Precision Matrix based Feature Learning Mechanism for Subspace Clustering Task

Haohan Zou[1] **Alexander Cloninger**[1,2]

[1]Department of Mathematics, UC San Diego
[2]Halıcıoğlu Data Science Institute, UC San Diego
hazou@ucsd.edu, acloninger@ucsd.edu

Abstract

In recent studies, the *Average Gradient Outer Product* (AGOP) has emerged as a powerful tool to understand feature learning in deep neural networks, particularly in supervised learning tasks such as image classification. In this work, we extend this perspective to unsupervised learning, particularly the task of subspace clustering. Building on the existing kernel-based subspace clustering approaches, we introduce a feature learning mechanism which iteratively projects the training data onto an averaged precision matrix. Notably, the relevant feature learning matrix we derived is the inverse of the traditional AGOP matrix. We explain this from the viewpoint of isotropic variance control in the latent domain, and illustrate that the proposed projection mechanism refines the data distribution and orthogonalizes the data in the latent space. Empirically, we visualize the evolution of projected data distribution, kernel matrix, and the emergence of pronounced block-diagonal structure in affinity matrix on a toy example. Furthermore, our approach outperforms the state-of-the-art kernel-based subspace clustering method KTRR [Zhen et al., 2020] on the Extended Yale B dataset [Lee et al., 2005]. Full experiment implementation is available on Github.

1 Introduction

Unsupervised learning lies at the heart of understanding the geometric structure of high-dimensional data without relying on external supervision. Neural network-based unsupervised methods, such as variational autoencoders [Kingma and Welling, 2013] and contrastive learning frameworks [Chen et al., 2020, Tian et al., 2020], learn latent spaces that reflect meaningful geometric or semantic relationships inherent in the data by optimizing encoder and decoder parameters with respect to differentiable objectives that capture essential structural information. In parallel, kernel-based methods, such as kernel PCA [Schölkopf et al., 1997] or spectral clustering [Ng et al., 2001], obtain latent representations through an implicit feature mapping defined by a kernel function that measures pairwise similarities between samples. Despite their success, these approaches generally lack the mechanism to explicitly control or enforce the structural properties of the latent space, which limits their ability to select representative features aligned with the desired task objectives.

In the landscape of unsupervised learning, subspace clustering has emerged as an important task with a wide range of applications, such as image segmentation [Yang et al., 2008, Ma et al., 2007], motion segmentation [Vidal et al., 2009, Kanatani, 2001], and image clustering [Elhamifar and Vidal, 2013]. Under the assumption that high-dimensional data often lie approximately on a union of low-dimensional subspaces, the objective is to identify these underlying subspaces and cluster the data accordingly. Among the various methods proposed, self-expressiveness based methods [Vidal et al., 2009, Liu et al., 2010, Lu et al., 2012, Elhamifar and Vidal, 2013, Ji et al., 2014, Li and Vidal, 2015, You et al., 2016] have demonstrated remarkable effectiveness. These methods represent each

Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science(TAG-DS 2025).

data point as a linear combination of other points in data space or latent space, thereby encoding the pairwise relationships in the form of representation coefficients, which are then used to construct an affinity matrix. However, the performance depends critically on the quality of induced feature space.

The recent advances in *average gradient outer product* (AGOP) provide a new perspective on understanding feature learning in deep neural network, specifically for supervised learning. The AGOP, defined as the expected outer product of model's input gradients, captures how a model's predictions change with respect to perturbations in the input space. This object encodes crucial geometric information about the data manifold and the predictor's sensitivity, thereby serving as a powerful descriptor of learned representations. In Beaglehole et al. [2023], authors incorporated AGOP into kernel-based architectures and proposed a backpropagation-free models, which demonstrated strong feature extraction ability and improved generalization on vision benchmarks. In addition, Beaglehole et al. [2024] empirically and theoretical demonstrated that iteratively projecting data onto AGOP induce the emergence of *Deep Neural Collapse*: intra-class variability collapses while inter-class features form approximately orthogonal subspaces.

Building on these insights, we introduce the first AGOP-based feature learning algorithm for unsupervised learning problems. We specifically focus on subspace clustering and the reconstruction from a local neighborhood function, and design an iterative mechanism that projects the data onto an averaged precision matrix prior to applying nonlinear mapping. From the perspective of isotropic variance control in the latent domain, we analytically show that this projection explicitly reshapes the data distribution, making it more conducive to the self-supervised regression task and thereby improving the quality of constructed affinity matrix. We validate the effectiveness of our approach on synthetic and real-world datasets. Visualizations on a toy example demonstrate how iterative projections refine the data distribution and induce pronounced block-diagonal structure in the affinity matrix. On the challenging Extended Yale B dataset [Lee et al., 2005], our method outperforms the state-of-the-art Kernel Truncated Regression Representation (KTRR) approach, highlighting the benefits of proposed precision matrix based feature learning for subspace clustering.

Paper Organization. Section 2 reviews related studies on self-expressiveness based subspace clustering and AGOP in feature learning. Section 3 formalizes the subspace clustering problem and summarizes key properties of kernel-based subspace methods. Section 4 presents the motivation of proposed projection mechanism and its integration into kernel-based subspace clusteirng algorithms. Section 5 visualizes the block-diagonal structure in the affinity matrix induced by projection on a toy example and compares performance with state-of-art KTRR approach on the ExYaleB dataset.

2 Related Work

2.1 Self-expressiveness based Subspace Clustering

Among existing subspace clustering approaches, self-expressiveness based methods have emerged as the most popular and effective, achieving the state-of-the-art performance in tasks like image clustering and motion segmentation. The essence of self-expressiveness based methods is to first represent each data point as a (non)linear combination of other data points, and then construct the affinity matrix based on the resulting combination coefficients, which will be applied to normalized cuts [Shi and Malik, 2000] or spectral clustering [Ng et al., 2001] for membership assignment. Compared to alternative techniques, self-expressiveness based methods are typically more robust to noise and outliers due to the use of regularization terms that account for data corruptions. Moreover, by modeling relationships among all data points, self-expressiveness based methods capture the global structure of the data. Early works [Vidal et al., 2009, Liu et al., 2010, Lu et al., 2012, Elhamifar and Vidal, 2013, Ji et al., 2014, Li and Vidal, 2015, You et al., 2016] focused on the setting where data points lie in a union of linear subspace and constructed affinity matrix directly from the linear combination coefficients. To address cases where subspaces are inherently nonlinear, kernel-based extensions have been proposed [Patel et al., 2013, Patel and Vidal, 2014, Xiao et al., 2015, Yin et al., 2016, Zhen et al., 2020], which first map the data onto a higher-dimensional feature space—often a Reproducing Kernel Hilbert Space (RKHS)—where the subspaces may become linear, and then apply the same affinity matrix construction strategy and clustering membership assignment techniques. However, these approaches face a key limitation: there is no systematic way to select which kernels are most well-suited to the datasets for the affinity matrix construction.

