Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article

JMM

Softmax self representation learning for unsupervised feature selection

Hossein Nasserassadi[†], Faranges Kyanfar[†*], Farid Saberi-Movahed[§], Abbas Salemi[†,‡]

[†]*Department of Applied Mathematics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran*
[‡]*Mahani Math Center, Afzalipour Research Institute, Shahid Bahonar University of Kerman, Kerman, Iran*
[§]*Department of Computer Science, Faculty of Sciences and Modern Technologies, Graduate University of Advanced Technology, Kerman, Iran*

Email(s): hnaserasadi@math.uk.ac.ir, kyanfar@uk.ac.ir, f.saberimovahed@kgut.ac.ir, salemi@uk.ac.ir

Abstract. Self-Representation (SR) models play a fundamental role in numerous unsupervised learning tasks, particularly in feature selection and clustering, by capturing intrinsic relational structures. However, learning reliable weight matrices in SR models remains challenging, as conventional nonnegativity constraints are often insufficient to control coefficient magnitudes. This limitation can lead to dense, unstable, and weakly interpretable solutions. To address these issues, we propose a softmax-based reparameterization for both sample and feature SR weight matrices. This probabilistic normalization enforces nonnegativity and unit-sum constraints, suppresses coefficient explosion, and induces competitive and interpretable affinity structures. Moreover, the proposed reparameterization transforms the original constrained optimization into an unconstrained problem, enabling efficient and stable gradient-based optimization. Building on this framework, we develop three SR variants, termed Softmax SR, Softmax Mixture SR, and Softmax Bilinear SR, each equipped with an efficient iterative optimization scheme. Extensive experiments on four benchmark datasets demonstrate that the proposed methods consistently outperform state-of-the-art approaches, highlighting the effectiveness of our approach in addressing key challenges in SR model optimization and its practical potential in real-world applications. Moreover, they achieve consistent performance gains, delivering overall improvements of around 2–7 percentage points in clustering accuracy (ACC), normalized mutual information (NMI), and Purity compared to competing non-Softmax SR and related methods across four benchmark datasets. The code is also available at https://github.com/FaridSabM/SSR.

Keywords: Unsupervised feature selection, self-representation, softmax, mixture SR, bilinear SR.
AMS Subject Classification 2010: 34A34, 65L05.

*Corresponding author
Received: 22 December 2025/ Revised: 21 May 2026/ Accepted: 21 May 2026
DOI: 10.22124/jmm.2026.32592.2964

© 2026 University of Guilan

Corrected Proof

1 Introduction

High-dimensional data arise naturally across many scientific and engineering domains, including computer vision, biomedical analysis, remote sensing, and intelligent control systems [14]. Although such data provide rich descriptive information, their high dimensionality often introduces redundancy, noise, and task-irrelevant variations that obscure the underlying latent structure. These challenges are particularly significant when label information is unavailable, requiring unsupervised methods to rely solely on intrinsic data relationships [21]. Learning compact and discriminative representations is therefore essential for improving performance and interpretability in high-dimensional systems. In supervised settings, label information guides models to learn representations that enhance class separability and task relevance. However, acquiring large labeled datasets is often costly and impractical [28]. Unsupervised approaches overcome this limitation by discovering patterns based solely on data structure, though they may fail to capture task-specific discriminative information [3]. Semi-supervised learning bridges these paradigms by combining limited labeled data with abundant unlabeled samples, enabling models to exploit both supervision and intrinsic data structure to produce more robust and effective representations [22].

In recent years, research in unsupervised feature selection has increasingly gravitated toward subspace learning formulations–particularly those grounded in *manifold preservation* and *Self-Representation (SR) learning* [15]. These perspectives aim to identify informative low-dimensional structures embedded within high-dimensional observations, enabling the selection of features that reflect meaningful geometric or statistical patterns without relying on supervision. The following developments in both manifold-based and SR-based modeling illustrate the evolution of these ideas and their relevance to our study.

To explicitly model geometric structure, manifold-based approaches incorporate local relationships through graph regularization to preserve neighborhood consistency. Subspace learning-based Graph regularized Feature Selection (SGFS) [24] enhances earlier factorization models by embedding feature-manifold information into a graph-regularized framework, enabling the retention of local geometric relationships that purely factorization-based methods often overlook. Sparse and Low-redundant Subspace learning-based Dual-graph regularized Robust feature selection (SLSDR) [25] advances this principle through a dual-graph regularization strategy that simultaneously accounts for feature-manifold and data-manifold structures, yielding a more comprehensive geometric preservation at the cost of increased model complexity. Unsupervised Feature Selection for High-order embedding learning and Sparse learning (UFSHS) [8] further improves geometric modeling by constructing an optimal similarity graph derived from high-order structural dependencies and integrating embedding learning with sparsity to achieve an efficient row-sparse projection matrix. More recently, Unsupervised feature selection algorithm based on Dual-graph clustering Learning and Adaptive weighting (UDLA) [32] has demonstrated that manifold information extracted from multiple spaces can significantly enhance unsupervised feature selection when combined with adaptive weighting and latent subspace learning, though this comes with additional components that must be carefully balanced. Beyond these generic feature-learning tasks, representation learning methodologies have also been applied to specialized domains–for example, personalized neural prosthesis control systems leverage deep models to learn discriminative neural activity patterns [10], underscoring the importance of extracting effective representations from complex, high-dimensional signals.

Parallel to manifold-oriented approaches, *SR learning* has emerged as a powerful mechanism for revealing intrinsic relational structure in high-dimensional data [31]. The SR models reconstruct each data entity as a combination of others, allowing representation coefficients to encode latent similarities

Corrected Proof

that facilitate both subspace discovery and feature selection. Early efforts such as self-representation based Dual-graph regularized Feature Selection Clustering (DFSC) [23] constructed dual SR models to separately capture feature–feature and sample–sample dependencies while preserving locality, though the independent treatment of the two spaces limited their mutual reinforcement. Subspace Clustering unsupervised Feature Selection (SCFS) [20] subsequently coupled subspace learning with adaptive similarity reconstruction, enabling the recovered similarities to reflect latent cluster structures, but at the expense of sensitivity to noise in similarity estimation. Bilinear Self-representation for Unsupervised Feature Selection with Structure Learning (BSUFSL) [1] introduced a bilinear SR formulation to model feature and sample relationships simultaneously, yet the interaction between the two remained implicit, limiting the consistency of the learned structure. More recent work such as Feature-Weighted Hypergraph and Clustering Similarity Self-Representation (FWHSR) [9] improved interpretability by deriving SR coefficients from structural similarities in a learned target space, whereas dual low-rank constrained SR frameworks [12] integrated global low-rank priors, projection-distance penalties, and local manifold constraints to enhance robustness in challenging tasks like SAR image change detection. Together, these developments demonstrate the value of geometric modeling, relational structure learning, and structural regularization in unsupervised feature selection.

Through the SR mechanism, each sample (or feature) is expressed as a linear combination of the remaining samples (or features), enabling the discovery of latent dependencies and underlying subspaces [27]. Classical SR methods can be broadly classified into three groups: sample-level SR, in which each sample is reconstructed from other samples, feature-level SR, where each feature is reconstructed from other features, and mixture-level SR, which attempts to exploit both sample and feature dependencies simultaneously [29]. Despite their effectiveness, these approaches exhibit several limitations. Sample-level and feature-level SR consider only a single structural view of the data, thereby ignoring potentially informative relationships in the complementary domain [16]. Mixture-level SR methods, although formulated to utilize both perspectives, typically learn the sample and feature representation matrices independently [26]. As a result, important interactions between samples and features remain unexplored, and removing one component of the mixture-level objective does not affect the learning of the other [5]. Bilinear self-representation was recently introduced as a principled approach to address this deficiency by coupling sample-level and feature-level SR through a bilinear form of the data matrix structure [1]. This model enables a unified and coherent representation of the data, where both sample and feature relationships jointly contribute to the reconstruction process. However, existing SR frameworks still suffer from several fundamental limitations. First, the nonnegativity constraints imposed on the weight matrices are not sufficient to regulate the magnitude of the coefficients. As a result, the learned representations often become dense, numerically unstable, and difficult to interpret. Second, classical multiplicative-update optimization schemes lack normalization mechanisms and therefore cannot enforce any probabilistic structure among the representation coefficients.

To address these limitations, we propose a softmax-based reparameterization of the weight matrices in self-representation models. Instead of directly optimizing the weights under nonnegativity constraints, we represent them through a softmax mapping. This transformation converts each row (or column) of the weight matrix into a probability distribution, ensuring nonnegativity, normalization, and controlled coefficient magnitudes. As a result, the learned representations become more stable, sparse in practice, and significantly more interpretable. Softmax turns each row/column of weight matrices into an attention-like probability distribution, where only the most relevant samples/features are emphasized [6]. This probabilistic formulation offers a new perspective on SR, bridging classical matrix factorization models

Corrected Proof

with modern normalized attention-based mechanisms. An additional advantage of the proposed formulation is that the softmax reparameterization transforms the original constrained optimization problem into an unconstrained one with respect to the underlying variables. This enables the use of efficient gradient-based optimization methods, leading to improved numerical stability and convergence behavior. Furthermore, unlike previous SR studies that treat different SR variants independently, the proposed formulation provides a unified softmax-based framework that can be applied to feature-level SR, mixture SR, and bilinear SR models. This unified perspective allows the probabilistic normalization mechanism to enhance multiple SR architectures in a consistent manner. The main contributions of this work can be summarized as follows:

- We introduce a softmax-based probabilistic reparameterization for the weight matrices in self-representation models. Unlike classical SR approaches that only impose nonnegativity constraints, the proposed formulation produces normalized probabilistic coefficients, improving stability and interpretability.
- We show that the softmax reparameterization transforms the original constrained SR optimization problem into an unconstrained formulation, which can be efficiently solved using gradient-based optimization techniques.
- We develop a unified softmax-based framework that can be applied to multiple SR architectures, including feature-level SR, mixture SR, and bilinear SR models.
- Extensive experiments on several benchmark datasets demonstrate that the proposed framework improves clustering accuracy and feature selection performance compared with existing SR-based methods.

The rest of this paper is organized as follows. Section 2 introduces the preliminaries of sample-level, feature-level, and bilinear self-representation, along with the motivation for adopting a softmax-based SR formulation. Section 3 describes the proposed softmax-based self-representation framework in detail. The optimization strategy, including gradient analysis of the softmax function, is presented in Section 4. Experimental results and performance analysis are reported in Section 5, and the paper concludes with final remarks in Section 6.

2 Background and motivation

This section provides the theoretical foundations and conceptual motivations underlying the proposed method. We first introduce the notations used throughout the paper for clarity and consistency. Then, we review classical SR models employed in unsupervised feature selection, including sample-level, feature-level, mixture-level, and bilinear formulations, highlighting their structural properties and limitations. Finally, we discuss the motivation for introducing a softmax-based reparameterization, which addresses key challenges in conventional SR models related to stability, and interpretability.

2.1 Notations

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ denote the data matrix, where each row corresponds to a sample and each column corresponds to a feature. Specifically, the i-th sample is denoted by $\mathbf{x}_i \in \mathbb{R}^{1 \times n}$, while the j-th feature is

Corrected Proof

represented as $\mathbf{f}_j \in \mathbb{R}^{m\times 1}$. For any matrix $\mathbf{Q}$, we use $\mathbf{Q}_{i,:}$ and $\mathbf{Q}_{:,j}$ to indicate its i-th row and j-th column, respectively, and $\mathbf{q}_{ij}$ denotes the entry located at the i-th row and j-th column. The Frobenius norm and the $\ell_{2,1}$-norm of $\mathbf{Q}$ are written as $\|\mathbf{Q}\|_F$ and $\|\mathbf{Q}\|_{2,1}$. In addition, $\mathbf{Q}^\top$ represents the transpose of $\mathbf{Q}$, and $\mathrm{Tr}(\mathbf{Q})$ denotes its trace. Unless otherwise specified, $\mathbb{R}_+$ refers to the set of non-negative real values.

