$	$100.0 \pm 0.0\%$	$63.7 \pm 0.9\%$
Canny Edges	$75.0 \pm 1.3\%$	$59.6 \pm 1.6\%$	$100.0 \pm 0.0\%$

ments in multi-representation learning. The results are presented in Tables 2, 3, and 4. Table 2 shows that while MvDA achieves 100% recognition when trained and tested on the same feature representation (or equivalently, the same view), its performance declines substantially in cross-representation settings. Table 3 demonstrates that LE-MvDA consistently outperforms MvDA, particularly in cross-view scenarios, with stronger generalization and consistently higher recognition rates, while still achieving 100% recognition on the same feature representation. Beyond pairwise comparisons, Table 4 compares LE-MvDA with other baseline methods across various training and testing feature representation combinations on the ORL dataset. The results confirm that LE-MvDA consistently outperforms all baseline methods, achieving up to 89.7% accuracy and underscoring its superior generalization capability and robust performance.

Figure 11(b) presents the affinity matrix for the ORL dataset (with 40 classes, 3 views, and 20 samples per class), which also demonstrates a clear block-diagonal pattern. As discussed in Figure 11(a), the diagonal blocks reflect high intra-class and intra-view similarity, while the darker off-diagonal areas confirm low cross-class and cross-view affinity. Despite the smaller scale of ORL compared to Multi-PIE, the locality-preserving affinity structure remains effective, capturing the essential class-level relationships required for subspace learn-

TABLE 4. Comparison of classification performance between LE-MvDA and baseline methods across different feature representation pairs on the ORL dataset. Results are reported as rank-1 recognition accuracy (%) in the form of mean ± standard deviation over five folds for each test view.

Train	FFT, Canny Edges	Original View, Canny Edges	Original View, FFT
Test	Original View	FFT	Canny Edges
MCCA	66.5 ± 1.9%	60.0 ± 2.1%	68.8 ± 1.7%
MvDA	70.2 ± 2.3%	65.5 ± 1.8%	71.0 ± 2.0%
pc-MvDA	83.5 ± 1.4%	78.0 ± 2.2%	82.4 ± 1.5%
MvDA-vc	76.0 ± 1.6%	72.5 ± 2.0%	75.9 ± 1.9%
LE-MvDA	**89.7 ± 1.2%**	**85.0 ± 1.7%**	**87.5 ± 1.4%**

TABLE 5. Rank-1 recognition accuracy (%) of MvDA on the RoboMNIST dataset, evaluated across all pairwise combinations of camera views. Results are reported as mean ± standard deviation over five folds.

Test	**Train**		
	View 1	View 2	View 3
View 1	100.0 ± 0.0%	84.8 ± 1.4%	60.0 ± 1.6%
View 2	78.0 ± 0.8%	100.0 ± 0.0%	68.5 ± 1.9%
View 3	75.0 ± 1.1%	55.5 ± 1.5%	100.0 ± 0.0%

TABLE 6. Rank-1 recognition accuracy (%) of LE-MvDA on the RoboMNIST dataset, evaluated across all pairwise combinations of camera views. Results are reported as mean ± standard deviation over five folds.

Test	**Train**		
	View 1	View 2	View 3
View 1	100.0 ± 0.0%	95.0 ± 1.2%	94.7 ± 1.1%
View 2	95.7 ± 0.6%	100.0 ± 0.0%	92.3 ± 0.9%
View 3	95.0 ± 0.9%	91.0 ± 1.4%	100.0 ± 0.0%

TABLE 7. Comparison of classification accuracy between LE-MvDA and baseline methods across various training and testing view angle combinations on the RoboMNIST dataset. Results are reported as rank-1 recognition accuracy (%) in the format mean ± standard deviation over five folds.

Train	View 1, View 2	View 1, View 3	View 2, View 3
Test	View 3	View 2	View 1
MCCA	70.5 ± 1.3%	71.0 ± 1.1%	73.0 ± 1.8%
MvDA	77.0 ± 1.3%	78.5 ± 1.6%	80.0 ± 0.8%
pc-MvDA	80.0 ± 1.9%	81.5 ± 1.0%	81.5 ± 1.4%
MvDA-vc	78.5 ± 1.7%	79.0 ± 0.9%	80.1 ± 1.0%
LE-MvDA	**95.3 ± 0.6%**	**94.7 ± 1.5%**	**97.0 ± 1.2%**

TABLE 8. Overall classification performance of LE-MvDA and baseline methods on each benchmark dataset using a 70%/30% train–test split, averaged over 5-fold cross-validation. Results are reported as mean ± standard deviation.

Dataset	MCCA	MvDA	pc-MvDA	MvDA-vc	LE-MvDA
Multi-PIE	79.9±1.2%	84.9±1.5%	85.8±1.3%	89.4±1.1%	**92.8±0.9%**
ORL	72.1±1.6%	74.0±1.3%	82.0±1.1%	80.4±1.0%	**91.9±0.7%**
RoboMNIST	75.9±1.4%	81.6±1.2%	91.4±0.9%	90.2±0.8%	**97.7±0.5%**

ing.

As shown in Table 8, both pc-MvDA and LE-MvDA outperform the other methods when evaluated on the entire ORL dataset using a conventional 70% training and 30% testing split. Notably, LE-MvDA achieves a 9.9% improvement over pc-MvDA, demonstrating its superior ability to learn robust representations across different feature transformations.