2.2 Feature Learning via AGOP

The *Average Gradient Outer Product* (AGOP), formulated as

$$AGOP(f, \mathbf{X}) = \frac{1}{n} \sum_{x_i \in \mathbf{X}} \frac{\partial f(x_i)}{\partial x} \frac{\partial f(x_i)}{\partial x}^\top,$$

represents the uncentered covariance matrix of the input–output gradients of a predictor f averaged over the training data $\mathbf{X}$. Recent studies has leveraged AGOP to investigate the mechanisms of a range of surprising phenomena in neural networks, such as grokking, lottery tickets, simplicity bias, and adversarial examples [Radhakrishnan et al., 2024]. Extending this viewpoint, Beaglehole et al. [2023] proposed the *Deep Recursive Feature Machines* (Deep RFM), which incorporates layer-wise linear transformation with the AGOP into kernel machines. This backpropagation-free method not only improves the performance of convolutional kernels on vision datasets but also reproduces the edge-detection capabilities of convolutional neural networks. More recently, Beaglehole et al. [2024] provided both empirical and theoretical evidence that, beyond data-agnostic analyses, projecting data onto AGOP via Deep RFM offers a principled mechanism for the emergence of *Deep Neural Collapse* in supervised classification tasks, establishing that the occurrence of deep neural collapse can be explained through data-dependent setting with feature learning.

3 Preliminary: Subspace Clustering

Definition 1 (Sparse Subspace Clustering (SSC) [Elhamifar and Vidal, 2013])**.** Let $\{\mathcal{S}_\ell\}_{\ell=1}^n$ denote an arrangement of n subspaces in $\mathbb{R}^D$ with respective dimensions $\{d_\ell\}_{\ell=1}^n$. Consider a collection of N noise-free data points $\{x_i\}_{i=1}^N$ lying in the union of these subspaces. Define the data matrix:

$$\mathbf{X} \triangleq [x_1 \quad \cdots \quad x_N] = [\mathbf{X}_1 \quad \cdots \quad \mathbf{X}_n]\, \mathbf{\Gamma}, \tag{1}$$

where $\mathbf{X}_\ell \in \mathbb{R}^{D \times N_\ell}$ is a rank-d_ℓ matrix containing the $N_\ell > d_\ell$ points from subspace $\mathcal{S}_\ell$, and $\mathbf{\Gamma} \in \mathbb{R}^{N \times N}$ is an unknown permutation matrix. Neither the bases of the subspaces nor the subspace memberships of the data points are assumed to be known. The task of ***subspace clustering*** aims to find the number of subspaces, their dimensions, a basis for each subspace, and the segmentation of the data points in $\mathbf{X}$ according to their underlying subspaces.

Definition 2 (Self-expressiveness property [Elhamifar and Vidal, 2013])**.** If each subspace $\mathcal{S}_\ell$ is linear, point $x_i \in \{\mathcal{S}_\ell\}_{\ell=1}^n$ defined in Definition 1 can be efficiently reconstructed by a combination of other points in the dataset, i.e.

$$x_i = \mathbf{X} c_i = [x_1 \quad \cdots \quad x_N]\, c_i,$$

where $c_i \triangleq [c_{i1}, \cdots, c_{iN}]$ and the constraint $c_{ii} = 0$ eliminates the trivial solution of writing a point as a linear combination of itself. The data matrix $\mathbf{X}$ can be considered as a self-expressive dictionary in which each point can be written as a linear combination of other points.

Proposition 1 (Kernel Truncated Regression Representation (KTRR) [Zhen et al., 2020])**.** *For a given data set $\{x_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^m$, define a matrix $\mathbf{X} = [x_1, x_2, \cdots, x_n]$. Let $\phi : \mathbb{R}^m \to \mathcal{H}$ be a nonlinear mapping which transforms the input data from $\mathbb{R}^m$ into a kernel space $\mathcal{H}$, and denote $\phi(\mathbf{X}_i) = [\phi(x_1), \cdots, \phi(x_{i-1}), 0, \phi(x_{i+1}), \cdots, \phi(x_n)]$. For the objective function:*

$$\min_{c_i} \frac{1}{2} \|\phi(x_i) - \phi(\mathbf{X}_i) c_i\|_2^2 + \frac{\lambda}{2} \|c_i\|_2^2, \tag{2}$$

where λ is a positive real number controlling the strength of the ℓ_2-norm regularization, the optimal solution for Equation (2) *is:*

$$c_i = \left(\phi(\mathbf{X_i})^\top \phi(\mathbf{X_i}) + \lambda \mathbf{I}\right)^{-1} \phi(\mathbf{X}_i)^\top \phi(x_i). \tag{3}$$

c_i gives the optimal coefficients to reconstruct $\phi(x_i)$ as a linear combination of all other data in $\phi(\mathbf{X})$ except itself under ℓ_2-norm regularization.

For the purposes of this paper, we aim to learn the relevant features of the function $\phi(\mathbf{X}_i)c_i$. This is the prediction of where a point would lie in the latent space given its neighbors. For testing data, this expands to our function of interest,

$$f(x) = \phi(\mathbf{X}) c(x), \qquad c(x) = \left(\phi(\mathbf{X})^\top \phi(\mathbf{X}) + \lambda \mathbf{I}\right)^{-1} \phi(\mathbf{X})^\top \phi(x). \tag{4}$$

4 Proposed Method

4.1 Precision Matrix & Isotropic Variance control

Let $x \in \mathbb{R}^m$ denotes an input data, and consider a set of perturbed samples $\{z_i = x + \varepsilon_i\}_{i=1}^l$, where each perturbation $\varepsilon_i \sim \mathcal{N}(0, \mathbf{I}_m)$ is drawn from a standard isotropic Gaussian distribution. Let $f : \mathbb{R}^m \to \mathbb{R}^k$ be a mapping from input domain to output domain. By Taylor's theorem, the output at a perturbed input z_i admits a first-order approximation around x:

$$\begin{aligned} f(z_i) &= f(x) + \nabla f(x)^\top (z_i - x) + \mathcal{O}(\varepsilon_i^2), \\ &\approx f(x) + \nabla f(x)^\top \varepsilon_i, \end{aligned} \tag{5}$$

where $\nabla f(x) \in \mathbb{R}^{m \times k}$ denotes the Jacobian of f at x. Equation (5) implies that the perturbed outputs $\{f(z_i)\}_{i=1}^l$ are approximately distributed according to a Gaussian distribution in $\mathbb{R}^k$, centered at $f(x)$ with covariance $\Sigma = \nabla f(x)^\top \nabla f(x)$, i.e. $\mathcal{N}\left(f(x), \nabla f(x)^\top \nabla f(x)\right)$.

