i-IF-Learn: Iterative Feature Selection and Unsupervised Learning for High-Dimensional Complex Data

Chen Ma
SUSTech

Wanjie Wang
NUS

Shuhao Fan
NUS

Abstract

Unsupervised learning of high-dimensional data is challenging due to irrelevant or noisy features obscuring underlying structures. It's common that only a few features, called the influential features, meaningfully define the clusters. Recovering these influential features is helpful in data interpretation and clustering. We propose *i-IF-Learn*, an iterative unsupervised framework that jointly performs feature selection and clustering. Our core innovation is an adaptive feature selection statistic that effectively combines pseudo-label supervision with unsupervised signals, dynamically adjusting based on intermediate label reliability to mitigate error propagation common in iterative frameworks. Leveraging low-dimensional embeddings (PCA or Laplacian eigenmaps) followed by k-means, i-IF-Learn simultaneously outputs influential feature subset and clustering labels. Numerical experiments on gene microarray and single-cell RNA-seq datasets show that i-IF-Learn significantly surpasses classical and deep clustering baselines. Furthermore, using our selected influential features as preprocessing substantially enhances downstream deep models such as DeepCluster, UMAP, and VAE, highlighting the importance and effectiveness of targeted feature selection.

1 INTRODUCTION

Unsupervised learning, or clustering, is a fundamental learning task in various domains, including computer vision, natural language processing, and biomedical data analysis. Suppose $X_i \in \mathbb{R}^p$ are observed, $1 \leq i \leq n$, each with a dimension of p. Clustering is to recover a label vector $\ell \in \{1, \cdots, K\}^n$, which reveals the hidden group structure of these n data points. Unlike supervised learning, there is no prior knowledge of ℓ and direct optimization methods are not available.

Nowadays, researchers are facing challenges of complex data. For example, the feature dimension is much larger than the sample size, i.e., $p \gg n$, but many features may be irrelevant or even misleading for clustering. Simply applying a clustering algorithm on all dimensions often results in poor performance due to the curse of dimensionality (Donoho et al., 2000; Elman et al., 2020). The features related to the intrinsic structure, which we call the influential features, are comparatively sparse. A feature selection step is critical. It identifies these influential features, which offers insights in the dataset and scientific problem. Furthermore, clustering methods can be applied on these influential features.

Even with a correct set of influential features, recovering the label vector ℓ remains difficult due to complex dependencies and low signal-to-noise ratios. Low-dimensional embedding methods (Belkin and Niyogi, 2003; McInnes et al., 2020; Kingma and Welling, 2013) can extract the manifold structure and suppress noise. Incorporating these embeddings into the clustering step should enhance performance.

An iterative framework appears particularly promising, where estimated labels from previous iterations guide subsequent rounds of feature selection and clustering. However, iterative approaches face inherent challenges: early clustering errors may propagate, potentially reinforcing misleading patterns. Balancing the exploitation of early insights with safeguards against error propagation is thus essential.

1.1 Related Work

Clustering has been widely studied, with classical algorithms such as k-means (Hartigan and Wong, 1979), DBSCAN (Ester et al., 1996), and spectral clustering (Lee et al., 2010). While effective in low dimensions, these methods are sensitive to irrelevant features in

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

high-dimensional data. To mitigate this, unsupervised feature selection approaches have been developed, see surveys (Li et al., 2017; Zhao et al., 2019). For example, sparse k-means (Witten and Tibshirani, 2010; Zhang et al., 2020) incorporates feature weighting, whereas Influential Features PCA (IFPCA) (Jin et al., 2017; Jin and Wang, 2016) and IF Variational Auto-Encoder (IFVAE) (Chen et al., 2023) integrate feature screening with clustering.

Methods without explicit feature selection have also been proposed; see Kiselev et al. (2017); Satija et al. (2015). Examples include manifold fitting (Yao et al., 2024), deep learning-based clustering methods (Li et al., 2020a; Svirsky and Lindenbaum, 2024), and latent representation approaches (Jiang et al., 2017). These approaches focus on data reconstruction or latent representations. However, such methods cannot provide a direct understanding of the original features. Recent works address interpretability through differentiable feature selection (Lindenbaum et al., 2021; Lee et al., 2022; Upadhya and Cohen, 2024; Qiu et al., 2024), under the assumptions of dense features, semi-supervised settings, scenarios where $n \gg p$ or specific tpye of data. Furthermore, several advanced clustering frameworks (Eisenberg et al., 2025; Li et al., 2020b; Qiu et al., 2024) demand supplementary information.