3) Experimental results on RoboMNIST dataset

a: Comparison with Multi-View Subspace Learning Methods

To evaluate the effectiveness of traditional multi-view learning techniques, including both pairwise and multi-view approaches, we apply baseline methods to the RoboMNIST dataset. Tables 5, 6, and 7 present the rank-1 recognition rates for MvDA pairwise evaluation, LE-MvDA pairwise evaluation, and a comprehensive multi-view comparison of all methods, respectively. Notably, as shown in Table 7, LE-MvDA consistently outperforms all baselines, achieving up to 97.0% accuracy in multi-view training scenarios—representing a 15.5% absolute improvement over the best baseline (pc-MvDA)—and demonstrating superior generalization across the RoboMNIST dataset.

Table 8 reports overall accuracy on the whole dataset using a 70%/30% train–test split. For RoboMNIST, LE-MvDA achieves 97.7% accuracy, surpassing the best baseline by 6.3% and demonstrating strong generalization across the entire dataset.

Figure 11(c) presents the affinity matrix visualization for the RoboMNIST dataset, which comprises 10 classes and 3 views per class, yielding a total of 900 samples. The matrix reveals a clear block-diagonal structure, where each larger diagonal block corresponds to a distinct class, and the three smaller sub-blocks within each represent different views. This visually confirms that samples from the same class and view are highly similar in the learned space, while those from different classes or views remain dissimilar. The dark blue off-diagonal regions signify low affinity values, reinforcing the matrix's ability to preserve strong intra-class and intra-view coherence while minimizing inter-class and inter-view similarities.

In a separate analysis, we compare the classification accuracy of five baseline methods across ten action labels in the RoboMNIST dataset, as shown in Figure 12. The x-axis represents the action labels (0–9), while the y-axis shows the classification accuracy in percentage. Each action label is associated with five colored bars, representing the accuracy achieved by each method. The performance comparison reveals that LE-MvDA consistently outperforms the other methods, achieving high accuracy, often exceeding 90% across most action labels. MvDA-vc and pc-MvDA also exhibit strong performance, though they fall slightly behind LE-MvDA in some cases. In contrast, MCCA consistently performs the worst among the methods, with significantly lower accuracy, particularly for certain action labels.

In summary, LE-MvDA consistently outperforms all other methods, demonstrating strong robustness across various datasets. While pc-MvDA and MvDA-vc show competitive performance, they still lag slightly behind, with MCCA being the least effective method overall. Notably, on the RoboMNIST dataset, the performance gap between LE-MvDA and the other methods is even more pronounced compared to other datasets. This is likely due to LE-MvDA's superior handling of temporal data, such as video sequences, where it significantly outperforms the other approaches.

b: Comparison with Deep Learning Methods.

To contextualize the effectiveness of LE-MvDA, it is important to compare it against modern deep learning approaches for multi-view data. This comprehensive evaluation highlights the strengths and weaknesses of both LE-MvDA (a shallow subspace method) and deep learning models in multi-view scenarios through a dual-level comparison strategy. Our evaluation encompasses both paradigm-level baseline comparisons and state-of-the-art video understanding model comparisons to provide comprehensive insights. Initially, we compare LE-MvDA against vanilla CNN and Transformer models with the same preprocessing pipeline to establish fundamental baseline comparisons between shallow subspace learning methods and deep learning paradigms. These representative architectures demonstrate the inherent trade-offs between mathematically interpretable shallow methods and deep learning, providing valuable insights into scenarios where shallow methods offer advantages in interpretability, and computational efficiency. To provide a more rigorous evaluation, we further extend our comparison to include state-of-the-art end-to-end video understanding models such as SlowFast [23] and TimeSformer [24]. This dual-level comparison strategy allows us to demonstrate both the fundamental paradigmatic advantages of LE-MvDA and its competitive performance against current deep learning state-of-the-art, ensuring a balanced and comprehensive evaluation. All these models, alongside our proposed LE-MvDA method, were applied to the RoboMNIST dataset for comprehensive performance assessment.

For the baseline comparison, we selected widely adopted CNN and Transformer architectures which continue to serve as strong baselines in contemporary research. To ensure fair comparison, a consistent processing pipeline was adopted where motion data from the robotic arm was recorded as video sequences, analyzed using a pre-trained ResNet-50 model through DeepLabCut to extract high-level spatial pose representations, as detailed in Section IV-B. The resulting time-series pose data was used as input to both the CNN and Transformer models for classification, as well as to the LE-MvDA method for dimensionality reduction and classification, ensuring uniform evaluation criteria.

The evaluation focuses on several key performance metrics, including classification accuracy, model complexity, training runtime, and inference speed—defined as the time required to classify a single test sample. In the context of LE-MvDA, model complexity is quantified by the number of parameters in the learned projection matrix, training time reflects the computation for constructing scatter matrices and solving the generalized eigenvalue problem, and inference speed measures the time for subspace projection and k-NN classification. Notably, While training requires $\mathcal{O}(n^2)$ memory for affinity matrix construction, inference only requires storing the projection matrices and training samples.

As illustrated in Figure 13, the CNN model features two 1-D convolutional layers with a kernel size of 3, using 32 and 64 filters, respectively. Each is followed by max-pooling (pool size of 2) to downsample and extract key temporal features, with a 0.1 dropout applied after the second pooling layer to reduce overfitting. The output is flattened and passed through a dense layer with 128 neurons activated by the Rectified Linear Unit (ReLU) function, followed by a softmax layer for classification.The model was trained for 50 epochs.

We employed a Transformer-based model comprising four attention heads, a point-wise feedforward network with a depth of 64, and three Transformer encoder blocks, followed by a Multi-Layer Perceptron (MLP) classification head with two dense layers of 128 and 64 units. The feedforward network uses ReLU activation function. Additionally, a GlobalAveragePooling1D layer aggregates the output tensor from the Transformer encoders. The model was trained within 200 epochs, without extensive hyperparameter tuning. An overview of the architecture is presented in Figure 14.