Now considering a self-supervised learning task, where the objective is to learn a mapping $g' : \mathbb{R}^m \to \mathbb{R}^m$ that preserves inputs, i.e. $g'(x) = x, \ \forall \ x \in \mathbb{R}^m$. For perturbed inputs $\{z_i\}$, which follow an isotropic Gaussian distribution around x, it is desirable that their mapped outputs $\{g'(z_i)\}$ also retain this isotropic structure. Specifically, after applying g', the perturbed samples should still approximately follow an isotropic Gaussian distribution centered at $g'(x)$, i.e. $g'(z_i) \sim \mathcal{N}(g'(x), \mathbf{I}_m)$.

Lemma 1. *Let $\lambda \in \mathbb{R}, \lambda > 0$ be a regularization parameter. Define $g' : \mathbb{R}^m \to \mathbb{R}^m$ be a continuously differentiable mapping and $\mathbf{P}(x) = \left(\nabla g'(x)^\top \nabla g'(x) + \lambda \mathbf{I}_m\right)^{-1}$. Denote $\Sigma := \nabla g'(0)^\top \nabla g'(0)$, and $\hat{g}(x) = g'(\mathbf{P}(x)^{\frac{1}{2}} x), \ \hat{g} : \mathbb{R}^m \to \mathbb{R}^m$ be a modified mapping. For perturbed inputs $z_i \sim \mathcal{N}(0, \mathbf{I}_m)$, $\hat{g}(z_i)$ evaluated via the first-order Taylor's approximation about origin gives:*

$$\hat{g}(z_i) \overset{approx.}{\sim} \mathcal{N}\big(g'(0), \ \hat{\Sigma}(\lambda)\big), \qquad \hat{\Sigma}(\lambda) := \left(\Sigma + \lambda \mathbf{I}_m\right)^{-\frac{1}{2}} \Sigma \left(\Sigma + \lambda \mathbf{I}_m\right)^{-\frac{1}{2}}.$$

Moreover, as regularization parameter $\lambda \to 0$, if Σ is symmetric positive definite, we have $\hat{\Sigma}(\lambda) \to \mathbf{I}_m$, and therefore the approximated distribution of $\{\hat{g}(z_i)\}$ converges to $\mathcal{N}(g'(0), \mathbf{I}_m)$. Otherwise, $\hat{\Sigma}(\lambda)$ will converge to some projection matrix as $\lambda \to 0$.

The proof of Lemma 1 is presented in Appendix A. The projection matrix $\mathbf{P}(x) = \left(\nabla g'(x)^\top \nabla g'(x) + \lambda \mathbf{I}\right)^{-1}$ is a regularized precision matrix (inverse covariance matrix), where the regularization term $\lambda \mathbf{I}$ avoids the case where covariance matrix $\nabla g'(x)^\top \nabla g'(x)$ is rank-deficient. Introducing such a projection operation enforces the data distribution around origin before and after the mapping to be approximately the same, a desirable property for the self-supervised learning task.

4.2 Feature Learning Mechanism to Subspace Clustering

Self-expressiveness based subspace clustering constructs the affinity matrix by solving a self-supervised regression task, where each data point is represented as a linear combination of the others, either in the data space $\mathbb{R}^m$ or a feature space $\mathcal{H}$. In Lemma 1, we illustrates that the proposed projection matrix $\mathbf{P}$ reshapes the data distribution more suitable for the self-supervised task. Motivated by this, we extend this projection mechanism to kernel-based subspace clustering algorithms, aiming to improve the quality of affinity matrix and obtain better clustering performance.

Suppose $\mathbf{X} = \{x_i\}_{i=1}^n$, $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a dataset for subspace clustering. Let $\phi : \mathbb{R}^m \to \mathcal{H}$ maps inputs onto a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. Define $\phi(\mathbf{X}_i) = [\phi(x_1), \cdots, \phi(x_{i-1}), 0, \phi(x_{i+1}), \cdots, \phi(x_n)]$. Let $f : \mathcal{H} \to \mathcal{H}$ reconstructs $\phi(x_i)$ from other samples $\phi(\mathbf{X}_i)$ as presented in Equation (4). In Lemma 1, the projection employs the precision matrix $\mathbf{P}(0) = \left(\nabla f(0)^\top \nabla f(0) + \lambda \mathbf{I}\right)^{-1}$ to refine the data distribution. However, in the feature space $\mathcal{H}$, the origin 0 is not well-defined. To address this, we approximate the local covariance $\nabla f(0)^\top \nabla f(0)$ via the empirical mean of local covariance at $f(\phi(x_i))$ for $i = 1, \cdots, n$.

Suppose data in $\phi(\mathbf{X})$ are uniformly sampled by random variable z from a union of subspace $\{\mathcal{S}_\ell\}$ in latent space such that $\mathbb{E}[z] = 0$. By Taylor's theorem, the covariance matrix at $f(z)$ can be

Algorithm 1 Precision Matrix-based Feature Learning Mechanism for Subspace Clustering

Input: Dataset $\mathbf{X}$, iteration number L, ridge parameter λ, and the number of subspaces k.
Output: The cluster labels of input data.

for all $j = 1, \cdots, L$ **do**
  Calculate kernel matrix $\mathbf{K}$.
  Obtain the mapping $f : f(\phi(x_i)) = \phi(\mathbf{X}_i)c_i^*$ where c_i^* is derived via Equation (3).
  Calculate the emprirical mean of local covariance: $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} \nabla f(\phi(x_i))^\top \nabla f(\phi(x_i))$.
  Perform SVD decomposition to $\hat{\Sigma}$: $U, D, V^\top = svd(\hat{\Sigma})$.
  Inverse singular values: $D^{-1} = \frac{D}{(D+\lambda)^2}$.
  Regularize inverse singular values: $D^{-1} = D^{-1}/\max(D^{-1})$.
  Take square root of inverse singular values: $D^{-\frac{1}{2}} = \sqrt{D^{-1}}$.
  Calculate square root averaged precision matrix: $\mathbf{P}^{\frac{1}{2}} = UD^{-\frac{1}{2}}V^\top$.
  Project the data: $\mathbf{X} = \mathbf{P}^{\frac{1}{2}}\mathbf{X}$.
  Normalize the data.
end for
For each point $x_i \in \mathbb{R}^m$, calculate the coefficients c_i^* following Equation (3).
Construct the affinity matrix $\mathbf{W} = |(\mathbf{C}^*)^\top| + |(\mathbf{C}^*)|$.
Apply spectral clustering algorithm [Ng et al., 2001] on $\mathbf{W}$ to obtain the clustering membership.