2.2 Background: self-representation and bilinear models

In this subsection, we present a systematic overview of SR models employed for feature selection. The discussion is organized according to the principal SR paradigms, with each level described in terms of its underlying formulation. For completeness, the corresponding final objective function associated with each SR level is also provided

Levels of self-representation. Existing SR approaches differ in the domain where linear reconstruction is applied:

- **Sample-level SR:** each sample is reconstructed as a combination of other samples. The goal is to learn $\mathbf{A} \in \mathbb{R}^{m\times m}$ such that

$$\mathbf{X} \approx \mathbf{A}\mathbf{X}. \tag{1}$$

  A typical formulation solves

$$\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{A}\mathbf{X}\|_F^2 + \mathrm{Reg}(\mathbf{A}), \tag{2}$$

  where nonnegativity enhances interpretability (coefficients act as additive contributions) and $\mathrm{Reg}(\mathbf{A})$ is the regularization term applied to the weight matrix $\mathbf{A}$ [4].

- **Feature-level SR:** each feature is expressed using other features. The objective is to learn

$$\mathbf{X} \approx \mathbf{X}\mathbf{B}, \tag{3}$$

  where $\mathbf{B} \in \mathbb{R}^{n\times n}$. A standard model solves

$$\min_{\mathbf{B}} \|\mathbf{X} - \mathbf{X}\mathbf{B}\|_F^2 + \mathrm{Reg}(\mathbf{B}), \tag{4}$$

  where $\mathrm{Reg}(\mathbf{B})$ is the regularization term applied to the weight matrix $\mathbf{B}$ [37].

- **Mixture-level SR:** both sample and feature relations are learned jointly by optimizing

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{A}\mathbf{X}\|_F^2 + \|\mathbf{X} - \mathbf{X}\mathbf{B}\|_F^2 + \mathrm{Reg}(\mathbf{A}) + \mathrm{Reg}(\mathbf{B}), \tag{5}$$

  where $\mathbf{A}$ models sample dependencies and $\mathbf{B}$ captures feature relationships. In mixture-level models, the matrices $\mathbf{A}$ and $\mathbf{B}$ are optimized independently, meaning feature structure cannot directly influence sample structure and vice versa. As a result, structural information discovered in one domain does not reliably propagate to the other. Furthermore, constraints such as nonnegativity or sparsity alone do not control the scale of rows or columns, which may lead to dense or unstable coefficient matrices and reduced interpretability [23].

Corrected Proof

- **Bilinear SR:** Bilinear SR models strengthen the coupling between sample-level and feature-level structure by reconstructing the data through both matrices simultaneously:

$$\mathbf{X} \approx \mathbf{AXB}. \tag{6}$$

The mapping $\mathbf{X} \mapsto \mathbf{AXB}$ is bilinear, preserving linearity in one variable when the other is fixed. This property enables more effective cross-domain information flow: updates to $\mathbf{A}$ influence the feature-level reconstruction, and vice versa. Bilinear SR can incorporate sparsity, smoothness, or group-structured regularizers, and its two-sided structure provides a richer representation than either sample-only or feature-only models [1]. A practical formulation augments (6) with constraints and regularization:

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{AXB}\|_F^2 + \mathrm{Reg}(\mathbf{A}) + \mathrm{Reg}(\mathbf{B}), \tag{7}$$

where $\mathrm{Reg}(\cdot)$ encourages sparsity or structural coherence.

To clarify the scope of our literature review, we focus on the three core families of self-representation (SR) models for feature selection: Feature-level SR, Mixture-level SR, and Bilinear SR. These families constitute the fundamental frameworks developed in this field from 2015 to 2025. Many subsequent methods are extensions of these core models, typically incorporating additional regularization or sparsity constraints. By emphasizing these principal SR paradigms, we provide a structured overview while highlighting the foundations upon which recent works are built. In this work, we introduce differentiable softmax-based reparameterizations that enforce normalization, enhance interpretability, and enable efficient gradient-based optimization. These ideas form the basis of the unified SR framework developed in the next section.

2.3 Motivation and softmax-based reparameterization

The SR models lie at the core of numerous unsupervised learning techniques, particularly in clustering and feature selection [35]. These models express each sample or feature as a combination of others, capturing the intrinsic relational structure of high-dimensional data. Classical SR methods enforce nonnegativity on the weight matrices to maintain interpretability and ensure additive reconstruction. However, despite this restriction, several key challenges persist [17].

First, nonnegativity alone does not regulate the scale or distribution of SR coefficients. It only enforces sign constraints without imposing any upper bound or normalization. This results in multiple scaling-equivalent solutions, where coefficients may grow arbitrarily large or become overly dense, while yielding similar reconstruction errors. Consequently, the learned representations become highly sensitive to noise and often lack discriminative power.

Second, the optimization procedures commonly used in SR models, such as multiplicative updates or alternating minimization, do not incorporate implicit normalization mechanisms. Without row-wise or column-wise normalization, coefficient updates become uncoupled. This leads to scale drift, poorly conditioned solutions, and the absence of competitive interactions. As a result, these methods fail to produce selective and interpretable representations, as increasing one coefficient does not suppress others.

Building on these limitations, we now discuss how softmax-based reparameterization naturally resolves them. Instead of directly constraining weight matrices $\mathbf{A}$ and $\mathbf{B}$ to be nonnegative, we learn

Corrected Proof

unconstrained matrices $\mathbf{Y}$ and $\mathbf{Z}$ and define

$$\mathbf{A} = \text{softmax}(\mathbf{Y}), \qquad \mathbf{B} = \text{softmax}(\mathbf{Z}).$$

The softmax function is applied row-wise or column-wise, depending on the SR setting. This ensures that each column of $\mathbf{A}$ or each row of $\mathbf{B}$ forms a valid probability distribution. The resulting coefficients satisfy

$$\mathbf{a}_{ij} \geq 0, \qquad \mathbf{b}_{ij} \geq 0, \qquad \sum_j \mathbf{b}_{ij} = 1 \text{ or } \sum_i \mathbf{a}_{ij} = 1,$$

which imposes a normalized structure with inherent competition. Weak relationships are exponentially suppressed, while dominant dependencies naturally emerge, yielding more discriminative and interpretable representations. Because softmax normalizes each row or column through a shared denominator, increasing one coefficient necessarily suppresses the others. This creates a competitive structure that classical SR does not provide [30].

Softmax also prevents scale inflation. In classical SR, a sample may be assigned very large weights because no mechanism exists to prevent this. In contrast, softmax forces all outgoing weights to remain within a normalized probability simplex [2].

The constraints imposed on the weight matrices, such as nonnegativity, row/column stochasticity, and sparsity, are not arbitrary. In real-world applications such as image representation, gene expression analysis, and document clustering, interpretable and probabilistic feature/sample dependencies are critical. Nonnegative, normalized coefficients ensure that each feature or sample is represented as a meaningful convex combination of other elements, which aligns with practical requirements for interpretability and stability in downstream tasks.

Applying the softmax function along appropriate dimensions naturally enforces constraints that are meaningful in practice: row-stochastic feature weights and column-stochastic sample weights correspond to probability distributions. These distributions quantify relative importance, enabling interpretable reconstructions consistent with real data structures observed in applications such as gene co-expression networks and document-topic modeling. For instance, in single-cell RNA sequencing datasets, each cell's expression profile can be represented as a convex combination of other cells. Feature interactions correspond to co-regulated gene modules. The probabilistic softmax constraints ensure that reconstructed profiles reflect biologically plausible relationships rather than arbitrary combinations.

A major advantage of this reparameterization is that the optimization problem becomes unconstrained with respect to $\mathbf{Y}$ and $\mathbf{Z}$ [7]. The softmax mapping is smooth and differentiable, enabling the direct use of gradient-based methods. The Jacobian of the softmax operation further ensures that the learned coefficients evolve within a stable and well-conditioned probabilistic simplex [18]. This provides improved numerical robustness, prevents coefficient explosion, and enhances convergence behavior.

Within the bilinear framework **AXB**, softmax reparameterization introduces an implicit but effective coupling between sample and feature representations [36]. Each domain simultaneously influences and reflects the structure learned from the other, allowing the model to capture richer interactions than classical SR approaches. As a result, the proposed softmax-based formulation establishes a principled, stable, and probabilistically interpretable alternative to traditional SR models, providing a unified foundation for robust learning in unsupervised tasks [13].

In summary, the softmax-based SR framework addresses the key shortcomings of classical SR by combining nonnegativity, normalization, probabilistic interpretability, and unconstrained optimization

Corrected Proof

into a cohesive formulation. This unified perspective enhances model stability, discriminative power, and cross-domain interaction, making it especially well-suited for clustering and unsupervised feature selection. Although softmax normalizes the coefficients, sparsity regularization remains crucial for eliminating irrelevant samples or features that would otherwise maintain small but nonzero probabilities.

3 Methodology

The SR frameworks seek to uncover the internal structure of data by expressing each sample or feature as a weighted combination of others. When weight matrices are unconstrained, the resulting solutions are often dense and difficult to interpret. To overcome this challenge, we introduce a family of softmax-based SR models in which each weight matrix is generated through a smooth, competitive normalization function. The use of softmax enforces nonnegativity, stochasticity, and implicit sparsity while maintaining fully differentiable optimization. This framework leads to three progressively coupled SR formulations: Softmax SR, Softmax Mixture SR, and Softmax Bilinear SR. To compare the different SR models obtained using softmax in this section and the models introduced in Section 2, for the sake of fairness, we only use the term sparsity regularization, and since the ultimate goal is feature selection, all models being compared use sparsity on the rows of $\mathbf{B}$ as a constraint.

3.1 Sparsity

A central theme in self-representation models is that not all samples or features contribute equally to the reconstruction of the data. Whether the representation is built at the sample level, feature level, or through a mixture or bilinear SR, the objective is always to identify a small set of informative components that can most faithfully describe the structure of the dataset. Sparsity provides a principled mechanism for achieving this behavior. Consider a generic self-representation mapping in which the data matrix $\mathbf{X}$ is approximated by an operator involving a learnable weight matrix (or matrices):

$$\mathbf{X} \approx \mathscr{T}(\mathbf{X};\Theta),$$

where Θ denotes one or several weight matrices (e.g., $\mathbf{A}$, $\mathbf{B}$). Each column or row of a weight matrix encodes how strongly a given sample or feature relies on others to explain its behavior. For example, when these rows contain many nonzero entries, the resulting reconstruction is dense and the learned relations become difficult to interpret. In contrast, sparse rows highlight only the most influential contributors, leading to a more concise and discriminative representation of sample or feature relationships. To encourage model components to select only a few meaningful dependencies, sparsity is commonly imposed through row-wise penalties. The mixed $\ell_{2,1}$-norm,

$$\|\Theta\|_{2,1} = \sum_i \|\Theta_{i,:}\|_2$$

shrinks entire rows toward zero simultaneously. A column or row that vanishes indicates that the corresponding sample (in sample-level SR) or feature (in feature-level SR) plays little or no role in reconstructing the data. This behavior is particularly desirable in high-dimensional settings, where most variables contribute negligibly to the intrinsic structure of the dataset. Regardless of the specific SR formulation, sparsity can be incorporated by augmenting the reconstruction objective with an $\ell_{2,1}$ penalty:

$$\min_{\Theta} \|\mathbf{X} - \mathscr{T}(\mathbf{X};\Theta)\|_F^2 + \lambda \|\Theta\|_{2,1}, \tag{8}$$

Corrected Proof

where $\lambda > 0$ controls the degree of sparsity. In sample-level models, Θ may refer to $\mathbf{A}$; in feature-level models, to $\mathbf{B}$; and in bilinear or mixture formulations, to one or both matrices. Although the operator $\mathscr{T}$ differs across SR frameworks, the effect of the regularizer is consistent: it suppresses uninformative connections and yields a more interpretable and structurally meaningful representation.

3.2 A unified softmax-based SR framework

While sparsity provides a common regularization principle across different SR formulations, it does not by itself determine the probabilistic structure of the representation coefficients. To address this aspect in a unified way, we next introduce a softmax-based framework that systematically imposes normalized stochastic constraints on the SR weight matrices.