As shown in the baseline comparison in Table 9, while the CNN achieves the highest accuracy (99.6%), LE-MvDA offers a compelling balance between accuracy and efficiency, attaining 97.7% accuracy with dramatically fewer parameters (30K vs. 917K for the CNN) and minimal training time (0.002 seconds vs. 934.08 seconds for CNN), making it highly suitable for resource-constrained applications.

Beyond the baseline comparison, we evaluate LE-MvDA against sophisticated video understanding models including SlowFast [23] and TimeSformer [24]. Unlike the vanilla CNN and Transformer models which served as paradigm-level baselines, these models represent current state-of-the-art in video analysis. SlowFast employs dual-pathway architecture to capture both spatial details and temporal dynamics, while TimeSformer applies self-attention across both spatial and temporal dimensions for video understanding. Both models were fine-tuned for our multi-view robotic motion classification task, using raw video sequences as input for end-to-end learning—unlike LE-MvDA, which relies on a pose-based pipeline. As shown in Table 10, TimeSformer achieves the highest accuracy (98.1%), while SlowFast reaches 97.2%. Remarkably, LE-MvDA delivers competitive performance (97.7%) that falls between these two state-of-the-art models while requiring 100× fewer parameters and achieving over $250,000\times$ faster training with 40× faster inference than TimeSformer. This demonstrates that LE-MvDA provides an excellent accuracy-efficiency trade-off, making it particularly valuable for applications where computational efficiency is paramount.

These results highlight the complementary nature of both approaches: deep learning models may be more suitable when maximum accuracy, large training datasets, complex non-linear modeling, or end-to-end learning from raw data

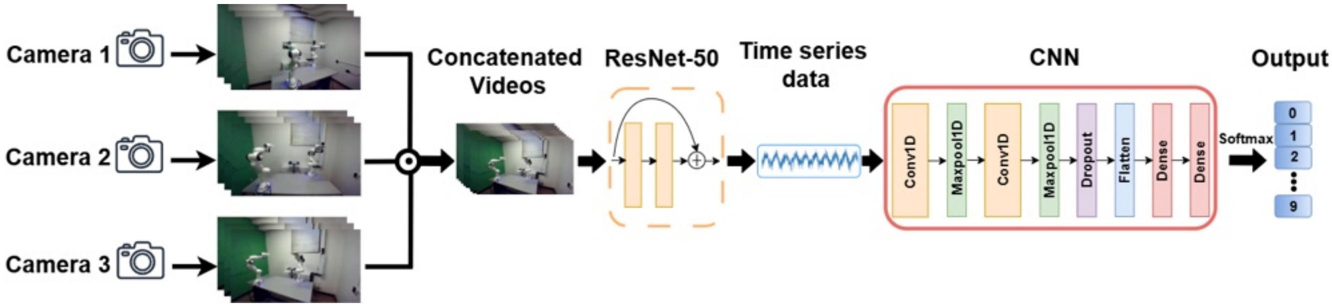


FIGURE 13. Architecture of the CNN model for action recognition, illustrating the processing pipeline from multi-view input frames through a ResNet-50 feature extractor and convolutional layers to final action classification.

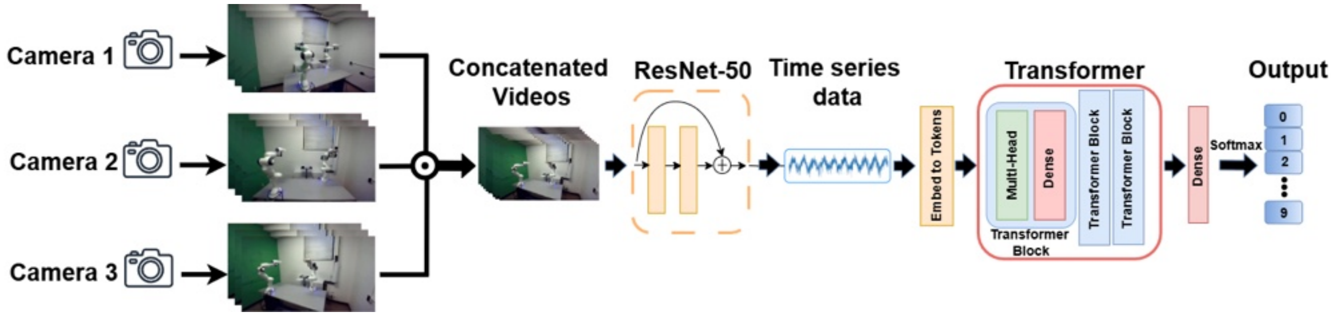


FIGURE 14. Architecture of the Transformer-based model for action recognition, illustrating the processing pipeline from multi-view input frames through a ResNet-50 feature extractor and Transformer encoder blocks to the final action classification.

TABLE 9. Classification accuracy (%) of LE-MvDA, CNN, and Transformer models on RoboMNIST using a 70%/30% train–test split, averaged over 5-fold cross-validation. Results are reported as mean $\pm$ standard deviation

Model	Acc.	# Param	Train Time	Infer Time
CNN	$\mathbf{99.6 \pm 0.2}$%	917,162	934.08 s	19.09 ms
Transformer	97.0 ± 1.7%	84,526	1800.50 s	60.04 ms
LE-MvDA	97.7 ± 0.5%	**30,000**†	**0.002** s	**0.040** ms

†Calculated as 100×300, where 100 is the PCA-reduced input dimension (k_{PCA}) and 300 is the subspace dimension (r).