approximated up to first order as:

$$\begin{aligned}\nabla f(z)^\top \nabla f(z) \approx & \left(\nabla f(0) + \nabla^2 f(0)^\top z\right)^\top \left(\nabla f(0) + \nabla^2 f(0)^\top z\right), \\ \approx & \nabla f(0)^\top \nabla f(0) + \left((z^\top \nabla^2 f(0) \nabla f(0)\right) + \\ & \left(\nabla f(0)^\top \nabla^2 f(0)^\top z\right) + \left(z^\top \nabla^2 f(0) \nabla^2 f(0)^\top z\right).\end{aligned}$$

Taking expectation and substitute $\mathbb{E}[z] = 0$ yields:

$$\begin{aligned}\mathbb{E}[\nabla f(z)^\top \nabla f(z)] \approx & \mathbb{E}\left[\left(\nabla f(0) + \nabla^2 f(0)^\top z\right)^\top \left(\nabla f(0) + \nabla^2 f(0)^\top z\right)\right], \\ \approx & \nabla f(0)^\top \nabla f(0) + \mathbb{E}[z^\top]\nabla^2 f(0)\nabla f(0) \\ & \nabla f(0)^\top \nabla^2 f(0)^\top \mathbb{E}[z] + \mathbb{E}[z^\top](\nabla^2 f(0)\nabla^2 f(0)^\top)\mathbb{E}[z], \\ \approx & \nabla f(0)^\top \nabla f(0).\end{aligned}$$

By the law of large numbers, the expectation of $\nabla f(z)^\top \nabla f(z)$ can be approximated by the empirical mean of local covariances at $f(\phi(x_i))$ so that:

$$\nabla f(0)^\top \nabla f(0) \approx \mathbb{E}[\nabla f(z)^\top \nabla f(z)] \approx \sum_{i=1}^{n} \nabla f(\phi(x_i))^\top \nabla f(\phi(x_i)).$$

For instance, suppose ϕ is a quadratic kernel mapping, for data $y \in \mathbb{R}^m, y \notin \mathbf{X}$, the local covariance at $f(\phi(y))$ can be represented as:

$$\left(\sqrt{\phi(\mathbf{X})^\top \phi(y)} \circ 2\ \mathbf{X}^\top\right)^\top (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}\left(\sqrt{\phi(\mathbf{X})^\top \phi(y)} \circ 2\ \mathbf{X}^\top\right), \tag{6}$$

where $\mathbf{K} = (\mathbf{X}^\top\mathbf{X})^2$. Derivation of Equation (6) is shown in Appendix B. The kernel matrix and dataset can be augmented for the calculation of local covariance at $x_i \in \mathbf{X}$. The empirical mean of $\nabla f(\phi(x_i))^\top \nabla f(\phi(x_i))$ over dataset $\mathbf{X}$ gives an surrogate covariance $\hat{\Sigma}$ to approximate $\nabla f(0)^\top \nabla f(0)$, thereby the approximated projection matrix can be derived as $\hat{\mathbf{P}}(0) = (\hat{\Sigma} + \lambda\mathbf{I})^{-1}$.

Algorithm 1 summarizes the integration of this projection mechanism into kernel-based subspace clustering methods. Prior to calculating the affinity matrix used for membership assignment, our algorithm iteratively projects the data onto $\hat{\mathbf{P}}(0)$ to refine the data distribution in latent space, where the reconstruction mapping f is derived from the Equation (3), i.e., $f(\phi(x_i)) = \phi(\mathbf{X}_i)c_i^*$.

4.3 Extension of Variance Control to Data Space

Instead of enforcing isotropic variance in the latent domain, our proposed feature learning mechanism can be extended to data space to shape variance according to the underlying subspace geometry. Specifically, for data $x \in \mathbb{R}^m$, after obtaining the self-reconstructed point $f(\phi(x))$, we learn a linear mapping $g : \mathcal{H} \to \mathbb{R}^m$ such that the composite mapping $h(x) = g(f(\phi(x))) \approx x$ reconstructs the input in data space. Unlike the isotropic regularization in latent space, following the idea of AGOP, we design the projection to be a covariance matrix, which reflects the intrinsic directions of the underlying subspaces: high variance along within subspace directions and near-zero variance orthogonal to them. This yields an elliptical covariance structure that enchances subspace separability by preserving intra-subspace consistency while suppressing inter-subspace interference. For example, for quadratic kernel ϕ, the local covariance at $h(x)$ can be approximated in terms of $\mathbf{K}$ and $\mathbf{X}$:

$$\left(\sqrt{\phi(\mathbf{X})^\top \phi(x)} \circ 2\,\mathbf{X}^\top\right)^\top \hat{\mathbf{K}}\mathbf{X}^\top\mathbf{X}\hat{\mathbf{K}}\left(\sqrt{\phi(\mathbf{X})^\top \phi(x)} \circ 2\,\mathbf{X}^\top\right)$$

where $\hat{\mathbf{K}} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}$. The derivation is presented in Appendix C. The empirical mean of covariance at $h(x_i)$ averaged over $\mathbf{X}$ calculates the projection matrix. By adjusting the target covariance structure, such as isotropic, elliptical, or low-rank, the proposed mechanism provides a general framework for covariance shaping in both data space and latent space. This enables better alignment between the induced latent representation and the learning objective.

5 Numerical Experiment

5.1 A Motivating Toy Example

To build intuition on how the proposed precision matrix-based feature learning reshapes data representations to better support affinity matrix construction under the principle of self-expressiveness, we present a motivating toy example. The dataset consists of three clusters of data, each sampled from a distinct linear subspace in $\mathbb{R}^5$. The bases of the three linearly independent subspaces are given by:

$$\left\{\begin{bmatrix}1\\1.3\\0\\0\\0\end{bmatrix}, e_3, e_4, e_5\right\}, \quad \left\{\begin{bmatrix}1.3\\1\\0\\0\\0\end{bmatrix}, e_3, e_4, e_5\right\}, \quad \left\{\begin{bmatrix}\frac{5.2}{\sqrt{5}}\\\frac{1}{\sqrt{5}}\\0\\0\\0\end{bmatrix}, e_3, e_4, e_5\right\},$$

where e_i denotes the i-th standard basis. The top-left panel of Figure 1 visualizes the first two dimensions of this toy example. We applied our proposed feature learning framework in Algorithm 1 to this toy dataset using both a quadratic kernel and a Gaussian kernel. Figures 1 & 2 illustrate, across iterations, the evolution of i). the projected data distribution, ii). the kernel matrix $\mathbf{K}$, and iii). the affinity matrix (combination coefficient matrix) calculated via Equation 3.