The motivation for introducing softmax into self-representation is twofold. First, unlike conventional optimization approaches that mainly enforce feasibility through nonnegativity, sparsity, or alternating constrained updates, softmax provides a normalized probabilistic structure on the representation coefficients. This makes the learned weights more interpretable, since each row or column can be viewed as a distribution of relative importance over samples or features. Second, softmax enables a reparameterization of the constrained SR problem into an unconstrained differentiable optimization problem over latent variables, which can be efficiently solved using standard gradient-based methods. Therefore, the proposed approach offers both modeling advantages and optimization advantages compared with many existing SR formulations.

Considering the variety of SR models discussed earlier, the application of the softmax function to their weight matrices must follow the structural role of each matrix. Specifically, softmax should be applied along the dimension that encodes the contribution of individual components. For example, when a weight matrix operates at the feature level, each row corresponds to a feature and the coefficients within that row quantify how strongly that feature participates in reconstructing other features. Therefore, the softmax function should be applied row-wise. In contrast, when a weight matrix is used in a sample-level SR, each column represents the contribution of other samples to the representation of a given sample. Thus, in this case, the softmax function must be applied column-wise. Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ denote the data matrix. We introduce latent matrices $\mathbf{Y} \in \mathbb{R}^{m\times m}$ and $\mathbf{Z} \in \mathbb{R}^{n\times n}$, which are transformed by a softmax function:

$$\mathbf{A} = \mathrm{softmax}_{\mathrm{col}}(\mathbf{Y}), \qquad \mathbf{B} = \mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}).$$

The softmax operator is defined as:

$$\mathrm{softmax}(\mathbf{z}) = \frac{e^{\mathbf{z}_i}}{\sum_j e^{\mathbf{z}_j}}.$$

Under this representation, all softmax-based SR models can be written as:

$$\min_{\Theta} \|X - \mathscr{T}(X;\Theta)\|_F^2 + \lambda\|\Theta\|_{2,1}, \qquad \Theta = \{A,B\},$$

where $\Theta = \{\mathbf{A},\mathbf{B}\}$ denotes the softmax-induced weight matrices. The operator $\mathscr{T}(\mathbf{X};\Theta)$ specifies how the sample-level and feature-level representations interact during reconstruction. Different choices of $\mathscr{T}$ give rise to various softmax-based SR formulations, ranging from single-domain representations to fully coupled bilinear or mixture models.

Corrected Proof

3.3 Softmax SR

To make the above unified framework concrete, we first consider its simplest instantiation in the classical feature-level SR setting. This model serves as the baseline softmax-based variant and shows how the general formulation specializes to a standard self-representation structure. The classical SR model reconstructs each feature as a combination of other features. We integrate softmax to obtain a competitive, probabilistic dependency structure. The model is:

$$\min_{\mathbf{B}} \|\mathbf{X}-\mathbf{X}\mathbf{B}\|_F^2+\lambda\|\mathbf{B}\|_{2,1}, \qquad \mathbf{B}=\mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}).$$

Each row of $\mathbf{B}$ forms a probability distribution describing how strongly feature i depends on other features. Because softmax enforces competition, only a few features tend to dominate the representation, yielding interpretable sparse relations without explicit hard constraints. The resulting weight matrix $\mathbf{B}$ acts as a probabilistic feature-to-feature transition operator, so the SR reconstruction is performed through the linear mapping

$$\mathbf{X} \mapsto \mathbf{X}\mathbf{B}.$$

Remark 1 (Row-wise softmax induces valid feature distributions)**.** *Let $\mathbf{Z}\in\mathbb{R}^{n\times n}$ be an arbitrary real-valued matrix and define*

$$\mathbf{B}=\mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}), \qquad \mathbf{b}_{ij}=\frac{e^{z_{ij}}}{\sum_{k=1}^{n} e^{z_{ik}}}.$$

By construction, all entries of $\mathbf{B}$ are strictly positive and each row of $\mathbf{B}$ sums to one, that is,

$$\sum_{j=1}^{n}\mathbf{b}_{ij}=1, \qquad i=1,\ldots,n.$$

As a result, each row of $\mathbf{B}$ can be interpreted as a probability distribution over the target features. Consequently, in the feature-level self-representation

$$\mathbf{f}_j=\mathbf{X}\mathbf{B}_{:,j}=\sum_{i=1}^{n}\mathbf{b}_{ij}\mathbf{f}_i,$$

the reconstruction coefficients are nonnegative and normalized, ensuring that each reconstructed feature $\mathbf{f}_j$ is expressed as a convex combination of the original features. This property provides a probabilistic and interpretable structure for the learned self-representation.

3.4 Softmax Mixture SR

While Softmax SR assigns a probabilistic structure only at the feature level, many datasets contain meaningful relationships in both the feature and sample domains. To capture these complementary structures simultaneously, we employ two independent softmax-normalized operators:

$$\mathbf{A}=\mathrm{softmax}_{\mathrm{col}}(\mathbf{Y}), \qquad \mathbf{B}=\mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}).$$

Corrected Proof

Here, $\mathbf{A}$ governs how each sample is influenced by other samples, whereas $\mathbf{B}$ describes how individual features participate in reconstructing one another. The Softmax Mixture SR model is formulated as

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X}-\mathbf{X}\mathbf{B}\|_F^2+\alpha\|\mathbf{X}-\mathbf{A}\mathbf{X}\|_F^2+\beta\|\mathbf{B}\|_{2,1} \quad \text{s.t.} \quad \mathbf{A}=\mathrm{softmax}_{\mathrm{col}}(\mathbf{Y}),\ \mathbf{B}=\mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}). \tag{9}$$

The two reconstruction pathways operate in parallel: the term $\|\mathbf{X}-\mathbf{X}\mathbf{B}\|_F^2$ captures explanatory power at the feature level, while $\|\mathbf{X}-\mathbf{A}\mathbf{X}\|_F^2$ reflects how samples relate within the data manifold. Because $\mathbf{A}$ is column-stochastic, each sample expresses itself through a probability-weighted aggregation of other samples; conversely, the row-stochastic structure of $\mathbf{B}$ ensures that each feature distributes its influence over alternative features. Crucially, the mixture model retains a clear separation between these two domains. The sample-level and feature-level dependencies are estimated independently yet optimized jointly, allowing each domain to contribute its own structural information without interfering with the interpretability of the other. As a result, the model captures both global geometric patterns (sample domain) and fine-grained functional relations (feature domain) in a probabilistically coherent manner.

Remark 2 (Column-wise softmax induces valid sample distributions)**.** *Let $\mathbf{Y}\in\mathbb{R}^{m\times m}$ be an arbitrary real-valued matrix and define*

$$\mathbf{A}=\mathrm{softmax}_{\mathrm{col}}(\mathbf{Y}), \qquad \mathbf{a}_{ij}=\frac{e^{y_{ij}}}{\sum_{k=1}^{m}e^{y_{kj}}}.$$

By construction, all entries of $\mathbf{A}$ are strictly positive and each column of $\mathbf{A}$ sums to one, namely,

$$\sum_{i=1}^{m}\mathbf{a}_{ij}=1, \qquad j=1,\ldots,m.$$

Therefore, each column $\mathbf{A}_{:,j}$ can be interpreted as a probability distribution that quantifies how sample j aggregates information from other samples. Consequently, in the sample-level self-representation

$$\mathbf{x}_j=\mathbf{A}\mathbf{x}_j=\sum_{i=1}^{m}\mathbf{a}_{ij}\mathbf{x}_i,$$

the reconstruction coefficients are nonnegative and normalized, ensuring that each reconstructed sample $\mathbf{x}_j$ is expressed as a convex combination of the original samples. This probabilistic structure enhances the interpretability and numerical stability of the sample-level self-representation.

3.5 Softmax Bilinear SR

The Softmax Bilinear SR model integrates sample-level and feature-level dependencies into a single operator, creating the strongest possible coupling between the two domains. Instead of treating sample and feature reconstructions independently, the model propagates information through both directions simultaneously:

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X}-\mathbf{A}\mathbf{X}\mathbf{B}\|_F^2+\lambda\|\mathbf{B}\|_{2,1}, \qquad \mathbf{A}=\mathrm{softmax}_{\mathrm{col}}(\mathbf{Y}),\ \mathbf{B}=\mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}).$$

The constraints impose complementary probabilistic structures:

Corrected Proof

- **A** is column-stochastic, describing how each sample pools information from other samples.
- **B** is row-stochastic, encoding how features redistribute their influence across other features.

When combined as $\mathbf{X} \mapsto \mathbf{AXB}$, these operators form a two-sided propagation mechanism. Left multiplication with **A** diffuses each sample across the data manifold, while right multiplication with **B** redistributes feature contributions through a softmax-induced probability structure. As a result, the transformation simultaneously smooths inter-sample relationships and reconfigures feature interactions in a unified representation framework.

Figure 1 provides an intuitive comparison of the three proposed formulations. Softmax-SR models only feature-level dependencies through a normalized transition matrix **B**. Softmax Mixture-SR separates the sample-level and feature-level pathways and optimizes them jointly. Softmax Bilinear-SR further couples these two pathways into a single bilinear operator, thereby capturing their interaction in a unified representation. This schematic view is intended to make the methodology more accessible to readers without relying solely on the mathematical derivations.

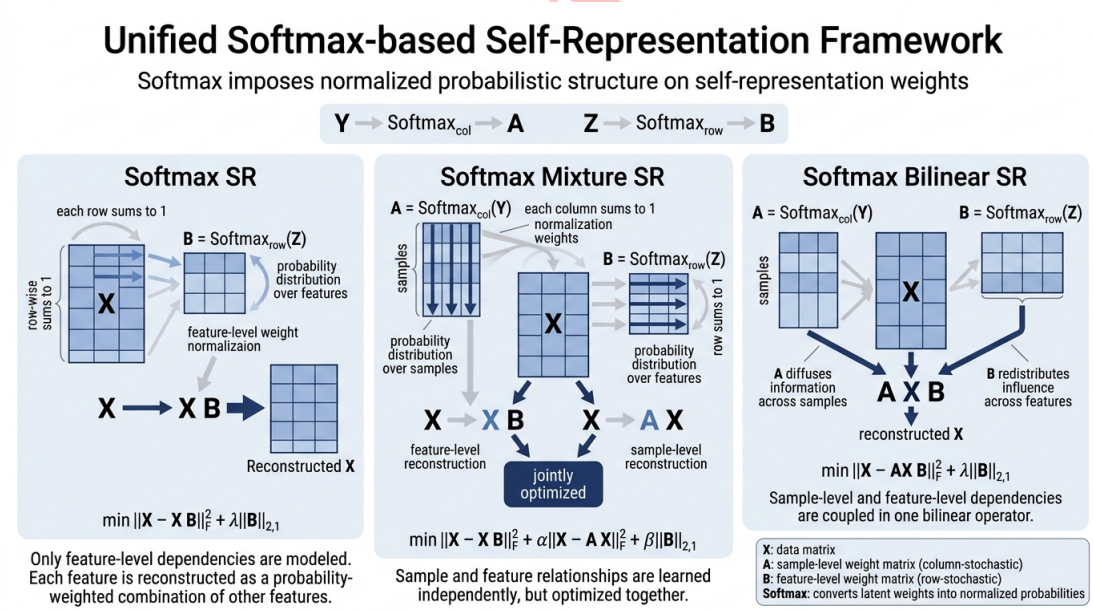

Figure 1: Schematic illustration of the three softmax-based self-representation models

4 Optimization

Having established the probabilistic structure of the proposed softmax-based representations, we now turn to the question of how these constrained models can be optimized in practice. This section presents the optimization procedures developed for the proposed softmax-based SR models and explains how the mathematical formulations introduced in the previous section are translated into executable update rules.

In particular, we describe a two-stage optimization procedure for the proposed softmax-based SR models, providing an intuitive overview before the mathematical derivations. The optimization alternates

Proof

between two stages:

1. **Multiplicative update:** This stage efficiently updates the main matrices while preserving non-negativity, producing an initial estimate of the parameters.

2. **Gradient-based softmax projection:** This stage refines the updates by projecting the parameters onto the softmax simplex, ensuring probabilistic consistency.

These two stages are iterated until convergence, balancing computational efficiency and solution accuracy. We then present a unified optimization framework for all softmax-based SR models introduced in this work. We begin by analyzing the optimization of the softmax reparameterization, which plays a key role in enforcing stochasticity constraints on weight matrices. After establishing the gradient flow through the softmax function, we derive the update rules for Softmax-SR, Softmax-Mixture SR, and Softmax-Bilinear SR, and then summarize their optimization algorithms.