Recently, there has been significant interest in iterative clustering frameworks. Deep clustering methods such as DEC (Xie et al., 2016) and DeepCluster (Caron et al., 2018) iteratively refine neural embeddings and cluster assignments, achieving strong empirical results in large-scale image and text datasets. However, these neural network-based methods often lack interpretability, limiting their utility in domains like genomics, where feature relevance insights are crucial. Methods like IDC (Svirsky and Lindenbaum, 2024) and CLEAR (Han et al., 2022) are also designed to improve interpretability, incorporated feature selection into clustering frameworks.

Outside the clustering context, feature selection in an unsupervised setting is also of great interest (Wu and Cheng, 2021; Boutemedjet et al., 2007) to understand the data. Algorithms suggest that clustering labels could guide on feature selection (Boutsidis et al., 2009; Lindenbaum et al., 2021). However, these methods perform feature selection independently from clustering, missing opportunities for iterative joint refinement.

1.2 Our Contribution

We propose the iterative Influential Features Learning (**i-IF-Learn**) framework, explicitly designed for joint feature selection and clustering in high-dimensional, noisy datasets. Our approach integrates adaptive feature selection directly into the clustering pipeline, iteratively improving both clustering accuracy and feature interpretability. Our main contributions include:

- We introduce a novel composite statistic that adaptively balances supervised (pseudo-label-based) and unsupervised statistics for feature selection. Based on the reliability of the pseudo-labels, our statistic dynamically adjusting the reliance on them to mitigate error propagation in iterative clustering frameworks.
- We develop two embedding-based clustering variants within our framework, i-IF-PCA and i-IF-Lap, employing PCA and Laplacian eigenmaps, respectively. Empirical results indicate superior performance of nonlinear embeddings (i-IF-Lap), highlighting the benefits of capturing complex manifold structures.
- Our method simultaneously outputs cluster labels and an interpretable set of influential features. This feature subset substantially enhances downstream analyses, improving the clustering performance of state-of-the-art methods like DeepCluster, UMAP, and VAE when used as preprocessing.
- We establish consistency results for both label recovery and feature selection under a weak signal model. Experiments on microarray and single-cell RNA-seq datasets show superior performance over classical and deep clustering methods. Furthermore, i-IF-Lap+deep clustering methods leads to significant improvements.

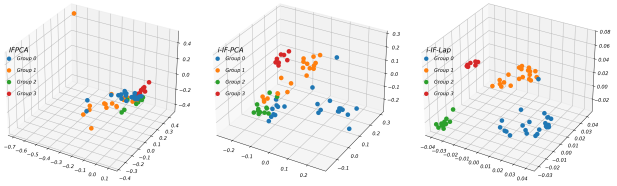

Figure 1: Eigenvector projection of data points in 3D space using three clustering pipelines.

To illustrate the effect of iteration and embedding choice, we analyze SRBCT dataset using three methods: (a) IFPCA, a non-iterative baseline(Jin and Wang, 2016); (b) i-IF-PCA, our iterative variant using PCA; and (c) i-IF-Lap, using Laplacian eigenmaps for nonlinear embedding. Figure 1 displays the resulting 3D embeddings by these methods, colored by ground-truth labels. With the iteration step, both i-IF-PCA and i-IF-Lap have a better embedding than IFPCA. Further, due to the nonlinear property of gene microarray

data, i-IF-Lap using a Laplacian eigenmap embedding performs better than i-IF-PCA. The comparison shows that both iteration and nonlinear embeddings substantially improve cluster separation. More comprehensive numerical results can be found in Sections 4 and 5.

1.3 Organization

We first introduce our i-IF-Learn framework in Section 2 with technical details. In Section 3, we introduce the model and theoretical results. Numerical results can be found in Section 4 on real data sets and Section 5 on synthetic data. Proofs, implementation details, and data description are left to the appendix.