For the quadratic kernel matrix $\mathbf{k}(x, y) = (x^\top y)^2$, the iterative projection gradually orthogonalizes the subspaces: after eight iterations, the kernel matrix exhibits a clear block-diagonal structure (middle row of Figure 1), and the affinity matrix reveals that nonzero coefficients occur almost exclusively among intra-cluster points (bottom row of Figure 1). A similar phenomenon is observed with the Gaussian kernel $\mathbf{k}(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$. The projection widens inter-cluster angles and increases pairwise Euclidean distances, thereby incurring block diagonal structure in both the kernel matrix and affinity matrix (middle & bottom rows of Figure 2).

This toy example highlights that our precision matrix based feature learning mechanism actively reshapes the data geometry: under the quadratic kernel by orthogonalizing subspaces, and under the Gaussian kernel by enlarging inter-cluster distances. In both cases, iterative projection induces a kernel matrix that are naturally aligned with self-expressiveness property, leading to the emergence of pronounced block-diagonal structure in affinity matrices.

5.2 Performance on ExYaleB Dataset

The Extended Yale B dataset [Lee et al., 2005] (ExYaleB) is a widely used benchmark to examine subspace clustering algorithms. It contains 38 subjects, each represented by 64 face images captured

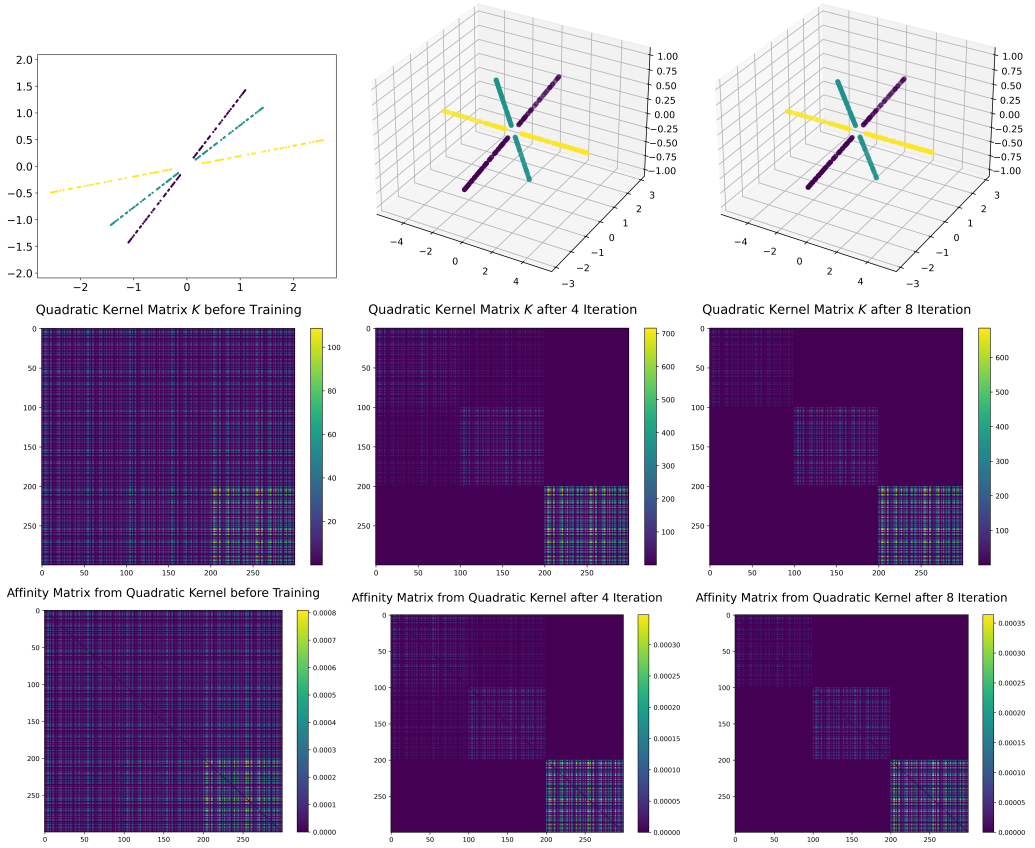

Figure 1: Across eight iterations of projection, the top row visualizes the evolution of projected data distribution, the middle row visualizes the evolution of *quadratic kernel matrix* $\mathbf{K}$, and the bottom row visualizes the evolution of affinity matrix obtained from Equation 3.

under varying illumination conditions. Under the Lambertian reflectance assumption, the images of a single subject with fixed pose under different lighting conditions lie in a low-dimensional non-linear subspace (or sub-manifold) with intrinsic dimension close to nine [Basri and Jacobs, 2003], making this dataset particularly suitable for subspace clustering. To reduce computational overhead, we down-sampled the original images from 192×168 pixels to 48×42 pixels.

We compared proposed framework against the Kernel Truncated Regression Representation (KTRR) approach [Zhen et al., 2020] on the Extended Yale B dataset. In our method, the Gaussian kernel matrix is iteratively updated for 10 iterations following Algorithm 1, while KTRR constructs its Gaussian kernel matrix directly following the Equation $\mathbf{k}(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$. For a fair comparison, both approaches adopt the same self-expressiveness based affinity construction approach and the same spectral clustering procedure used in [Elhamifar and Vidal, 2013].

To examine robustness with respect to the number of clusters, we consider $k \in \{10, 20, 30, 38\}$. Each subject contributes 64 face images. To obtain a manageable size of experiment, the k subjects test of ExYaleB dataset is selected as follows: 1). number subjects from 1 to 38, and 2). for each k, we take consecutive blocks of k subjects (e.g., for $k = 10$, we take the images from subject 1–10, 3–12, ..., 29–38, total 15 experimental trials). Each trail's performance is averaged over 5 seedings in K-means. Clustering quality is evaluated via two widely adopted metrics: normalized mutual information (NMI) [Cheng et al., 2010] and adjusted Rand index (ARI) [Hubert and Arabie, 1985]. Both metrics range from 0 (predicted labels totally mismatch with ground truth) to 1 (perfect clustering).