4.1 Optimization of the softmax reparameterization

Since all proposed models rely on softmax-based reparameterization to enforce stochastic constraints, the first step is to establish the corresponding gradient expressions. These derivatives provide the common analytical foundation required to derive the update rules of all subsequent softmax-based SR variants. We therefore begin with the Jacobian of the row-wise softmax mapping, which will be repeatedly used in the optimization procedures developed in the following subsections.

Proposition 1 (Gradient of the Row-wise softmax)**.** *Let $\mathbf{Z} \in \mathbb{R}^{m\times n}$ be a real-valued matrix and define the row-wise softmax mapping $\mathbf{W} \in \mathbb{R}^{m\times n}$ by*

$$\mathbf{W}_{i,:} = \mathrm{softmax}(\mathbf{Z}_{i,:}), \qquad \mathbf{W}_{ij} = \frac{e^{(\mathbf{Z}_{ij})}}{\sum_{t=1}^{n} e^{(\mathbf{Z}_{it})}}.$$

Then each row of $\mathbf{W}$ lies on the probability simplex, i.e., $\mathbf{W}_{ij} \geq 0$ and $\sum_{j=1}^{n} \mathbf{W}_{ij} = 1$. Moreover, the Jacobian matrix of $\mathbf{W}_{i,:}$ with respect to $\mathbf{Z}_{i,:}$ is given by

$$\frac{\partial \mathbf{W}_{i,:}}{\partial \mathbf{Z}_{i,:}} = \mathrm{diag}(\mathbf{W}_{i,:}) - \mathbf{W}_{i,:}^{\top}\mathbf{W}_{i,:}. \tag{10}$$

Proof. From the softmax definition, for a fixed row i we have

$$\mathbf{W}_{ip} = \frac{e^{\mathbf{Z}_{ip}}}{\sum_{t=1}^{n} e^{\mathbf{Z}_{it}}} = \frac{e^{\mathbf{Z}_{ip}}}{\mathbf{S}_i},$$

where the normalization term is defined as

$$\mathbf{S}_i = \sum_{t=1}^{n} e^{\mathbf{Z}_{it}}.$$

We aim to compute the partial derivative of $\mathbf{W}_{ip}$ with respect to $\mathbf{Z}_{iq}$ for all $p, q \in \{1, \ldots, n\}$.

Corrected Proof

- The derivative of the numerator term $e^{\mathbf{Z}_{ip}}$ is:

$$\frac{\partial e^{\mathbf{Z}_{ip}}}{\partial \mathbf{Z}_{iq}} = \begin{cases} e^{\mathbf{Z}_{ip}}, & \text{if } p = q, \\ 0, & \text{if } p \neq q. \end{cases}$$

- The derivative of the denominator $\mathbf{S}_i = \sum_{t=1}^{n} e^{\mathbf{Z}_{it}}$ with respect to $\mathbf{Z}_{iq}$ is:

$$\frac{\partial \mathbf{S}_i}{\partial \mathbf{Z}_{iq}} = e^{\mathbf{Z}_{iq}}.$$

Simplifying by factoring $e^{\mathbf{Z}_{ip}}$ from the numerator:

$$\frac{\partial \mathbf{W}_{ip}}{\partial \mathbf{Z}_{iq}} = e^{\mathbf{Z}_{ip}} \frac{(\delta_{pq}\mathbf{S}_i - e^{\mathbf{Z}_{iq}})}{\mathbf{S}_i^2}.$$

Now, using the definition $\mathbf{W}_{ip} = \frac{e^{\mathbf{Z}_{ip}}}{\mathbf{S}_i}$, we can rewrite the result as:

$$\frac{\partial \mathbf{W}_{ip}}{\partial \mathbf{Z}_{iq}} = \mathbf{W}_{ip}(\delta_{pq} - \mathbf{W}_{iq}).$$

Hence, we obtain the compact and well-known softmax derivative formula:

$$\boxed{\frac{\partial \mathbf{W}_{ip}}{\partial \mathbf{Z}_{iq}} = \mathbf{W}_{ip}(\delta_{pq} - \mathbf{W}_{iq}),} \tag{11}$$

which shows that the Jacobian of the softmax function for row i is given by

$$\frac{\partial \mathbf{W}_{i,:}}{\partial \mathbf{Z}_{i,:}} = \operatorname{diag}(\mathbf{W}_{i,:}) - \mathbf{W}_{i,:}^{\top}\mathbf{W}_{i,:}.$$

□

Finally, for each row i, the gradient of the $\mathbf{W}_{i,:}$ with respect to $\mathbf{Z}_{i,:}$ is obtained by the chain rule

$$\nabla_{\mathbf{Z}_{i,:}}\mathscr{L} = \frac{\partial \mathscr{L}}{\partial \mathbf{W}_{i,:}} \left(\operatorname{diag}(\mathbf{W}_{i,:}) - \mathbf{W}_{i,:}^{\top}\mathbf{W}_{i,:}\right). \tag{12}$$

Stacking all row gradients together yields:

$$\nabla_{\mathbf{Z}}\mathscr{L} = \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mathbf{W}_{1,:}} \left(\operatorname{diag}(\mathbf{W}_{1,:}) - \mathbf{W}_{1,:}^{\top}\mathbf{W}_{1,:}\right) \\ \vdots \\ \frac{\partial \mathscr{L}}{\partial \mathbf{W}_{n,:}} \left(\operatorname{diag}(\mathbf{W}_{n,:}) - \mathbf{W}_{n,:}^{\top}\mathbf{W}_{n,:}\right) \end{bmatrix}. \tag{13}$$

This softmax gradient formulation enables us to optimize weight matrices indirectly through unconstrained weights. In the following subsections, we apply this principle to derive efficient update rules for all variants of self-representation models: Softmax SR, Softmax Mixture SR, and Softmax Bilinear SR. Each model combines multiplicative updates with softmax-based gradient steps tailored to its specific objective function.

Corrected Proof

4.2 Optimization of the Softmax-SR model

Let $\mathbf{Z} \in \mathbb{R}^{n\times n}$ denote the unconstrained matrix. The row-stochastic self-representation matrix $\mathbf{B}$ is obtained via the row-wise softmax mapping:

$$\mathbf{B} = \mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}). \tag{14}$$

The optimization problem of the proposed Softmax-SR model is formulated as

$$\min_{\mathbf{B}} \mathscr{L}(\mathbf{B}) = \left\|\mathbf{X} - \mathbf{X}\mathbf{B}\right\|_F^2 + \alpha\|\mathbf{B}\|_{2,1}, \quad \text{s.t. } \mathbf{B} = \mathrm{softmax}_{\mathrm{row}}(\mathbf{Z}), \tag{15}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the data matrix. Due to the coupling introduced by the softmax constraint, direct optimization with respect to $\mathbf{Z}$ is nontrivial. To address this issue, we adopt a two-stage alternating optimization strategy that iterates between: (i) an unconstrained multiplicative update on $\mathbf{B}$, and (ii) a gradient descent step on $\mathbf{Z}$ to restore the stochastic constraint.

Update strategy: The optimization process alternates between two stages: (i) a multiplicative update step, and (ii) a gradient descent step. These stages are repeated until convergence, which is determined when either the change in the objective function is smaller than a predefined threshold, or the maximum number of iterations is reached. In each iteration, the following steps are performed:

- Initialize $\mathbf{Z}$ and compute $\mathbf{B}$ using the softmax mapping.
- Apply the multiplicative update rule to improve $\mathbf{B}$.
- Project the updated $\mathbf{B}$ onto the softmax simplex by updating $\mathbf{Z}$.
- Check the convergence criterion: If the objective function has not changed significantly or the iteration limit has been reached, stop; otherwise, repeat.

The optimization alternates between these two steps until the algorithm converges, ensuring that the row-stochastic constraint on $\mathbf{B}$ is enforced through the gradient projection.

(a) Multiplicative update of B. Temporarily ignoring the softmax constraint in (14), the objective in (15) admits the following multiplicative update rule:

$$\mathbf{B} \leftarrow \mathbf{B} \odot \frac{\mathbf{X}^\top\mathbf{X}}{\mathbf{X}^\top\mathbf{X}\mathbf{B} + \alpha\mathbf{G}\mathbf{B}}, \tag{16}$$

where $\odot$ denotes the Hadamard product and $\mathbf{G} \in \mathbb{R}^{n\times n}$ is a diagonal matrix defined as

$$\mathbf{G}_{ii} = \frac{1}{2\max(\|\mathbf{B}_{i,:}\|_2, \varepsilon)}. \tag{17}$$

This update monotonically decreases the reconstruction loss while enforcing row-wise sparsity through the $\ell_{2,1}$ regularization.

Corrected Proof

(b) Gradient descent update of Z. After the multiplicative update (16), the matrix $\mathbf{B}$ may violate the row-stochastic constraint. To project the solution back onto the probability simplex, we update $\mathbf{Z}$ using the chain rule. From (14), the gradient of the objective with respect to $\mathbf{Z}$ is given by

$$\frac{\partial \mathscr{L}}{\partial \mathbf{Z}} = \frac{\partial \mathscr{L}}{\partial \mathbf{B}} \frac{\partial \mathbf{B}}{\partial \mathbf{Z}}. \tag{18}$$

For the i-th row, the Jacobian of the row-wise softmax is

$$\mathbf{J}_i = \operatorname{diag}(\mathbf{B}_{i,:}) - \mathbf{B}_{i,:}^{\top} \mathbf{B}_{i,:}. \tag{19}$$

Accordingly, the gradient with respect to $\mathbf{Z}_{i,:}$ is computed as

$$\nabla_{\mathbf{Z}_{i,:}} \mathscr{L} = \mathbf{J}_i^{\top} \nabla_{\mathbf{B}_{i,:}} \mathscr{L}. \tag{20}$$

The update of $\mathbf{Z}$ is then performed via gradient descent:

$$\mathbf{Z} \leftarrow \mathbf{Z} - \eta \nabla_{\mathbf{Z}} \mathscr{L}, \tag{21}$$

followed by restoring the softmax constraint using (14). This alternating procedure allows the optimization to benefit from efficient multiplicative updates in the unconstrained space, while maintaining the probabilistic interpretation of $\mathbf{B}$ enforced by the softmax mapping.

Algorithm 1: Optimization of the Softmax-SR Model

1: Initialize $\mathbf{Z}$ and compute $\mathbf{B}$ using (14).
2: **repeat**
3: Compute $\mathbf{G}$ according to (17).
4: Update $\mathbf{B}$ using the multiplicative update rule (16).
5: Compute the softmax Jacobians $\mathbf{J}_i$ via (19).
6: Update $\mathbf{Z}$ using the gradient descent step (21).
7: Project the updated $\mathbf{Z}$ onto the softmax simplex: $\mathbf{B} \leftarrow \operatorname{softmax}(\mathbf{Z})$.
8: **until** convergence
9: **Output** the weight matrix $\mathbf{B}$. Compute the ℓ_2-norm of each row of $\mathbf{B}$ and rank the rows in descending order according to their norms. The top k ranked rows are then selected, and the corresponding features are returned as the final output of the Softmax-SR method.

4.3 Optimization of the Softmax Mixture-SR model

Let $\mathbf{Y} \in \mathbb{R}^{m \times m}$ and $\mathbf{Z} \in \mathbb{R}^{n \times n}$ denote the unconstrained matrices corresponding to the sample-level and feature-level self-representation matrices, respectively. The column- and row-stochastic matrices $\mathbf{A}$ and $\mathbf{B}$ are obtained via the softmax mappings

$$\mathbf{A} = \operatorname{softmax}_{\text{col}}(\mathbf{Y}), \qquad \mathbf{B} = \operatorname{softmax}_{\text{row}}(\mathbf{Z}). \tag{22}$$

The objective function of the Softmax Mixture-SR model is defined as

$$\min_{\mathbf{A},\mathbf{B}} \mathscr{L}(\mathbf{A},\mathbf{B}) = \left\|\mathbf{X} - \mathbf{X}\mathbf{B}\right\|_F^2 + \alpha \left\|\mathbf{X} - \mathbf{A}\mathbf{X}\right\|_F^2 + \beta \left\|\mathbf{B}\right\|_{2,1}, \tag{23}$$

Corrected Proof

where $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the data matrix. Due to the coupled stochastic constraints imposed by the dual softmax mappings in (22), the problem in (23) is optimized using an alternating two-stage strategy. For each variable, we first perform a multiplicative update in an unconstrained space, followed by a gradient descent step on the corresponding logits to enforce the stochastic constraints.