TABLE 10. Classification accuracy (%) of LE-MvDA and state-of-the-art end-to-end deep learning models on RoboMNIST using a 70%/30% train–test split, averaged over 5-fold cross-validation. Results are reported as mean $\pm$ standard deviation.

Model	Acc.	# Param	Train Time	Infer Time
SlowFast	97.2 ± 0.4%	956,290	1824.25 s	3.17 ms
TimeSformer	$\mathbf{98.1 \pm 1.3}$%	3,060,128	517 s	1.64 ms
LE-MvDA	97.7 ± 0.5%	**30,000**†	**0.002** s	**0.04** ms

†Calculated as 100×300, where 100 is the PCA-reduced input dimension (k_{PCA}) and 300 is the subspace dimension (r).

are prioritized. Conversely, shallow subspace methods like LE-MvDA offer superior advantages for small-to-medium datasets, where competitive performance at minimal computational cost, along with efficiency, and interpretability are critical considerations.

Additionally, LE-MvDA offers better interpretability through its explicit projection matrices, which allow direct analysis of feature importance by examining projection weights, enable visualization of which original features contribute most to class separation across different views, and provide mathematical transparency in how the dimensionality reduction process transforms the data. This contrasts with deep learning models where feature transformations occur through multiple non-linear layers that are difficult to interpret directly.

D. ABLATION STUDIES AND PARAMETER ANALYSIS

To comprehensively evaluate the design choices and parameter sensitivity of LE-MvDA, we conduct a comprehensive ablation study focusing on three aspects: different affinity matrix constructions methods, the number of nearest neighbors K, and the effect of PCA dimensionality on classification performance. These studies provide insights into the method's robustness and guide optimal parameter selection across different datasets.

TABLE 11. Classification accuracy (%) across different affinity matrix construction methods on Multi-PIE, ORL, and RoboMNIST datasets.

Affinity Construction Method	Multi-PIE	ORL	RoboMNIST
Locally Scaled Gaussian (Proposed) [19]	**92.8**	**91.9**	**97.7**
Standard Heat Kernel ($\sigma = 1.0$) [40]	89.4	88.4	95.1
Sparse Heat Kernel ($K = 7, \sigma = 1.0$) [41]	89.5	89.0	92.9
Binary K-NN ($K = 7$) [41]	90.6	90.1	91.3

a: Affinity Matrix Construction Analysis

We compare four commonly used graph construction strategies in subspace learning:

- **Locally Scaled Gaussian Kernel (Proposed):** Employs adaptive local scaling parameters σ_i computed from the distance to the K-th nearest neighbor, as defined

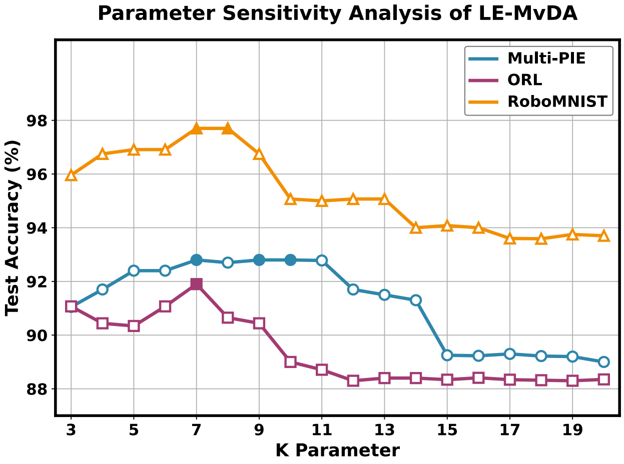


FIGURE 15. Sensitivity analysis of the K parameter in k-nearest neighbor affinity matrix construction for LE-MvDA. Classification accuracy is evaluated as a function of neighborhood size across three benchmark datasets. Filled markers indicate optimal K values achieving peak performance.

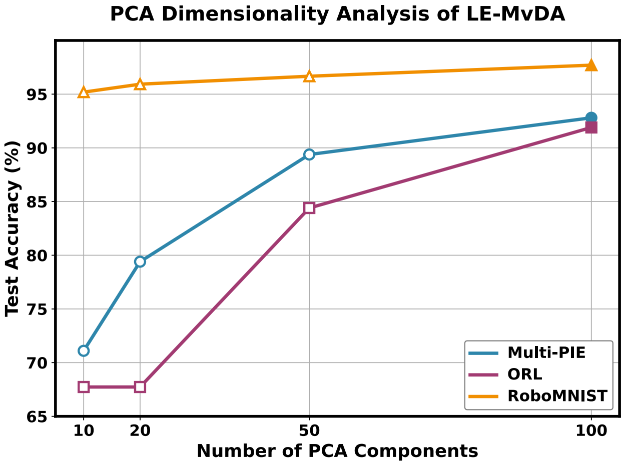


FIGURE 16. Impact of PCA dimensionality on LE-MvDA classification performance. Accuracy trends are shown as a function of retained principal components across three benchmark datasets, illustrating the trade-off between dimensionality reduction and information preservation.

in Equation (8). This approach adapts to local density variations in the feature space [19].

- **Standard Heat Kernel:** Utilizes a fixed global bandwidth parameter σ shared across all data samples, following the conventional Gaussian kernel formulation $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$ [40].
- **Sparse Heat Kernel:** Combines k-nearest neighbor sparsification with Gaussian weighting, applying the heat kernel only to the K nearest neighbors while setting all other connections to zero [41].
- **Binary K-NN Graph:** Uses binary connectivity weights, assigning unit weight to the K nearest neighbors and zero otherwise, representing the simplest graph construction approach [41].