Table 1 summarizes the comparison results on the ExYaleB dataset. The proposed method consistently outperforms KTRR approach across all values of k, with roughly 10–25 percentage points improvement in NMI score and about 10–20 percentage points improvement in ARI score. This comparison

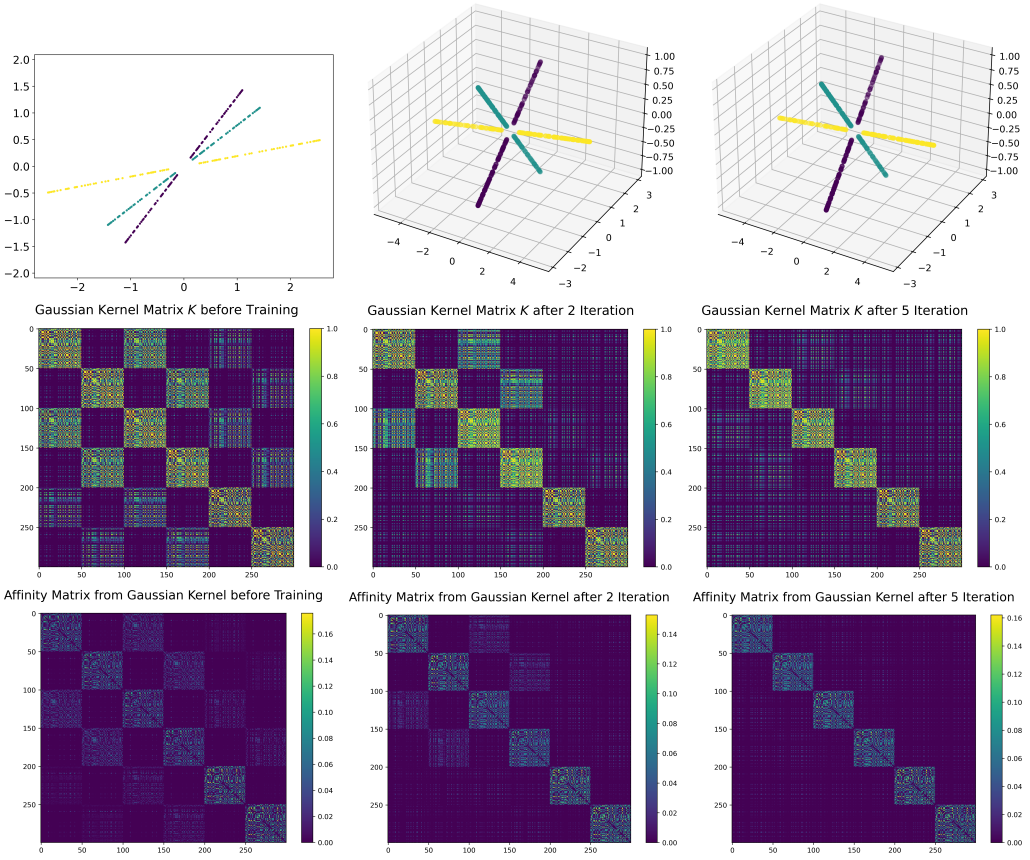

Figure 2: Across five iterations of projection, the top row visualizes the evolution of projected data distribution, the middle row visualizes the evolution of *Gaussian kernel matrix* $\mathbf{K}$, and the bottom row visualizes the evolution of affinity matrix obtained from Equation 3.

Table 1: Clustering Performance Comparison on the ExYaleB Dataset: Proposed Method V.S. KTRR.

Frameworks	$\mathbf{k = 10}$		$\mathbf{k = 20}$		$\mathbf{k = 30}$		$\mathbf{k = 38}$	
	NMI	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**
Ours	**0.6193**	**0.4200**	**0.6310**	**0.3851**	**0.5660**	**0.3092**	**0.5567**	**0.2808**
KTRR	0.3664	0.2062	0.4129	0.2037	0.4411	0.1939	0.4537	0.1905

highlights that the projection mechanism introduced in Lemma 1 produces kernel representations that better capture the intrinsic subspace structure that are important to the self reconstruction mapping, leading to better constructed self-expressiveness based affinity matrix and improved clustering performance on the real-world image clustering dataset.

6 Conclusion

In this paper, we propose a precision matrix-based feature learning mechanism for the subspace clustering task. Building on the existing kernel-based approaches, we introduce an iterative projection scheme which transforms the data distribution more suitable for the self-expressiveness based affinity matrix construction. We analytically demonstrate how the projection matrix refines the data distribution from the perspective of isotropic variance control in latent domain, and empirically visualize this innovative projection operation can induce pronounced block-diagonal structure in affinity matrices on a toy example. On the Extended Yale B dataset, with more suitable affinity matrix, our framework outperforms the KTRR method for experiments with different numbers of subjects.

References

Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003.

Daniel Beaglehole, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in convolutional neural networks. *arXiv preprint arXiv:2309.00570*, 2023.

Daniel Beaglehole, Peter Súkeník, Marco Mondelli, and Misha Belkin. Average gradient outer product as a mechanism for deep neural collapse. *Advances in Neural Information Processing Systems*, 37:130764–130796, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, and Thomas S. Huang. Learning with ℓ^1-graph for image analysis. *IEEE Transactions on Image Processing*, 19(4):858–866, 2010. doi: 10.1109/TIP.2009.2038764.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. doi: 10.1109/TPAMI.2013.57.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. In *IEEE Winter conference on applications of computer vision*, pages 461–468. IEEE, 2014.

Ken-ichi Kanatani. Motion segmentation by subspace separation and model selection. In *Proceedings Eighth IEEE International Conference on computer Vision. ICCV 2001*, volume 2, pages 586–591. IEEE, 2001.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on pattern analysis and machine intelligence*, 27(5): 684–698, 2005.

Chun-Guang Li and Rene Vidal. Structured sparse subspace clustering: A unified optimization framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 277–286, 2015.

Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010.

Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *European conference on computer vision*, pages 347–360. Springer, 2012.

Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE transactions on pattern analysis and machine intelligence*, 29 (9):1546–1562, 2007.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

Vishal M Patel and René Vidal. Kernel sparse subspace clustering. In *2014 ieee international conference on image processing (icip)*, pages 2849–2853. IEEE, 2014.

Vishal M Patel, Hien Van Nguyen, and René Vidal. Latent space sparse subspace clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 225–232, 2013.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

Ehsan Elhamifar René Vidal et al. Sparse subspace clustering. In *2009 IEEE conference on computer vision and pattern recognition (CVPR)*, volume 6, pages 2790–2797, 2009.

Shijie Xiao, Mingkui Tan, Dong Xu, and Zhao Yang Dong. Robust kernel low-rank representation. *IEEE transactions on neural networks and learning systems*, 27(11):2268–2281, 2015.

Allen Y Yang, John Wright, Yi Ma, and S Shankar Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.