Update strategy: The optimization alternates between multiplicative updates and gradient-based projections for both $\mathbf{A}$ and $\mathbf{B}$. Specifically, in each iteration:

- Update $\mathbf{A}$ in the unconstrained space to improve sample-level reconstruction, then project it onto the column-wise softmax simplex via $\mathbf{Y}$.
- Update $\mathbf{B}$ in the unconstrained space using multiplicative rules, then project it onto the row-wise softmax simplex via $\mathbf{Z}$.

These steps are repeated until convergence, which is determined when either the change in the objective function falls below a predefined threshold or the maximum number of iterations is reached. This ensures that both $\mathbf{A}$ and $\mathbf{B}$ remain valid probability distributions throughout the optimization.

(a) Multiplicative update of A. Ignoring the column-stochastic constraint in (22), the sample-level reconstruction term in (23) yields the following multiplicative update:

$$\mathbf{A} \leftarrow \mathbf{A} \odot \frac{\alpha \mathbf{X}\mathbf{X}^\top}{\alpha \mathbf{A}\mathbf{X}\mathbf{X}^\top}. \tag{24}$$

This step improves the sample-wise reconstruction capability of $\mathbf{A}$.

(b) Multiplicative update of B. Similarly, by temporarily discarding the row-stochastic constraint, the update of $\mathbf{B}$ is obtained as

$$\mathbf{B} \leftarrow \mathbf{B} \odot \frac{\mathbf{X}^\top \mathbf{X}}{\mathbf{X}^\top \mathbf{X}\mathbf{B} + \beta \mathbf{G}\mathbf{B}}, \tag{25}$$

where $\mathbf{G}$ is a diagonal matrix defined by

$$\mathbf{G}_{ii} = \frac{1}{2\max(\|\mathbf{B}_{i,:}\|_2, \varepsilon)}. \tag{26}$$

This update promotes row-wise sparsity in $\mathbf{B}$.

(c) Gradient descent update of Z. After applying (25), the stochasticity of $\mathbf{B}$ is restored by updating $\mathbf{Z}$. Using the chain rule, we obtain

$$\frac{\partial \mathscr{L}}{\partial \mathbf{Z}} = \frac{\partial \mathscr{L}}{\partial \mathbf{B}} \frac{\partial \mathbf{B}}{\partial \mathbf{Z}}. \tag{27}$$

For the i-th row, the Jacobian of the row-wise softmax is given by

$$\mathbf{J}_i = \operatorname{diag}(\mathbf{B}_{i,:}) - \mathbf{B}_{i,:}^\top \mathbf{B}_{i,:}. \tag{28}$$

Accordingly, the row-wise gradient is

$$\nabla_{\mathbf{Z}_{i,:}} \mathscr{L} = \mathbf{J}_i^\top \nabla_{\mathbf{B}_{i,:}} \mathscr{L}, \tag{29}$$

and $\mathbf{Z}$ is updated via

$$\mathbf{Z} \leftarrow \mathbf{Z} - \eta \nabla_{\mathbf{Z}} \mathscr{L}. \tag{30}$$

Corrected Proof

(d) Gradient descent update of Y. In a similar manner, the update of $\mathbf{Y}$ is performed to enforce the column-stochastic constraint on $\mathbf{A}$. For the j-th column, the Jacobian of the column-wise softmax is

$$\mathbf{J}_j = \operatorname{diag}(\mathbf{A}_{:,j}) - \mathbf{A}_{:,j}\mathbf{A}_{:,j}^{\top}. \tag{31}$$

Thus, the column-wise gradient is

$$\nabla_{\mathbf{Y}_{:,j}} \mathscr{L} = \mathbf{J}_j^{\top} \nabla_{\mathbf{A}_{:,j}} \mathscr{L}, \tag{32}$$

and $\mathbf{Y}$ is updated as

$$\mathbf{Y} \leftarrow \mathbf{Y} - \eta \nabla_{\mathbf{Y}} \mathscr{L}. \tag{33}$$

(e) Softmax projection. Finally, the stochastic constraints are explicitly restored via

$$\mathbf{A} = \operatorname{softmax}_{\text{col}}(\mathbf{Y}), \qquad \mathbf{B} = \operatorname{softmax}_{\text{row}}(\mathbf{Z}). \tag{34}$$

Algorithm 2: Optimization of the Softmax Mixture-SR Model

1: Initialize $\mathbf{Y}, \mathbf{Z}$ and compute $\mathbf{A}, \mathbf{B}$ using (22).
2: **repeat**
3: **Stage 1: Multiplicative Updates**
4: Update $\mathbf{A}$ using (24).
5: Compute $\mathbf{G}$ via (26) and update $\mathbf{B}$ using (25).
6: **Stage 2: Gradient-Based softmax Projection**
7: Update $\mathbf{Z}$ using (30) and project $\mathbf{B} \leftarrow \operatorname{softmax}_{\text{row}}(\mathbf{Z})$.
8: Update $\mathbf{Y}$ using (33) and project $\mathbf{A} \leftarrow \operatorname{softmax}_{\text{col}}(\mathbf{Y})$.
9: **until** convergence
10: **Output:** $\mathbf{A}, \mathbf{B}$. Compute row-wise ℓ_2-norms of $\mathbf{B}$, rank, and select top k features as the final output.

4.4 Optimization of the Softmax Bilinear-SR model

Let $\mathbf{Y} \in \mathbb{R}^{m \times m}$ and $\mathbf{Z} \in \mathbb{R}^{n \times n}$ denote the unconstrained matrices corresponding to the sample-level and feature-level self-representation matrices, respectively. The column- and row-stochastic matrices $\mathbf{A}$ and $\mathbf{B}$ are defined via the bilinear softmax mappings

$$\mathbf{A} = \operatorname{softmax}_{\text{col}}(\mathbf{Y}), \qquad \mathbf{B} = \operatorname{softmax}_{\text{row}}(\mathbf{Z}). \tag{35}$$

The objective function of the Softmax Bilinear-SR model is formulated as

$$\min_{\mathbf{A},\mathbf{B}} \mathscr{L}(\mathbf{A},\mathbf{B}) = \left\|\mathbf{X} - \mathbf{AXB}\right\|_F^2 + \alpha \left\|\mathbf{B}\right\|_{2,1}, \tag{36}$$

where $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the data matrix. Similar to the previous Softmax-SR variants, the optimization problem in (36) is solved using an alternating two-stage strategy. Specifically, multiplicative updates are first applied in an unconstrained space to improve reconstruction and sparsity, followed by gradient descent steps on the logits to enforce the bilinear softmax constraints.

Corrected Proof

Update strategy: The optimization alternates between multiplicative updates and gradient-based projections for both $\mathbf{A}$ and $\mathbf{B}$. In each iteration:

- $\mathbf{A}$ is first updated in the unconstrained space to improve sample-level reconstruction, then projected onto the column-wise softmax simplex through $\mathbf{Y}$.
- $\mathbf{B}$ is updated using a multiplicative rule to enforce sparsity, then projected onto the row-wise softmax simplex through $\mathbf{Z}$.

This alternating process continues until convergence, determined by either a small change in the objective function or reaching the maximum number of iterations. The procedure ensures that both $\mathbf{A}$ and $\mathbf{B}$ remain valid probability distributions while capturing bilinear dependencies between samples and features.

(a) Multiplicative update of A. By temporarily ignoring the column-stochastic constraint in (35), the reconstruction term in (36) yields the following multiplicative update:

$$\mathbf{A} \leftarrow \mathbf{A} \odot \frac{\mathbf{X}\mathbf{B}^\top\mathbf{X}^\top}{\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{B}^\top\mathbf{X}^\top}. \tag{37}$$

This step enhances the sample-level reconstruction prior to enforcing stochasticity.

(b) Multiplicative update of B. Similarly, discarding the row-stochastic constraint leads to the update

$$\mathbf{B} \leftarrow \mathbf{B} \odot \frac{\mathbf{X}^\top\mathbf{A}^\top\mathbf{X}}{\mathbf{X}^\top\mathbf{A}^\top\mathbf{A}\mathbf{X}\mathbf{B} + \alpha\mathbf{G}\mathbf{B}}, \tag{38}$$

where $\mathbf{G}$ is a diagonal matrix defined as

$$\mathbf{G}_{ii} = \frac{1}{2\max(\|\mathbf{B}_{i,:}\|_2, \varepsilon)}. \tag{39}$$

This update promotes row-wise sparsity in the feature-level representation.

(c) Gradient descent update of Z. After applying (38), the row-stochasticity of $\mathbf{B}$ is restored by updating $\mathbf{Z}$. Using the chain rule, we have

$$\frac{\partial\mathscr{L}}{\partial\mathbf{Z}} = \frac{\partial\mathscr{L}}{\partial\mathbf{B}}\frac{\partial\mathbf{B}}{\partial\mathbf{Z}}. \tag{40}$$

For the i-th row, the Jacobian of the row-wise softmax is

$$\mathbf{J}_i = \mathrm{diag}(\mathbf{B}_{i,:}) - \mathbf{B}_{i,:}^\top\mathbf{B}_{i,:}. \tag{41}$$

Thus, the row-wise gradient is computed as

$$\nabla_{\mathbf{Z}_{i,:}}\mathscr{L} = \mathbf{J}_i^\top\nabla_{\mathbf{B}_{i,:}}\mathscr{L}, \tag{42}$$

and $\mathbf{Z}$ is updated according to

$$\mathbf{Z} \leftarrow \mathbf{Z} - \eta\nabla_{\mathbf{Z}}\mathscr{L}. \tag{43}$$

Corrected Proof

(d) Gradient descent update of Y. To enforce the column-stochastic constraint on $\mathbf{A}$, we update $\mathbf{Y}$ via gradient descent. For the j-th column, the Jacobian of the column-wise softmax is

$$\mathbf{J}_j = \operatorname{diag}(\mathbf{A}_{:,j}) - \mathbf{A}_{:,j}\mathbf{A}_{:,j}^\top. \tag{44}$$

Accordingly, the column-wise gradient is

$$\nabla_{\mathbf{Y}_{:,j}} \mathscr{L} = \mathbf{J}_j^\top \nabla_{\mathbf{A}_{:,j}} \mathscr{L}, \tag{45}$$

and the update rule is

$$\mathbf{Y} \leftarrow \mathbf{Y} - \eta \nabla_{\mathbf{Y}} \mathscr{L}. \tag{46}$$

(e) Softmax projection. Finally, the bilinear softmax constraints are explicitly restored via

$$\mathbf{A} = \operatorname{softmax}_{\text{col}}(\mathbf{Y}), \qquad \mathbf{B} = \operatorname{softmax}_{\text{row}}(\mathbf{Z}). \tag{47}$$

Algorithm 3: Optimization of the Softmax Bilinear-SR Model

1: Initialize $\mathbf{Y}, \mathbf{Z}$ and compute $\mathbf{A}, \mathbf{B}$ using (35).
2: **repeat**
3: **Stage 1: Multiplicative Updates**
4: Update $\mathbf{A}$ using (37) to improve sample-level reconstruction.
5: Compute $\mathbf{G}$ using (39) and update $\mathbf{B}$ using (38).
6: **Stage 2: Gradient-Based softmax Projection**
7: Update $\mathbf{Z}$ using (43) and project $\mathbf{B} \leftarrow \operatorname{softmax}_{\text{row}}(\mathbf{Z})$.
8: Update $\mathbf{Y}$ using (46) and project $\mathbf{A} \leftarrow \operatorname{softmax}_{\text{col}}(\mathbf{Y})$.
9: **until** convergence
10: **Output:** $\mathbf{A}, \mathbf{B}$. Compute row-wise ℓ_2-norms of $\mathbf{B}$, rank, and select top k features as the final output.