Table 11 presents the classification performance of the affinity matrix construction methods across all three datasets. The locally scaled Gaussian kernel consistently achieves the highest accuracy, demonstrating its effectiveness in capturing meaningful local neighborhood structures across diverse data distributions.

b: K-Nearest Neighbor Parameter Sensitivity

We further investigate the impact of the K parameter in the affinity matrix by evaluating LE-MvDA performance across a range of K values (Figure 15). Accuracy peaks around $K = 7$ for all three datasets, with RoboMNIST reaching 97.7%, Multi-PIE 92.8%, and ORL 91.9%. These values represent the highest observed performance and are highlighted in Figure 15. Beyond $K \approx 15$, performance stabilizes, indicating convergence and robustness to this hyperparameter. These results suggest that $K = 7$ offers an effective trade-off between classification accuracy and computational efficiency.

c: Impact of PCA Dimensionality

Figure 16 examines how PCA dimensionality affects LE-MvDA performance across datasets. RoboMNIST demonstrates consistent accuracy above 95% regardless of component count, indicating that temporal motion patterns are efficiently captured in lower dimensions. Conversely, facial recognition datasets (Multi-PIE and ORL) show substantial improvement with increased dimensionality: Multi-PIE rises from 71.1% to 92.8% and ORL from 67.7% to 91.9% when expanding from 10 to 100 components. These findings suggest that while RoboMNIST's temporal dynamics are effectively captured in lower-dimensional spaces, the spatial complexity of face recognition tasks in ORL and Multi-PIE necessitates higher-dimensional embeddings to retain discriminative information. Overall, the results establish 100 PCA components as the optimal configuration, offering a strong trade-off between computational efficiency and classification performance in the LE-MvDA framework.

V. LIMITATIONS AND FUTURE DIRECTIONS

While LE-MvDA demonstrates significant advantages in computational efficiency and competitive accuracy, it also presents several methodological limitations. One notable limitation is the need to tune hyperparameters, such as the number of neighbors K and the PCA dimension k_{PCA}, via cross-validation, which may increase computational cost and sensitivity to dataset-specific characteristics. More critically, LE-MvDA's main limitation is the $\mathcal{O}(n^2)$ memory requirement during training due to affinity matrix construction, where n represents the total number of samples. This quadratic scaling in both memory and computation becomes prohibitive for large datasets—particularly in high-dimensional spaces—and limits scalability to datasets with hundreds of thousands of samples. The primary computational bottleneck lies in computing pairwise distances between samples. To address these limitations, future extensions could incorporate techniques such as approximate nearest neighbor search, sparse affinity graphs, or mini-batch affinity updates to improve scalability.

Additionally, several research directions could broaden the applicability of LE-MvDA. Another promising direction is to adapt LE-MvDA to semi-supervised or unsupervised learning settings, where labeled data is scarce. Techniques such as pseudo-labeling, view-consistent regularization, or graph-based label propagation could be integrated into the affinity-based formulation to support label-efficient learning. Moreover, developing online variants of LE-MvDA would enable real-time adaptation to streaming multi-view data. Another valuable extension involves applying the method to more diverse domains, such as multimodal sensing or cross-view retrieval, to further evaluate its generalization and robustness in practical applications.

A. SCOPE OF APPLICABILITY

When to use LE-MvDA:

- Small-to-medium datasets (n < 100K samples)
- Real-time inference requirements (< 1ms)
- Need for model interpretability
- Limited computational resources (edge devices)

When to prefer deep learning:

- Large-scale datasets (n > 1M)
- Raw sensory input without feature extraction
- End-to-end learning requirements
- Sufficient computational resources

VI. CONCLUSION

This paper introduced Locally Enhanced Multi-view Discriminant Analysis (LE-MvDA), a supervised dimensionality reduction method for multi-view classification that combines the discriminative power of MvDA with the structure-preserving properties of LPP. By leveraging adaptive affinity matrix construction, LE-MvDA preserves local neighborhood relationships while enhancing class separability, particularly for closely positioned classes.

Comprehensive experimental validation across three benchmark datasets confirms the effectiveness of our approach. LE-MvDA demonstrates substantial performance improvements, achieving 92.8% on Multi-PIE (3.4% improvement), 91.9% on ORL (9.9% improvement), and 97.7% accuracy on RoboMNIST (6.3% improvement). Unlike most existing multi-view methods that focus primarily on static data, our approach successfully captures temporal patterns in multi-view video sequences, making it particularly suitable for dynamic recognition tasks.

Comparative analysis with baseline deep learning models (CNN and Transformer) and state-of-the-art video understanding models (TimeSformer and SlowFast) highlights complementary strengths. While deep learning approaches may achieve slightly higher accuracy, LE-MvDA provides significant advantages in computational efficiency, including substantially fewer parameters, faster training and inference, and minimal hardware requirements. These features make it especially suitable for resource-constrained or real-time applications. Moreover, LE-MvDA offers enhanced interpretability through its explicit projection matrices, enabling direct analysis of feature importance and visualization of class separation across different views.