Ming Yin, Yi Guo, Junbin Gao, Zhaoshui He, and Shengli Xie. Kernel sparse subspace clustering on symmetric positive definite manifolds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5157–5164, 2016.

Chong You, Daniel Robinson, and René Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3918–3927, 2016.

Liangli Zhen, Dezhong Peng, Wei Wang, and Xin Yao. Kernel truncated regression representation for robust subspace clustering. *Information Sciences*, 524:59–76, 2020.

A Proof for Lemma 1

Here is the proof for Lemma 1 presented in Subsection 4.1.

Proof. By the first-order Taylor expansion of $\hat{g}$ at origin:

$$\hat{g}(z_i) = \hat{g}(0) + \nabla\hat{g}(0)^\top z_i + \mathcal{O}(\|z_i\|_2^2).$$

Neglecting the higher order term ($\mathbb{E}(z_i) = 0, \mathrm{Cov}(z_i) = \mathbf{I}_m$), we have:

$$\begin{aligned}\hat{g}(z_i) &\approx \hat{g}(0) + \nabla\hat{g}(0)^\top z_i, \\ &\approx g'(0) + \nabla\hat{g}(0)^\top z_i.\end{aligned}$$

By the chain rule,

$$\nabla\hat{g}(0) = \nabla g'(0)\,\mathbf{P}(0)^{\frac{1}{2}} = \nabla g'(0)\big(\Sigma + \lambda\mathbf{I}_m\big)^{-\frac{1}{2}}.$$

It follows that the linear approximation of $\hat{g}(z_i)$ gives an affine transformation of the Gaussian distribution $z_i \sim \mathcal{N}(0, \mathbf{I}_m)$. Hence the approximated distribution of $\{\hat{g}(z_i)\}$ is Gaussian with mean $g'(0)$ and covariance

$$\begin{aligned}\hat{\Sigma}(\lambda) &= \big(\nabla g'(0)\mathbf{P}(0)^{\frac{1}{2}}\big)^\top\big(\nabla g'(0)\mathbf{P}(0)^{\frac{1}{2}}\big), \\ &= \mathbf{P}(0)^{\frac{1}{2}}\,\Sigma\,\mathbf{P}(0)^{\frac{1}{2}},\end{aligned}$$

which simplifies to

$$\hat{\Sigma}(\lambda) = \big(\Sigma + \lambda\mathbf{I}_m\big)^{-\frac{1}{2}}\,\Sigma\,\big(\Sigma + \lambda\mathbf{I}_m\big)^{-\frac{1}{2}}.$$

Since Σ is a symmetric (semi-)positive definite matrix, Σ has non-negative eigenvalues $\mu_1, \ldots, \mu_m$, and $v_1, \cdots, v_m$ are corresponding orthonormal eigenvectors. Thus,

$$\hat{\Sigma}(\lambda) = \sum_{j=1}^{m} \frac{\mu_j}{\mu_j + \lambda}\, v_j v_j^\top.$$

Suppose Σ is symmetric positive-definite, then for every $j = 1, \cdots, m$, we have $\dfrac{\mu_j}{\mu_j + \lambda} \to 1$ as $\lambda \to 0$, so $\hat{\Sigma}(\lambda) \to \mathbf{I}_m$, therefore the linearly approximated distribution converges to $\mathcal{N}(g'(0), \mathbf{I}_m)$ as claimed. Otherwise, $\hat{\Sigma}(\lambda)$ will converge to some projection matrix.

□

B Derivation of Covariance Matrix for Quadratic Kernel in Subsection 4.2

Suppose there are two features (z_1, z_2) in the data space, then the quadratic kernel, computed as $k(x, y) = (x^\top y)^2$, maps the data to feature space containing the following five features: $(z_1^2, z_2^2, z_1 z_2, z_1, z_2)$. Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ be a dataset, the kernel matrix $\mathbf{K}$ of $\mathbf{X}$ is computed as

$$\begin{aligned}\mathbf{K} &= \phi(\mathbf{X})^\top\phi(\mathbf{X}), \\ &= (\mathbf{X}^\top\mathbf{X})^2, \\ &= \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix}, \\ &= \begin{bmatrix} (x_1 \cdot x_1)^2 & \cdots & (x_1 \cdot x_n)^2 \\ \vdots & \ddots & \vdots \\ (x_n \cdot x_1)^2 & \cdots & (x_n \cdot x_n)^2 \end{bmatrix}.\end{aligned}$$

The kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ is symmetric, positive semi-definite.

Suppose $\phi : \mathbb{R}^m \to \mathcal{H}$ be a mapping from the input space to the quadratic kernel space. For a datapoint $y \in \mathbb{R}^m$, we denote $\phi(\mathbf{X})^\top \phi(y) = [k(x_1, y), \cdots, k(x_n, y)]^\top = [(x_1 \cdot y)^2, \cdots, (x_n \cdot y)^2]^\top$. Thus, the gradient of $\phi(\mathbf{X})^\top \phi(y)$ can be written as

$$\begin{aligned}
\nabla\phi(\mathbf{X})^\top \phi(y) &= \begin{bmatrix} \frac{\partial k(x_1,y)}{\partial y_1} & \cdots & \frac{\partial k(x_1,y)}{\partial y_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial k(x_n,y)}{\partial y_1} & \cdots & \frac{\partial k(x_n,y)}{\partial y_m} \end{bmatrix}, \\
&= \begin{bmatrix} \frac{\partial (x_1\cdot y)^2}{\partial y_1} & \cdots & \frac{\partial (x_1\cdot y)^2}{\partial y_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial (x_n\cdot y)^2}{\partial y_1} & \cdots & \frac{\partial (x_n\cdot y)^2}{\partial y_m} \end{bmatrix}, \\
&= \begin{bmatrix} 2(x_1 \cdot y)x_{1_1} & \cdots & 2(x_1 \cdot y)x_{1_m} \\ \vdots & \ddots & \vdots \\ 2(x_n \cdot y)x_{n_1} & \cdots & 2(x_n \cdot y)x_{n_m} \end{bmatrix}, \\
&= \underbrace{\sqrt{\phi(\mathbf{X})^\top \phi(y)}}_{\mathbb{R}^n} \circ \underbrace{2\mathbf{X}^\top}_{\mathbb{R}^{n\times m}},
\end{aligned}$$

where y_i represents the i-th feature of data y, and x_{i_j} means the j-th feature of data x_i. The $\circ$ operation is the Hadamard product, so in $\sqrt{\phi(\mathbf{X})^\top \phi(y)} \circ 2\mathbf{X}^\top$, the j-th row of $\mathbf{X}^\top$ is multiplied by the j-th entry in vector $\sqrt{\phi(\mathbf{X})^\top \phi(y)}$.