4.5 Computational complexity and runtime analysis

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ denote the data matrix, where m and n are the feature dimension and the number of samples, respectively, and let T be the number of optimization iterations. In the following, we report the dominant computational costs under dense matrix operations. For Softmax-SR, the main cost comes from updating $\mathbf{B} \in \mathbb{R}^{n \times n}$, where the multiplication $(\mathbf{X}^\top \mathbf{X})\mathbf{B}$ dominates the complexity. Since $\mathbf{X}^\top \mathbf{X}$ can be precomputed once with cost $\mathscr{O}(mn^2)$, the per-iteration complexity is dominated by $\mathscr{O}(n^3)$, leading to an overall complexity of $\mathscr{O}(mn^2 + Tn^3)$. For Softmax Mixture-SR, both $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$ are updated alternately. After precomputing $\mathbf{X}^\top \mathbf{X}$ and $\mathbf{X}\mathbf{X}^\top$ with cost $\mathscr{O}(mn^2 + m^2 n)$, the dominant per-iteration cost becomes $\mathscr{O}(m^3 + n^3)$. Therefore, the total complexity is $\mathscr{O}(mn^2 + m^2 n + T(m^3 + n^3))$. For Softmax Bilinear-SR, the bilinear term $\|\mathbf{X} - \mathbf{A}\mathbf{X}\mathbf{B}\|_F^2$ introduces additional matrix multiplications involving both $\mathbf{A}$ and $\mathbf{B}$. Thus, its per-iteration cost is higher than the previous two models and can be conservatively bounded by $\mathscr{O}(mn^2 + m^2 n + n^3)$, yielding an overall complexity of $\mathscr{O}(T(mn^2 + m^2 n + n^3))$. In practice, with efficient matrix multiplication ordering, the effective cost can be lower. Table 1 summarizes the complexity of the proposed methods and reports the average runtime per iteration measured in Python under the same implementation setting.

Corrected Proof

Table 1: Computational complexity and average runtime per iteration (in seconds) of the proposed methods. Runtime was measured in Python under the same implementation setting.

Method	Complexity	Jaffe (s)	Yale (s)	ORL (s)
Softmax-SR	$\mathcal{O}(mn^2+Tn^3)$	13.29	46.79	47.59
Softmax Mixture-SR	$\mathcal{O}(mn^2+m^2n+T(m^3+n^3))$	13.92	48.37	52.85
Softmax Bilinear-SR	$\mathcal{O}(T(mn^2+m^2n+n^3))$	13.57	47.37	51.20

4.6 Related optimization frameworks

Recent research has demonstrated the effectiveness of hybrid optimization frameworks that combine metaheuristic search with reinforcement learning strategies in a wide range of machine learning and decision-making tasks. For instance, Paniri et al. [19] proposed Ant-TD, a hybrid method that integrates Ant Colony Optimization (ACO) with Temporal Difference (TD) reinforcement learning for multi-label feature selection. In their approach, the feature selection process is modeled as a Markov Decision Process (MDP), where candidate features represent states and ant decisions correspond to actions. The heuristic component of ACO is not fixed but learned adaptively through TD learning, allowing the algorithm to refine its search behavior based on accumulated experience. This integration improves exploration of the feature space and leads to superior classification performance compared with traditional ACO-based feature selection methods. Similarly, Zamfirache et al. [33] introduced an adaptive reinforcement learning control framework that combines Proximal Policy Optimization (PPO) with the Slime Mould Algorithm (SMA). In this hybrid scheme, SMA dynamically adjusts the learning rate of the PPO agent during the training process, enabling the learning algorithm to adapt its update step according to the quality of previous experiences. Experimental validation on a tower crane control system demonstrated that the adaptive SMA-based PPO strategy can improve control performance compared with the classical PPO approach using fixed learning rates. More recently, Zavadskas et al. [34] proposed an intelligent decision-support framework for circular-oriented project investments that integrates multiple optimization techniques, including molecular fuzzy genetic algorithms, Q-learning, and multi-objective particle swarm optimization. In their system, genetic algorithms are used for criteria weighting, while swarm optimization is employed to rank investment alternatives under uncertainty modeled by molecular fuzzy sets. This hybrid framework demonstrates how combining evolutionary optimization with learning mechanisms can support complex decision-making problems involving multiple conflicting criteria.

While the above studies [19, 33, 34] demonstrate the effectiveness of hybrid metaheuristic and reinforcement learning-based optimization strategies in diverse applications, the optimization framework proposed in this work follows a fundamentally different and more structured design philosophy. Rather than relying on stochastic search mechanisms or adaptive policy exploration, our softmax-based self-representation models adopt a deterministic optimization scheme that alternates between multiplicative updates and gradient-based softmax projections. By explicitly exploiting the analytical structure of the objective functions and embedding probabilistic constraints directly through softmax reparameterization, the proposed method ensures stable convergence, reduced computational overhead, and scalable performance. This structured update mechanism avoids the additional complexity typically associated with heuristic population-based search, thereby providing a principled and computationally efficient solution for feature representation learning. Nevertheless, inspired by the success of hybrid optimization paradigms, future research may explore the integration of metaheuristic or reinforcement learning techniques to adaptively tune hyperparameters, guide initialization strategies, or dynamically adjust update schedules. Such hybrid

Corrected Proof

extensions could further enhance convergence speed and robustness, particularly for large-scale or highly nonconvex scenarios.

5 Performance evaluation and numerical tests

This section evaluates the effectiveness of the proposed softmax-based SR frameworks through extensive numerical tests on multiple benchmark datasets. The proposed method is compared with several classical SR-based and unsupervised feature selection approaches under unified evaluation settings. Performance is assessed using standard clustering metrics to analyze both quantitative accuracy and robustness with respect to the number of selected features.

5.1 Datasets

A variety of benchmark datasets is utilized to assess the effectiveness of the proposed approach. These datasets are widely adopted in the literature for validating feature selection algorithms.

We provide more detailed descriptions and rationale for selecting each dataset, as follows:

1. **TOX_171**: This is a high-dimensional biological microarray dataset with 171 samples and 5748 features. It is commonly used to benchmark feature selection methods in genomics due to its challenging dimensionality.

2. **ORL** and **Yale**: Both are widely-used face image datasets, with ORL consisting of 400 images and Yale containing 165 images, each with 1024 features. ORL represents 40 classes, while Yale contains 15 classes. These datasets are valuable for evaluating feature selection methods in high-dimensional visual recognition tasks.

3. **Jaffe**: This dataset includes 213 facial images labeled across 10 expressions. It is a compact yet challenging dataset for testing feature selection in facial expression analysis.

These datasets were selected because they cover diverse domains, including biological microarray, facial recognition, and expression analysis. Furthermore, they vary in sample size and feature dimensionality, providing a comprehensive testbed for our approach. An overview of their basic characteristics is presented in Table 2, where m denotes the total number of samples, n represents the dimensionality of the feature space, and c indicates the number of class categories.

All datasets are publicly accessible through the Scikit-Feature repository [11]. Additionally, to facilitate reproducibility, we have provided the processed datasets and links in the supplementary material of this submission.

5.2 Comparison methods

To evaluate the effectiveness of the proposed approaches, we benchmark them against a set of well-established SR and UFS techniques. The comparative methods considered in this study are summarized below:

https://jundongl.github.io/scikit-feature/datasets.html

Corrected Proof

Table 2: Overview of the datasets used in this study, with m indicating the number of samples, n the number of features, and c the number of classes.

Dataset	m	n	c	Type of Data
TOX_171	171	5748	4	Biological microarray
ORL	400	1024	40	Face Image Data
Yale	165	1024	15	Face Image Data
Jaffe	213	676	10	Face Image

1. **Baseline**: All features in the original dataset are used without any selection.

2. **Feature-level SR** [37]: A feature-level self-representation method, where the reconstruction loss is measured using the $L_{2,1}$-norm to encourage sparsity and robustness.

3. **Mixture-level SR** [23]: A mixture-level self-representation framework that simultaneously models self-representation across both samples and features by integrating two complementary reconstruction terms.

4. **Bilinear SR** [1]: The bilinear self-representation framework simultaneously models the intrinsic structures of the sample and feature spaces, facilitating their mutual interaction for feature selection.

5. **FWHSR** [9]: An unsupervised feature selection method based on feature-weighted hypergraph and clustering similarity self-representation. This approach introduces a feature weighting mechanism to compute sample similarities and constructs a hypergraph structure to capture higher-order relationships among samples. In addition, clustering similarity self-representation learning is employed to uncover latent global clustering structures, thereby enhancing the robustness and quality of the learned representation space for feature selection.

6. **GRSSLFS** [31]: A graph-regularized self-representation learning method for unsupervised feature selection using a feature space basis. The GRSSLFS first constructs a basis for the feature space by selecting features with the highest variance, aiming to reduce the influence of redundant and noisy features. It then integrates subspace learning and matrix factorization within a self-representation framework, while incorporating a manifold regularization term to preserve the intrinsic geometric structure of the data.

5.3 Testing and evaluation settings

To provide a fair and consistent comparison between our approach and competing methods, we unified the experimental protocol across all algorithms. This protocol specifies (i) hyperparameter tuning ranges, (ii) the number of selected features, (iii) stopping criteria for iterative procedures, and (iv) the evaluation procedure after feature selection. Establishing this common setup ensures that all methods are assessed under comparable and reproducible conditions.

The regularization parameters for each model were tuned over a predefined search space, $\{10^{-6}, 10^{-3}, 10^{0}, 10^{3}, 10^{6}\}$, and the best-performing values were selected for reporting. To investigate the impact of feature dimensionality, the number of selected features was varied within the set $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$.

Corrected Proof

For the proposed softmax-based models, we optimize unconstrained variables and map them to the feasible domain via softmax re-parameterizations (row-wise or column-wise, depending on the model). The corresponding variables were initialized before applying the softmax mapping, and the iterative optimization was terminated either when the maximum number of iterations was reached or when the objective value exhibited negligible change between successive iterations. In addition, the maximum iteration limit for iterative updates was selected from $\{5, 10, 30\}$; empirically, a limit of 5 or 10 iterations was typically sufficient for convergence in our experiments. To provide a statistically more reliable evaluation, each method was executed over multiple independent runs with different random initializations. For the proposed softmax models, we repeated the full training process 10 times using different random seeds. Within each independent run, the k-means clustering step was further repeated 10 times with different initializations, and the average ACC, NMI, and Purity values were recorded for that run.

After feature selection, the k-means algorithm was applied to the reduced feature sets. The number of clusters was set equal to the ground-truth number of classes for each dataset. Since k-means is sensitive to random initialization, it was executed 20 times with different random seeds, and we report the average results over these runs. Clustering quality was assessed using three widely used metrics: ACC, NMI, and Purity; higher values indicate better agreement between the obtained clusters and the true labels.

The experimental comparisons are designed to evaluate the effect of our proposed softmax reparameterization. We therefore compare the original SR models with their softmax-enhanced counterparts. This controlled setup allows a precise assessment of the contribution of softmax on the learned weight matrices and the resulting feature selection performance. While more recent deep or hybrid methods exist, they are often based on different assumptions and settings, which makes direct comparison less suitable for isolating the effect studied in this work. To avoid any ambiguity, the manuscript has been revised to clarify why the selected baselines are most appropriate for evaluating the proposed softmax-based approach.

In our approach, the theoretical frameworks introduced in earlier sections are applied through the iterative optimization processes of the models. Specifically, in Softmax-SR, the theory is implemented by iteratively updating the feature matrix $\mathbf{B}$ using a multiplicative update rule, while ensuring that the learned features are regularized using an $\ell_{2,1}$-norm to promote sparsity. Similarly, in Softmax Mixture-SR, we extend the application of softmax mappings to both the sample-level and feature-level matrices ($\mathbf{A}$ and $\mathbf{B}$), where $\mathbf{A}$ aggregates sample relationships probabilistically, and $\mathbf{B}$ ensures feature-level interaction. For the Softmax Bilinear-SR model, the bilinear structure is directly applied to capture the interactions between both samples and features, propagating information through both levels simultaneously. These theoretical elements are carefully incorporated into the optimization steps, such as the multiplicative updates and gradient-based softmax projections, to ensure that each model effectively captures the relationships within the data and produces meaningful results.