APPENDIX. THE DERIVATION OF $\mathbf{S}_{\mathcal{Z}}^{W}$ AND $\mathbf{S}_{\mathcal{Z}}^{B}$

$$
\begin{aligned}
\tilde{\mathbf{S}}_{\mathcal{Z}}^{W} &= \sum_{i=1}^{c}\sum_{j=1}^{v}\sum_{k:y_k=i}\left(\mathbf{z}_{ijk}-\frac{1}{n_i}\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)\left(\mathbf{z}_{ijk}-\frac{1}{n_i}\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)^{\top}\\
&= \sum_{i=1}^{c}\sum_{j=1}^{v}\sum_{k:y_k=i}\left[\mathbf{z}_{ijk}\mathbf{z}_{ijk}^{\top}-\frac{1}{n_i}\mathbf{z}_{ijk}\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}^{\top}\right.\\
&\quad -\frac{1}{n_i}\left(\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)\mathbf{z}_{ijk}^{\top}\\
&\quad \left.+\frac{1}{n_i^2}\left(\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)\left(\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)^{\top}\right]\\
&= \sum_{i=1}^{c}\sum_{j=1}^{v}\sum_{k:y_k=i}\left[\mathbf{z}_{ijk}\mathbf{z}_{ijk}^{\top}-\frac{1}{n_i}\left(\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)\mathbf{z}_{ijk}^{\top}\right]\\
&= \sum_{i=1}^{c}\sum_{j=1}^{v}\left[\sum_{k:y_k=i}\mathbf{z}_{ijk}\mathbf{z}_{ijk}^{\top}-\frac{1}{n_i}\left(\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)\left(\sum_{k:y_k=i}\mathbf{z}_{ijk}^{\top}\right)\right]\\
&= \sum_{i=1}^{c}\left[\sum_{j=1}^{v}\sum_{k:y_k=i}\mathbf{z}_{ijk}\mathbf{z}_{ijk}^{\top}-\frac{1}{n_i}\left(\sum_{j=1}^{v}\sum_{l:y_l=i}\mathbf{z}_{ijl}\right)\left(\sum_{j=1}^{v}\sum_{k:y_k=i}\mathbf{z}_{ijk}^{\top}\right)\right]\\
&= \sum_{i=1}^{c}\left(\sum_{j=1}^{v}\sum_{k:y_k=i}\mathbf{z}_{ijk}\mathbf{z}_{ijk}^{\top}-\frac{1}{n_i}\sum_{j=1}^{v}\sum_{r=1}^{v}\sum_{k,l:y_k,y_l=i}\mathbf{z}_{ijl}\mathbf{z}_{irk}^{\top}\right)\\
&= \sum_{i=1}^{c}\sum_{j=1}^{v}\sum_{k:y_k=i}\mathbf{z}_{ijk}\mathbf{z}_{ijk}^{\top}-\sum_{i=1}^{c}\frac{1}{n_i}\sum_{j=1}^{v}\sum_{r=1}^{v}\sum_{k,l:y_k,y_l=i}\mathbf{z}_{ijl}\mathbf{z}_{irk}^{\top}\\
&= \sum_{i=1}^{c}\sum_{j=1}^{v}\mathbf{T}_j^{\top}\left(\sum_{k:y_k=i}\mathbf{x}_{ijk}\mathbf{x}_{ijk}^{\top}\right)\mathbf{T}_j\\
&\quad -\sum_{i=1}^{c}\frac{1}{n_i}\sum_{j=1}^{v}\sum_{r=1}^{v}\mathbf{T}_j^{\top}\left(\sum_{k,l:y_k,y_l=i}\mathbf{x}_{ijl}\mathbf{x}_{irk}^{\top}\right)\mathbf{T}_r\\
&= \sum_{j=1}^{v}\mathbf{T}_j^{\top}\left(\sum_{i=1}^{c}\sum_{k:y_k=i}\mathbf{x}_{ijk}\mathbf{x}_{ijk}^{\top}\right)\mathbf{T}_j\\
&\quad -\sum_{j=1}^{v}\sum_{r=1}^{v}\mathbf{T}_j^{\top}\left(\sum_{i=1}^{c}\sum_{k,l:y_k,y_l=i}\frac{1}{n_i}\mathbf{x}_{ijl}\mathbf{x}_{irk}^{\top}\right)\mathbf{T}_r\\
&= \sum_{j=1}^{v}\mathbf{T}_j^{\top}\left[\sum_{i=1}^{c}\sum_{k:y_k=i}\left(\sum_{l:y_l=i}\mathbf{A}_{k,l}^{T}\right)\mathbf{x}_{ijk}\mathbf{x}_{ijk}^{\top}\right]\mathbf{T}_j\\
&\quad -\sum_{j=1}^{v}\sum_{r=1}^{v}\mathbf{T}_j^{\top}\left(\sum_{i=1}^{c}\sum_{k,l:y_k,y_l=i}\mathbf{A}_{k,l}^{T}\mathbf{x}_{ijl}\mathbf{x}_{irk}^{\top}\right)\mathbf{T}_r\\
&= \sum_{j=1}^{v}\sum_{r=1}^{v}\mathbf{T_j}^{\top}\mathbf{S}_{j,r}\mathbf{T_r}
\end{aligned}
$$

$$= \left[\mathbf{T}_1^\top \, \mathbf{T}_2^\top \, \dots \, \mathbf{T}_v^\top\right] \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1v} \\ S_{21} & S_{22} & \dots & S_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ S_{v1} & S_{v2} & \dots & S_{vv} \end{pmatrix} \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_v \end{bmatrix}$$

$$\mathbf{S}_{jr} = \begin{cases} \sum_{i=1}^{c} \sum_{k:y_k=i} \left(\sum_{l:y_l=i} \mathbf{A}_{k,l}^T \right) \mathbf{x}_{ijk} \mathbf{x}_{ijk}^\top \\ \quad - \sum_{i=1}^{c} \sum_{k,l:y_k,y_l=i} \mathbf{A}_{k,l}^T \mathbf{x}_{ijl} \mathbf{x}_{irk}^\top, & j = r \\ \\ - \sum_{i=1}^{c} \sum_{k,l:y_k,y_l=i} \mathbf{A}_{k,l}^T \mathbf{x}_{ijl} \mathbf{x}_{irk}^\top. & \text{otherwise} \end{cases} \tag{11}$$