Suppose $y \in \mathbb{R}^m, y \notin \mathbf{X}$. By Equation (4), we have:

$$c(y) = (\mathbf{K} + \lambda\mathbf{I})^{-1}\phi(\mathbf{X})^\top \phi(y).$$

Taking gradient on both sides with respect to y gives:

$$\nabla\phi(\mathbf{X})^\top \phi(y) = (\mathbf{K} + \lambda\mathbf{I})\nabla c(y) \implies \nabla c(y) = (\mathbf{K} + \lambda\mathbf{I})^{-1}\nabla\phi(\mathbf{X})^\top \phi(y).$$

Thus, the local covariance matrix at y can be approximated in terms of kernel matrix $\mathbf{K}$ and original data matrix $\mathbf{X}$:

$$\begin{aligned}
&\Big(\phi(\mathbf{X})\nabla c(y)\Big)^\top \Big(\phi(\mathbf{X})\nabla c(y)\Big) \\
=&\Big(\nabla c(y)\Big)^\top \Big(\phi(\mathbf{X})\Big)^\top \phi(\mathbf{X})\nabla c(y), \\
=&\Big(\nabla c(y)\Big)^\top \mathbf{K}\Big(\nabla c(y)\Big), \\
=&\Big((\mathbf{K} + \lambda\mathbf{I})^{-1}\nabla\phi(\mathbf{X})^\top \phi(y)\Big)^\top \mathbf{K}\Big((\mathbf{K} + \lambda\mathbf{I})^{-1}\nabla\phi(\mathbf{X})\phi(y)\Big), \\
=&\Big(\nabla\phi(\mathbf{X})\phi(y)\Big)^\top (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}\Big(\nabla\phi(\mathbf{X})\phi(y)\Big), \\
=&\left(\sqrt{\phi(\mathbf{X})^\top \phi(y)} \circ 2\,\mathbf{X}^\top\right)^\top (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}\left(\sqrt{\phi(\mathbf{X})^\top \phi(y)} \circ 2\,\mathbf{X}^\top\right).
\end{aligned}$$

Let $\mathbf{X}_{-i} \in \mathbb{R}^{m\times(n-1)}$ be the dataset excluding data point x_i. Denote $\phi(\mathbf{X}_{-i}) = \{\phi(x_1), \cdots, \phi(x_{i-1}), \phi(x_{i+1}), \cdots, \phi(x_n)\}$, and $\mathbf{K}_{-i} = \phi(\mathbf{X}_{-i})^\top \phi(\mathbf{X}_{-i})$ The objective function in Equation (2) is equivalent to

$$\min_c \frac{1}{2}\|\phi(x_i) - \phi(\mathbf{X}_{-i})c(x_i)\|_2^2 + \frac{\lambda}{2}\|c(x_i)\|_2^2.$$

Following the above derivation, the local covariance of optimal self-expressive mapping f at x_i can be similarly approximated in terms of kernel matrix $\mathbf{K}_{-i}$ and original data $\mathbf{X}$:

$$
\begin{aligned}
&\left(\phi(\mathbf{X}_{-i})\nabla c(x_i)\right)^\top \left(\phi(\mathbf{X}_{-i})\nabla c(x_i)\right) \\
&= \left(\sqrt{\phi(\mathbf{X}_{-i})^\top \phi(x_i)} \circ 2\, \mathbf{X}_{-i}^\top\right)^\top (\mathbf{K}_{-i} + \lambda \mathbf{I})^{-1} \mathbf{K}_{-i} (\mathbf{K}_{-i} + \lambda \mathbf{I})^{-1} \left(\sqrt{\phi(\mathbf{X}_{-i})^\top \phi(x_i)} \circ 2\, \mathbf{X}_{-i}^\top\right).
\end{aligned}
$$

C Derivation of Covariance Matrix for Quadratic Kernel in Subsection 4.3

To extend the variance control mechanism from latent to data space, we can introduce a nonlinear kernel mapping $\phi : \mathbb{R}^m \to \mathcal{H}$ that maps input data $x \in \mathbb{R}^m$ to a high-dimensional feature space. The reconstruction mapping $f : \mathcal{H} \to \mathcal{H}$ defined in Equation (4) reconstructs the $\phi(x)$ in latent space from all other data point $\phi(\mathbf{X})$. We then learn a linear regression mapping $g : \mathcal{H} \to \mathbb{R}^m$ maps the reconstructed $f(\phi(x))$ back to the data space x itself, satisfying:

$$
\mathbf{X}^\top = \phi(\mathbf{X})^\top \alpha^*
$$

where the optimal coefficients α^* of g are obtained from the normal equation:

$$
\begin{aligned}
\alpha^* &= (\phi(\mathbf{X})\phi(\mathbf{X})^T + \lambda \mathbf{I})^{-1} \phi(\mathbf{X})^T \mathbf{X}^T \\
&= \phi(\mathbf{X})(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{X}^T,
\end{aligned}
$$

is the optimal coefficients of mapping g calculated from the normal equation.

The composed mapping $h = g \circ f : \mathbb{R}^m \to \mathbb{R}^m$ then takes an explicit form:

$$
\begin{aligned}
h(x) &= \left((\phi(\mathbf{X})c(x))^\top \phi(\mathbf{X})(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top\right)^\top, \\
&= \left(c(x)^\top \mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top\right)^\top, \\
&= \mathbf{X}(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1} (\phi(\mathbf{X})^\top \phi(x)).
\end{aligned}
$$

Substitute $(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K} (\mathbf{K} + \lambda \mathbf{I})^{-1}$ as $\hat{\mathbf{K}}$. Following Lemma 1 in Subsection 4.1, the local covariance at $h(x)$ after mapping can be approximated as

$$
\begin{aligned}
&\nabla h(x)^\top \nabla h(x) \\
&= \left(\mathbf{X}\hat{\mathbf{K}}(\nabla \phi(\mathbf{X})^\top \phi(x))\right)^\top \left(\mathbf{X}\hat{\mathbf{K}}(\nabla \phi(\mathbf{X})^\top \phi(x))\right), \\
&= \left(\nabla \phi(\mathbf{X})^\top \phi(x)\right)^\top \hat{\mathbf{K}} \mathbf{X}^\top \mathbf{X} \hat{\mathbf{K}} \left(\nabla \phi(\mathbf{X})^\top \phi(x)\right).
\end{aligned}
$$

D Computation Resources for Experiments

We used a Macbook Pro Laptop M3 Pro CPU with 18GB RAM to run all the experiments presented in Section 5.