5.4 Results and analysis

In this section, we present a comprehensive evaluation of the SR-based methods across four benchmark datasets using three standard clustering metrics: ACC, NMI, and Purity. The numerical results for these metrics are summarized in three separate tables. Specifically, Table 3 reports the ACC values, Table 4 lists the NMI outcomes, and Table 5 provides the Purity scores. Together, these tables offer a structured comparison of how each method performs on different datasets.

We also examine how the clustering performance changes as the number of selected features varies.

Corrected Proof

This dependence is depicted in Figures 2, 3, and 4, where the horizontal axis corresponds to the number of selected features and the vertical axis shows the respective ACC, NMI, or Purity metric. These visualizations provide additional insight into the sensitivity of each method to the feature subset size and reveal performance trends across varying feature dimensions.

Table 3: ACC performance comparison across different datasets

Method	Jaffe	Yale	ORL	TOX
baseline	87.17	38.79	51.70	41.34
SR	79.81	38.79	56.25	53.22
Softmax SR	**99.53**	**48.48**	58.00	56.01
Mixture SR	73.71	40.00	56.75	48.03
Softmax Mixture SR	93.43	46.06	56.51	57.01
Bilinear SR	98.59	40.61	58.00	51.46
Softmax Bilinear SR	98.59	46.67	**59.00**	59.06
GRSSLFS	94.36	47.04	53.45	55.3
FWHSR	93.45	43.91	56.41	**59.21**

Table 4: NMI performance comparison across different datasets

Method	Jaffe	Yale	ORL	TOX
baseline	87.87	44.19	70.79	12.28
SR	82.96	47.14	74.19	32.49
Softmax SR	**99.18**	**53.25**	**75.70**	43.04
Mixture SR	80.45	46.08	73.84	39.03
Softmax Mixture SR	92.15	52.91	73.03	46.06
Bilinear SR	98.18	49.26	74.86	39.41
Softmax Bilinear SR	98.18	50.42	75.64	**47.01**
GRSSLFS	95.01	51.30	74.56	41.24
FWHSR	93.70	51.25	71.35	43.90

Table 5: Purity performance comparison across different datasets

Method	Jaffe	Yale	ORL	TOX
baseline	87.17	36.41	52.40	42.01
SR	81.22	40.00	61.00	54.97
Softmax SR	**99.53**	**49.09**	63.75	62.02
Mixture SR	75.59	41.82	61.00	52.07
Softmax Mixture SR	93.43	46.06	61.02	63.03
Bilinear SR	98.59	42.42	62.50	53.22
Softmax Bilinear SR	98.59	47.27	62.25	**65.10**
GRSSLFS	94.36	47.21	60.32	57.21
FWHSR	93.45	44.22	**64.25**	60.43

- The results in Tables 3–5 show that the softmax-based SR variants consistently improve upon the baseline and, in most cases, outperform their classical SR counterparts across the four datasets and

Corrected Proof

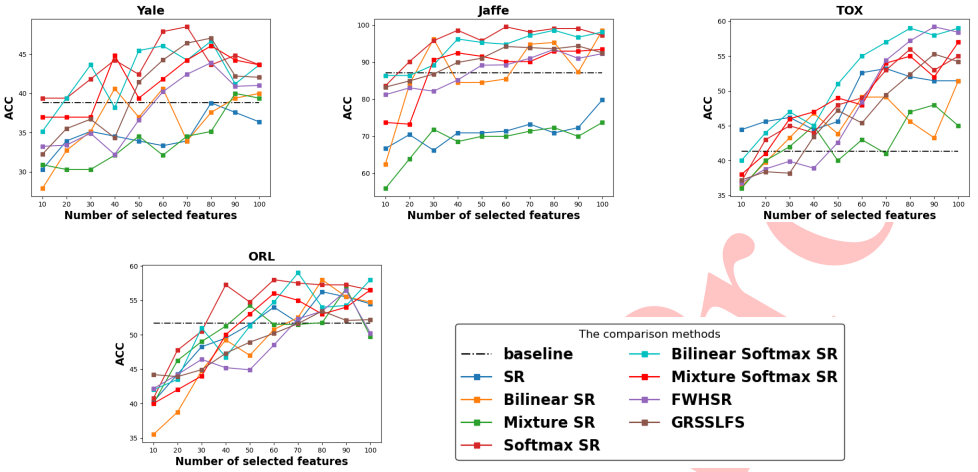

Figure 2: ACC of different methods on four datasets in terms of different numbers of selected features

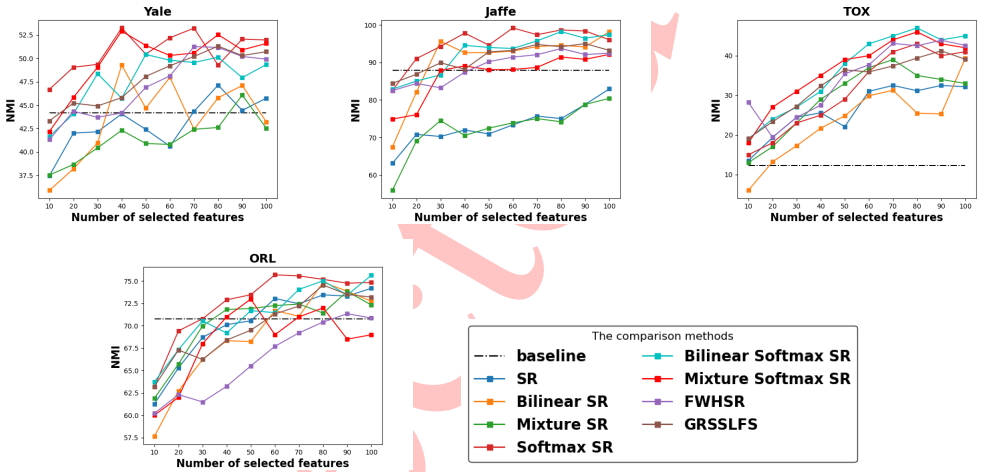

Figure 3: NMI of different methods on four datasets in terms of different numbers of selected features

three metrics. This improvement supports the premise that enforcing probabilistic normalization through softmax leads to more stable and interpretable coefficient matrices, which in turn yields higher-quality feature subsets for downstream clustering.

- Importantly, the reported clustering scores are not the outcome of a single k-means run: following the protocol described in Section 5.3, K-Means was repeated 20 times with different random seeds and the average ACC/NMI/Purity was reported. This reduces the effect of k-means initialization and provides a more reliable estimate of performance. Moreover, the performance curves in Figures 2–4 indicate that the softmax-based methods maintain competitive performance across a broad range of selected feature sizes, suggesting reduced sensitivity to the specific choice of k within the considered grid.

- Across datasets, Softmax SR achieves the strongest results on JAFFE in terms of ACC, NMI, and

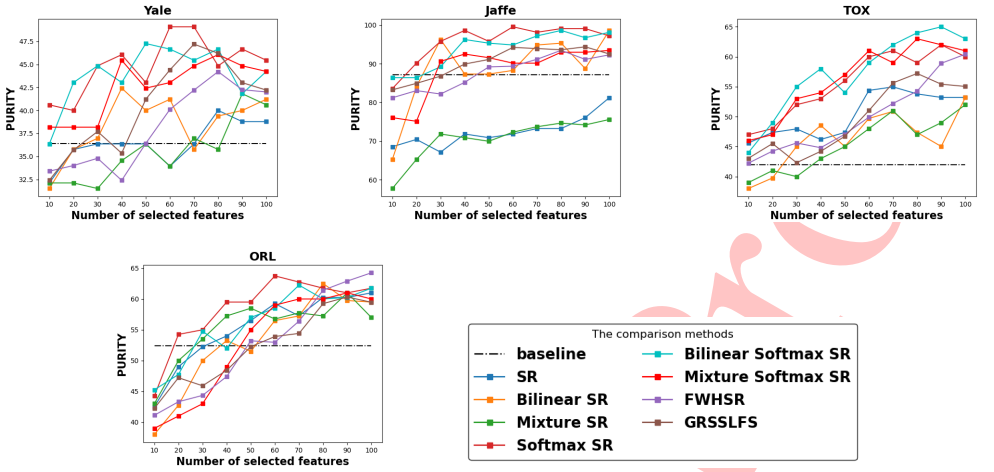

Figure 4: Purity of different methods on four datasets in terms of different numbers of selected features

Purity, whereas Softmax Bilinear SR shows particularly strong performance on TOX (notably in NMI and Purity). These patterns are consistent with the expected behavior of the models: the softmax normalization stabilizes the learned relations, while the bilinear coupling in Softmax Bilinear SR can better exploit cross-domain interactions between samples and features in more challenging settings.

5.5 Statistical evaluation

To further validate the effectiveness of the proposed softmax-based self-representation learning methods, a statistical significance analysis is conducted using the Friedman test over the ACC, NMI, and Purity metrics across all benchmark datasets. The Friedman test is a widely adopted non-parametric statistical approach for comparing multiple algorithms over multiple datasets [31]. Within this framework, datasets are treated as experimental blocks, while the compared feature selection methods are regarded as competing algorithms. A lower average rank indicates better clustering performance. The null hypothesis (H_0) assumes that all methods perform equivalently and that the observed differences are not statistically significant. The alternative hypothesis (H_1) states that at least two methods exhibit significantly different performance. Table 6 reports the average Friedman rankings for all compared methods.

It can be observed that the proposed **Softmax Bilinear SR** method consistently achieves the best average ranking across all evaluation metrics, obtaining ranks of 1.62, 1.62, and 1.38 for ACC, NMI, and Purity, respectively. These results demonstrate that incorporating the proposed softmax-based reparameterization into the bilinear self-representation framework significantly improves the discriminative capability and robustness of the learned feature representations. Among the remaining approaches, **Softmax SR** achieves the second-best rankings for all evaluation metrics, confirming the effectiveness of the proposed softmax normalization strategy even in the standard self-representation setting. In contrast, the conventional **Mixture SR** model obtains the worst rankings across all metrics, indicating that existing self-representation formulations without proper coefficient normalization may suffer from unstable or less discriminative affinity structures. The Friedman analysis further indicates that the proposed softmax-based methods consistently outperform traditional self-representation-based feature selection approaches,

Corrected Proof

Table 6: Average Friedman ranks of all compared methods over ACC, NMI, and Purity metrics. Lower ranks indicate better performance.

Method	ACC Rank	NMI Rank	Purity Rank
Baseline	6.50	6.50	7.00
SR	6.12	6.12	5.62
Softmax SR	2.12	2.12	2.12
Mixture SR	7.25	7.25	7.50
Softmax Mixture SR	3.62	3.62	4.00
Bilinear SR	4.00	4.00	3.62
Softmax Bilinear SR	**1.62**	**1.62**	**1.38**
GRSSLFS	4.50	4.50	5.12
FWHSR	4.25	4.25	3.62

including Bilinear SR, GRSSLFS, and FWHSR. In particular, the superiority of the proposed **Softmax Bilinear SR** method highlights the importance of jointly exploiting bilinear modeling and probabilistic coefficient normalization for robust unsupervised feature selection.

To further investigate the statistical significance of pairwise differences, Holm's post-hoc procedure [31] is employed by considering **Softmax Bilinear SR** as the control method. Tables 7, 8, and 9 summarize the post-hoc comparison results for ACC, NMI, and Purity, respectively.