where the weight matrix $\mathbf{A}_{k,l}^T$ is given by:

$$\mathbf{A}_{k,l}^T = \begin{cases} \dfrac{\exp\left(-\frac{\|\mathbf{x}_k - \mathbf{x}_l\|^2}{\sigma_k \sigma_l}\right)}{n_i} & \text{if } y_k = y_l = i, \\ 0 & \text{if } y_k \neq y_l. \end{cases} \tag{12}$$

where σ_k and σ_l are local scaling parameters defined based on the distance to the K-th nearest neighbor, as described in Section III following [19].
Similarly, the between-class scatter matrix $\tilde{\mathbf{S}}_{\mathcal{Z}}^B$ [2] for the low-dimensional embedding from multiple views is formulated as follows:

$$\begin{aligned} \tilde{\mathbf{S}}_{\mathcal{Z}}^B &= \sum_{i=1}^{c} n_i \left(\mu_i - \mu\right) \left(\mu_i - \mu\right)^\top \\ &= \sum_{i=1}^{c} n_i \mu_i \mu_i^\top - \sum_{i=1}^{c} n_i \mu_i \mu^\top - \sum_{i=1}^{c} n_i \mu \mu_i^\top + \sum_{i=1}^{c} n_i \mu \mu^\top \\ &= \sum_{i=1}^{c} n_i \mu_i \mu_i^\top - n \mu \mu^\top \\ &= \sum_{i=1}^{c} n_i \left(\frac{1}{n_i} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} \mathbf{z}_{ijk} \right) \left(\frac{1}{n_i} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} \mathbf{z}_{ijk} \right)^\top \\ &\quad - n \left(\frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} \mathbf{z}_{ijk} \right) \left(\frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} \mathbf{z}_{ijk} \right)^\top \\ &= \sum_{i=1}^{c} \frac{1}{n_i} \left(\sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} n_{ij} \mu_{ij} \right) \left(\sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} n_{ij} \mu_{ij} \right)^\top \\ &\quad - \frac{1}{n} \left(\sum_{j=1}^{v} \sum_{i=1}^{c} \sum_{k=1}^{n_{ij}} n_{ij} \mu_{ij} \right) \left(\sum_{j=1}^{v} \sum_{i=1}^{c} \sum_{k=1}^{n_{ij}} n_{ij} \mu_{ij} \right)^\top \\ &= \sum_{j=1}^{v} \sum_{r=1}^{v} \mathbf{T}_j^\top \left(\sum_{i=1}^{c} \frac{n_{ij} n_{ir}}{n_i} \mu_{ij}^{(x)} \mu_{ir}^{(x)\top} \right) \mathbf{T}_r \\ &\quad - \frac{1}{n} \sum_{j=1}^{v} \sum_{r=1}^{v} \mathbf{T}_j^\top \left(\sum_{i=1}^{c} n_{ij} \mu_{ij}^{(x)} \right) \left(\sum_{i=1}^{c} n_{ir} \mu_{ir}^{(x)} \right)^\top \mathbf{T}_r \\ &= \sum_{j=1}^{v} \sum_{r=1}^{v} \mathbf{T}_j^\top \mathbf{D}_{jr} \mathbf{T}_r \\ &= \left[\mathbf{T}_1^\top \mathbf{T}_2^\top \dots \mathbf{T}_v^\top\right] \begin{pmatrix} D_{11} & D_{12} & \dots & D_{1v} \\ D_{21} & D_{22} & \dots & D_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ D_{v1} & D_{v2} & \dots & D_{vv} \end{pmatrix} \begin{pmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_v \end{pmatrix} \end{aligned}$$

$$\mathbf{D}_{jr} = \left(\sum_{i=1}^{c} \frac{n_{ij} n_{ir}}{n_i} \mu_{ij}^{(x)} \mu_{ir}^{(x)\top} \right) - \frac{1}{n} \left(\sum_{i=1}^{c} n_{ij} \mu_{ij}^{(x)} \right) \left(\sum_{i=1}^{c} n_{ir} \mu_{ir}^{(x)} \right)^\top$$

ACKNOWLEDGMENT

This material is based upon work supported in part by the U.S. National Science Foundation under Grant Award 2208182, and in part by the U.S. DEVCOM Analysis Center and was accomplished under Contract Number W911QX-23-D0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the DEVCOM Analysis Center or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] H.-N. Tran, H.-Q. Nguyen, H.-G. Doan, T.-H. Tran, T.-L. Le, and H. Vu, "Pairwise-covariance multi-view discriminant analysis for robust cross-view human action recognition," *IEEE Access*, vol. 9, pp. 76097–76111, 2021.

[2] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, 2015.

[3] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2019.

[4] G. Chao, S. Sun, and J. Bi, "A survey on multiview clustering," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 146–168, 2021.

[5] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, 2017.

[6] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. 1991 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 586–587, 1991.

[7] H. Wang, X. Lu, Z. Hu, and W. Zheng, "Fisher discriminant analysis with L1-norm," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 828–842, Jun. 2014.

[8] X. He and P. Niyogi, "Locality preserving projections," *Adv. Neural Inf. Process. Syst.*, vol. 16, 2003.