2 ALGORITHM: ITERATIVE HIGH-DIMENSIONAL CLUSTERING

Consider the data matrix $X \in \mathbb{R}^{n\times p}$, where each row is $X_i \in \mathbb{R}^p$ with a high dimension p. For each data point, the label is denoted as $\ell_i \in [K]$ for a known constant K, where K is the number of clusters. Denote I to be the set of these influential features, where $I = \{1 \le j \le p : E[X_{ij}|\ell_i = k_1] \ne E[X_{ij}|\ell_i = k_2],$ for some $1 \le k_1 \ne k_2 \le K\}$, k_1 and k_2 represent two arbitrary distinct cluster labels in the set $\{1, \dots, K\}$. It means if there are two clusters with different expectations on this feature, then this feature if influential feature. We aim to recover both the latent cluster labels ℓ and the set of influential features I that drive the clustering structure.

We propose the i-IF-Learn algorithm, an iterative clustering framework with an initialization stage and an iterative refinement loop, as illustrated in Figure 2. The initialization stage recovers relatively strong signals with a noisy clustering label, and the iterative loop further recovers the weak signals and refines the clustering assignments. It is outlined as Algorithm 1. In Sections 2.1–2.5, we explain the idea of each step. The complete algorithm with every implementation detail can be found in Algorithm 4 in Appendix.

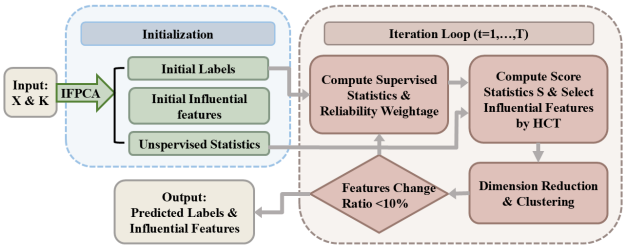

Figure 2: Overview of the i-IF-Learn framework.

Algorithm 1 i-IF-Learn Algorithm

Require: Data matrix $X \in \mathbb{R}^{n\times p}$, number of clusters K, maximum iterations T
Ensure: Clustering labels $\hat{\ell}$, selected features $\hat{I}$.
Stage 1: Initialization
Require: Data $X \in \mathbb{R}^{n\times p}$, number of clusters K
Ensure: Initial pseudo-labels $\ell^{(0)}$ and feature set $I^{(0)}$
Stage 2: Iterative Loop
for $t = 1, 2, \dots$ **do**
  Step 1: For each feature j, compute score $S_j^{(t)}$ based on $\ell^{(t-1)}$, set Higher Criticism threshold $\tau^{(t)}$, and update $I^{(t)} = \{1 \le j \le p : S_j \ge \tau^{(t)}\}$.
  Step 2: Construct the post-selection data matrix $X^{(t)} = X^{(I^{(t)})}$, perform low-dimensional embedding to obtain $U^{(t)}$, and run k-means on $U^{(t)}$ to get $\ell^{(t)}$.
  Step 3: Compute influential feature change ratio $r = r(I^{(t-1)}, I^{(t)})$.
  if $r \le 10\%$ or $t = T$ **then**
    break
  end if
end for
$\hat{\ell} = \ell^{(t)}$, $\hat{I} = I^{(t)}$.

In IF step, due to the large p, we employ a ranking and thresholding procedure for computation efficiency. For each feature j, we propose a composite score

$$S_j^{(t)} = \omega^{(t)} T_j^{sup}(\ell^{(t-1)}) + (1-\omega^{(t)})\, T_j^{unsup}, \quad (1)$$

where $T_j^{sup}(\ell^{(t-1)})$ is a pseudo-label-supervised test statistic and T_j^{unsup} is an unsupervised statistic.

The weight for the supervised statistic is defined as

$$\omega^{(t)} = \frac{w^{(t)}}{\sqrt{(w^{(t)})^2 + (1-w^{(t)})^2}}, \quad (2)$$

where $w^{(t)}$ reflects the reliability of the label estimates $\ell^{(t-1)}$, and $(1-\omega)$ correspondingly represents the weight assigned to the unsupervised statistic.