Table 7: Holm post-hoc comparisons for the ACC metric using Softmax Bilinear SR as the control method

Comparison	p-value	Holm-adjusted	Significant
Softmax Bilinear SR vs Baseline	0.0021	0.0168	Yes
Softmax Bilinear SR vs SR	0.0105	0.0630	Yes
Softmax Bilinear SR vs Mixture SR	0.0012	0.0108	Yes
Softmax Bilinear SR vs GRSSLFS	0.0417	0.0834	No
Softmax Bilinear SR vs FWHSR	0.0635	0.0834	No
Softmax Bilinear SR vs Bilinear SR	0.0852	0.0852	No
Softmax Bilinear SR vs Softmax Mixture SR	0.1432	0.1432	No
Softmax Bilinear SR vs Softmax SR	0.3173	0.3173	No

Table 8: Holm post-hoc comparisons for the NMI metric using Softmax Bilinear SR as the control method

Comparison	p-value	Holm-adjusted	Significant
Softmax Bilinear SR vs Baseline	0.0021	0.0168	Yes
Softmax Bilinear SR vs SR	0.0105	0.0630	Yes
Softmax Bilinear SR vs Mixture SR	0.0012	0.0108	Yes
Softmax Bilinear SR vs GRSSLFS	0.0417	0.0834	No
Softmax Bilinear SR vs FWHSR	0.0635	0.0834	No
Softmax Bilinear SR vs Bilinear SR	0.0852	0.0852	No
Softmax Bilinear SR vs Softmax Mixture SR	0.1432	0.1432	No
Softmax Bilinear SR vs Softmax SR	0.3173	0.3173	No

For the ACC metric (Table 7), the proposed method achieves statistically significant improvements

Corrected Proof

Table 9: Holm post-hoc comparisons for the Purity metric using Softmax Bilinear SR as the control method

Comparison	***p*-value**	**Holm-adjusted**	**Significant**
Softmax Bilinear SR vs Baseline	0.0014	0.0112	Yes
Softmax Bilinear SR vs SR	0.0062	0.0372	Yes
Softmax Bilinear SR vs Mixture SR	0.0008	0.0072	Yes
Softmax Bilinear SR vs GRSSLFS	0.0281	0.0562	No
Softmax Bilinear SR vs FWHSR	0.0514	0.0719	No
Softmax Bilinear SR vs Bilinear SR	0.0725	0.0725	No
Softmax Bilinear SR vs Softmax Mixture SR	0.1186	0.1186	No
Softmax Bilinear SR vs Softmax SR	0.2841	0.2841	No

over the Baseline, SR, and Mixture SR methods, as their corresponding p-values are below the Holm-adjusted significance thresholds. Similar observations can be made for the NMI metric (Table 8), where Softmax Bilinear SR significantly outperforms several conventional approaches. These findings confirm the stability and effectiveness of the proposed framework in preserving discriminative cluster structures. For the Purity metric (Table 9), the proposed method again demonstrates statistically significant superiority over Baseline, SR, and Mixture SR. Although the differences with respect to more advanced approaches such as Bilinear SR, FWHSR, and Softmax SR are not statistically significant under Holm's correction, the proposed Softmax Bilinear SR method still achieves the best overall average ranking across all evaluation metrics. Overall, the statistical evaluation confirms that the proposed softmax-based self-representation learning framework provides consistent and statistically reliable improvements over existing unsupervised feature selection methods. These results further demonstrate the effectiveness of combining softmax coefficient normalization with bilinear self-representation learning for robust clustering-oriented feature selection.

6 Conclusion

In this paper, we introduced a unified softmax-based self-representation (SR) learning framework for unsupervised feature selection, addressing the fundamental limitations of classical SR models regarding scale and interpretability. By employing a softmax reparameterization strategy, our approach transforms weight matrices into normalized probabilistic distributions. This innovation enforces natural competition among samples and features, ensures numerical stability, and allows for unconstrained gradient-based optimization while maintaining discriminative power.

We developed three progressively structured variants: Softmax SR, Softmax Mixture SR, and Softmax Bilinear SR. Extensive experiments on benchmark datasets consistently demonstrate the superiority of these methods over state-of-the-art approaches. Specifically, the Softmax Bilinear SR model achieved the most robust performance in terms of clustering accuracy, robustness, and feature discriminability by jointly modeling dual-view dependencies within a unified probabilistic framework.

The proposed softmax-based Self Representation (SR) framework offers several advantages by enforcing probabilistic constraints on the weight matrices, leading to more stable and interpretable feature-selection coefficients and improved robustness in clustering results. Moreover, the framework is flexible and can be extended to mixture and bilinear SR variants to capture more complex sample–feature interactions. However, the softmax mapping introduces additional nonlinearity that may increase com-

Corrected Proof

putational cost, and the probabilistic normalization can reduce sensitivity to subtle feature differences. In addition, the iterative optimization process remains dependent on hyperparameter tuning to achieve optimal performance. Future work will investigate scalable extensions of the proposed framework for large-scale and streaming data scenarios. It will also explore adaptive optimization schemes to further improve robustness and feature-selection performance.

Acknowledgement

The research of the second author was conducted using funds from Afzalipour Research Institute, Shahid Bahonar University of Kerman.

Conflict of Interest

The authors declare no competing financial interests or personal relationships that could have influenced this work.

References

[1] H.N. Assadi, F. Kyanfar, F. Saberi-Movahed, A. Salemi, *Bilinear Self-Representation for Unsupervised Feature Selection with Structure Learning*, Neurocomputing **625** (2025) 129557.

[2] L. Chen, M. Zhou, W. Su, M. Wu, J. She, K. Hirota, *Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction*, Inf. Sci. **428** (2018) 49–61.

[3] M. Duan, P. Song, S. Zhou, Y. Cheng, J. Mu, W. Zheng, *High-order correlation preserved multi-view unsupervised feature selection*, Eng. Appl. Artif. Intell. **139** (2025) 109507.

[4] E. Elhamifar, R. Vidal, *Sparse subspace clustering: Algorithm, theory, and applications*, IEEE Trans. Pattern Anal. Mach. Intell. **35** (2013) 2765–2781.

[5] Y. Guo, Y. Sun, Z. Wang, F. Nie, F. Wang, *Double-structured sparsity guided flexible embedding learning for unsupervised feature selection*, IEEE Trans. Neural Netw. Learn. Syst. **35** (2023) 13354–13367.

[6] D. Han, T. Ye, Y. Han, Z. Xia, S. Pan, P. Wan, S. Song, G. Huang, *Agent attention: On the integration of softmax and linear attention*, in: European Conference on Computer Vision (ECCV), Springer, 2024, pp. 124–140.

[7] T. Hasegawa, K. Fujino, S. Sakai, *Analytical softmax temperature setting from feature dimensions for model- and domain-robust classification*, Neural Comput. Appl. **37** (2025) 27985–28016.

[8] Z. Hu, J. Wang, J. Mandziuk, Z. Ren, N.R. Pal, *Unsupervised feature selection for high-order embedding learning and sparse learning*, IEEE Trans. Cybern. **55** (2025) 2355–2368.

Corrected Proof

[9] C. Huang, L. Huang, C. Guo, Z. Li, X. Huang, *FWHSR: An unsupervised feature selection framework via feature-weighted hypergraph and clustering similarity self-representation*, Neurocomputing **683** (2026) 133507.

[10] S. Lazarescu, F. Istudor, O. Ionescu, D. Dragomir, M. Ion, L. Szekely, S. Olasz, I. Tusez, G. Ionescu, S.P. Zaoutsos, C. Niculae, B. Sinatoma, *Personalized movement algorithms for neural forearm prostheses using convolutional neural networks*, Romanian J. Inf. Sci. Technol. **28** (2025) 377–384.

[11] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, *Feature selection: A data perspective*, ACM Comput. Surv. **50** (2017) 1–45.

[12] W. Li, Y. Li, Y. Zhu, *Dual low-rank constrained self-representation learning for unsupervised SAR image change detection*, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **19** (2026) 11165–11181.

[13] Q. Lv, L. Geng, Z. Cao, M. Cao, S. Li, W. Li, G. Fu, *Adaptive sparse softmax: An effective and efficient softmax variant*, IEEE/ACM Trans. Audio Speech Lang. Process. **33** (2025) 3148-3159.

[14] J. Miao, X. Zhang, T. Yang, C. Fan, Y. Tian, Y. Shi, M. Xu, *A comprehensive survey on subspace clustering: Methods and applications*, Artif. Intell. Rev. **58** (2025) 346.

[15] A. Moslemi, *A tutorial-based survey on feature selection: Recent advancements on feature selection*, Eng. Appl. Artif. Intell. **126** (2023) 107136.

[16] A. Moslemi, M. Bidar, *Dual-dual subspace learning with low-rank consideration for feature selection*, Physica A **651** (2024) 129997.

[17] M. Mozafari, S.A. Seyedi, R.P. Mohammadiani, F.A. Tab, *Unsupervised feature selection using orthogonal encoder-decoder factorization*, Inf. Sci. **663** (2024) 120277.

[18] K.M. Nakanishi, *Scalable-softmax is superior for attention*, arXiv preprint (2025).

[19] M. Paniri, M.B. Dowlatshahi, H. Nezamabadi-pour, *Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection*, Swarm Evol. Comput. **64** (2021) 100892.

[20] M.G. Parsa, H. Zare, M. Ghatee, *Unsupervised feature selection based on adaptive similarity learning and subspace clustering*, Eng. Appl. Artif. Intell. **95** (2020) 103855.

[21] R. Sheikhpour, M.A. Sarram, S. Gharaghani, M.A.Z. Chahooki, *A survey on semi-supervised feature selection methods*, Pattern Recognit. **64** (2017) 141–158.

[22] R. Sheikhpour, *A local spline regression-based framework for semi-supervised sparse feature selection*, Knowl.-Based Syst. **262** (2023) 110265.

[23] R. Shang, Z. Zhang, L. Jiao, C. Liu, Y. Li, *Self-representation based dual-graph regularized feature selection clustering*, Neurocomputing **171** (2016) 1242–1253.

[24] R. Shang, W. Wang, R. Stolkin, L. Jiao, *Subspace learning-based graph regularized feature selection*, Knowl.-Based Syst. **112** (2016) 152–165.

Corrected Proof

[25] R. Shang, K. Xu, F. Shang, L. Jiao, *Sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection*, Knowl.-Based Syst. **187** (2020) 104830.

[26] R. Shang, C. Liu, W. Zhang, Y. Li, S. Xu, *Unsupervised feature selection method based on dual manifold learning and dual spatial latent representation*, Expert Syst. Appl. **255** (2024) 124696.

[27] X. Song, *Self-representation with adaptive loss minimization via doubly stochastic graph regularization for robust unsupervised feature selection*, Int. J. Mach. Learn. Cybern. **16** (2025) 661–685.

[28] D. Theng, K.K. Bhoyar, *Feature selection techniques for machine learning: a survey of more than two decades of research*, Knowl. Inf. Syst. (2023) 1–63.

[29] D. Theng, K.K. Bhoyar, *Feature selection techniques for machine learning: a survey of more than two decades of research*, Knowl. Inf. Syst. **66** (2024) 1575–1637.

[30] Y. Tian, D. Su, S. Lauria, X. Liu, *Recent advances on loss functions in deep learning for computer vision*, Neurocomputing **497** (2022) 129–158.

[31] P. Tiwari, F. Saberi-Movahed, S. Karami, F. Saberi-Movahed, J. Lehmann, S. Vahdati, *A self-representation learning method for unsupervised feature selection using feature space basis*, Trans. Mach. Learn. Res. (2024).

[32] X. Wang, R. Shang, C. Liu, Y. Li, S. Xu, *Unsupervised feature selection based on dual-graph clustering learning and adaptive weighting*, Pattern Recognit. **178** (2026) 113401.

[33] I.A. Zamfirache, R.-E. Precup, E.M. Petriu, *Adaptive reinforcement learning-based control using proximal policy optimization and slime mould algorithm with experimental tower crane system validation*, Appl. Soft Comput. **160** (2024) 111687.

[34] E.K. Zavadskas, H. Dinçer, S. Yüksel, S. Eti, *Intelligent expert systems using molecular fuzzy genetic algorithms and multi-objective particle swarm optimization for circular-oriented project investments*, Romanian J. Inf. Sci. Technol. **28** (3) (2025) 260–273.

[35] W. Zhao, Q. Gao, S. Mei, M. Yang, *Contrastive self-representation learning for data clustering*, Neural Netw. **167** (2023) 648–655.

[36] C. Zheng, Y. Gao, G. Chen, H. Shi, J. Xiong, X. Ren, C. Huang, Z. Li, Y. Li, *Self-adjust softmax*, in: Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2025, pp. 7827–7847.

[37] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S.C.K. Shiu, *Unsupervised feature selection by regularized self-representation*, Pattern Recognit. **48** (2015) 438–446.

Corrected Proof