[9] K. Jiang, Y. Song, S. Li, and K. Chen, "Feature extraction and recognition of face image based on 2DPCA with LDA algorithm," in *Proc. IEEE 7th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, vol. 7, pp. 1861–1866, 2024.

[10] R. Rahayu, A. Candra, *et al.*, "Combination of 2DPCA, sPCA, and Ridge Regression Model for Face Recognition," in *Proc. 2024 Int. Conf. on Electrical Engineering and Informatics (ICELTICs)*, 2024, pp. 130–135.

[11] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in Statistics: Methodology and Distribution*, Springer, 1992, pp. 162–190.

[12] A. A. Nielsen, "Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 293–305, 2002.

[13] A. Jordao, A. C. Nazare, J. Sena, and W. R. Schwartz, "Covariance-free partial least squares: An incremental dimensionality reduction method," *Pattern Recognit.*, vol. 104, p. 107305, 2020, doi: 10.1016/j.patcog.2020.107305.

[14] S. S. Shivagunde and V. V. Saradhi, "View incremental decremental multi-view discriminant analysis," *Applied Intelligence*, vol. 53, no. 11, pp. 13593–13607, 2023.

[15] S. S. Shivagunde, A. Nadapana, and V. V. Saradhi, "Multi-view incremental discriminant analysis," *Information Fusion*, vol. 68, pp. 149–160, 2021.

[16] J. Yin and S. Sun, "Multiview uncorrelated locality preserving projection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3442–3455, 2019.

[17] J. Xu, S. Yu, X. You, M. Leng, X.-Y. Jing, and C. L. P. Chen, "Multi-view hybrid embedding: A divide-and-conquer approach," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3640–3653, 2019.

[18] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs, "Generalized multi-view analysis: A discriminative latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2160–2167, 2012.

[19] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1601–1608.

[20] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN variants for computer vision: History, architecture,

application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, pp. 1097–1105, 2012.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, pp. 5998–6008, 2017.

[23] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 6202–6211.

[24] G. Bertasius, H. Wang, and L. Torresani, "Is space–time attention all you need for video understanding?," in *Proc. Int. Conf. Mach. Learn. (ICML)*, July 2021.

[25] L. Gong, H. Chen, Y. Chen, T. Yao, C. Li, S. Zhao, and G. Han, "DPNet: Dynamic Pooling Network for Accurate and Efficient Size-Aware Tiny Object Detection," *IEEE Internet of Things Journal*, 2025.

[26] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "EfficientFormer: Vision Transformers at MobileNet speed," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 12934–12949, 2022.

[27] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Boston, MA, USA: Academic Press, 1990.

[28] F. R. K. Chung, *Spectral Graph Theory*, vol. 92, CBMS Regional Conference Series in Mathematics. Providence, RI, USA: American Mathematical Society, 1997.

[29] Q. Wang et al., "New algorithms for trace-ratio problem with application to high-dimension and large-sample data dimensionality reduction," *Mach. Learn.*, vol. 110, pp. 1509–1544, 2021.

[30] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, May 2010, doi: 10.1016/j.imavis.2009.08.002.

[31] *The Database of Faces*, AT&T Laboratories Cambridge, Cambridge, UK, 2002. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[32] S. S. Shivagunde and V. V. Saradhi, "2D multi-view discriminant analysis," *Inf. Sci.*, vol. 586, pp. 391–407, 2022.

[33] J. W. Cooley, P. A. W. Lewis, and P. D. Welsh, "The fast Fourier transform and its application," *IEEE Trans. Educ.*, vol. E-12, pp. 27–34, 1969.

[34] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.

[35] K. Behzad, R. Zandi, E. Motamedi, H. Salehinejad, and M. Siami, "RoboMNIST: A Multimodal Dataset for Multi-Robot Activity Recognition Using WiFi Sensing, Video, and Audio," *Scientific Data*, vol. 12, no. 1, pp. 326, 2025.

[36] Z. Li, Y. Zhang, *et al.*, "A Comprehensive Review on Discriminant Analysis for Addressing Challenges of Class-Level Limitations, Small Sample Size, and Robustness," *Processes*, vol. 12, no. 7, p. 1382, July 2024, doi: 10.3390/pr12071382.

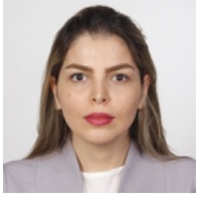
[37] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[40] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000, doi: 10.1109/34.868688.

[41] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003, doi: 10.1162/089976603321780317.



ELAHEH MOTAMEDI is a Ph.D. candidate in Electrical and Computer Engineering at Northeastern University, Boston, USA. She is a research assistant in the Siami Lab at Northeastern University. Her research focuses on dynamic motion representation and machine learning applications for robot action recognition.

MILAD SIAMI received his dual B.Sc. degrees in electrical engineering and pure mathematics from Sharif University of Technology in 2009, and his M.Sc. degree in electrical engineering from Sharif University of Technology in 2011. He received his M.Sc. and Ph.D. degrees in mechanical engineering from Lehigh University in 2014 and 2017, respectively. From 2017 to 2019, he was a postdoctoral associate at MIT Institute for Data, Systems, and Society. Since 2019, he has been with the Department of Electrical & Computer Engineering at Northeastern University, Boston, MA, where he is currently an Associate Professor.

His research interests include distributed control systems, distributed optimization, and sparse sensing, which is applied in robotics and cyber–physical systems. He has been recognized with several awards and fellowships, including a Gold Medal at the National Mathematics Olympiad in Iran, the Best Student Paper Award at the 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems, and the Rossin College Doctoral Fellowship at Lehigh University.