We then compute p-values based on $S_j^{(t)}$ and apply Higher Criticism thresholding (HCT) to obtain the updated influential feature set $I^{(t)}$. This procedure is entirely data-driven and requires no tuning parameters; details are in Sections 2.2–2.3.

In the Learn step, we perform low-dimensional embedding of $X^{(I^{(t)})}$ on the selected features $I^{(t)}$. The algorithm has variants depending on the selection of embedding methods. We call it **i-IF-PCA** when PCA is employed for embedding and **i-IF-Lap** when Laplacian eigemap is used for embedding. To capture the low-dimensional manifold structure, i-IF-Lap usually performs better. K-means clustering is then applied to the embedded data to generate updated labels $\ell^{(t)}$.

The algorithm continues until the feature set $I^{(t)}$ stabilizes or a maximum number of iterations is reached. The final outputs are the estimated labels $\hat{\ell} = \ell^{(T)}$ and influential feature set $\hat{I} = I^{(T)}$. The initialization step has a computation cost of $O(n + ns)$ where s is the number of selected features. For every iteration, the complexity for S_j is $O(n)$ and for clustering is $O(n^2 s)$. Therefore, the overall computation complexity is $O(Tn^2 s)$, where T is the number of iterations, n is the sample size and s is the number of selected features.

By the iterative framework, i-IF-Learn recovers features with both the relatively strong signals and weak signals. To illustrate the effectiveness of $\hat{I}$, we apply multiple clustering methods to data restricted on $\hat{I}$ to get the cluster labels, instead of the estimated labels by i-IF-Lap itself. In numerical analysis, we consider UMAP, DeepCluster, and VAE as the downstream clustering methods, and all of them have a significant improvement in clustering. It further proves that our algorithm provides interpretable results that deepens our understanding of data.

2.1 Initialization

The i-IF-Learn algorithm begins with an initialization step to estimate the initial cluster labels $\ell^{(0)}$ and the influential feature set $I^{(0)}$. While our theoretical guarantees allow for any reasonable initialization, we adopt the IFPCA method (Jin and Wang, 2016) for two main reasons. First, IFPCA has demonstrated strong theoretical and empirical performance in high-dimensional clustering. It reliably identifies features with relatively strong signals and produces stable clustering labels. Second, our iterative procedure leverages both T^{sup} and T^{unsup}. The unsupervised component T^{unsup} coincides with the IF step in IFPCA, reducing the computation cost. The IFPCA procedure is summarized in Algorithm 2, with implementation details deferred to Algorithm 3 in Appendix.

Algorithm 2 IFPCA Initialization Procedure

Require: Data $X \in \mathbb{R}^{n\times p}$, number of clusters K
Ensure: Initial labels $\ell^{(0)}$, influential feature set $I^{(0)}$
1: **Step 1**: Compute unsupervised test scores
$\psi_{n,j} \leftarrow$ Kolmogorove-Smirnov score between the empirical CDF of x_j and normal CDF.
Normalize scores: $\psi^*_{n,j} \leftarrow \frac{\psi_{n,j} - \text{mean}(\psi_{n,\cdot})}{\text{std}(\psi_{n,\cdot})}$
2: **Step 2**: Feature selection by HCT
$\pi_j \leftarrow 1 - F_0(\psi^*_{n,j})$, where F_0 is the null distribution.
$HC_j \leftarrow$ a function based on π_j
3: $\hat{j} \leftarrow \arg\max_j HC_{p,j}$, $t_p^{\text{HC}} \leftarrow \psi^*_{n,\hat{j}}$
Selected features: $I^{(0)} \leftarrow \{1 \le j \le p \mid \psi^*_{n,j} > t_p^{\text{HC}}\}$
4: **Step 3**: PCA embedding and k-means clustering
Apply PCA to post-selection data, retain top $K-1$ components. Denote it as U.
Labels: $\ell^{(0)} \leftarrow k\text{-means}(U, K)$

2.2 Novel Iterative Screening Statistic

A key ingredient in the IF step is the score statistic $S_j^{(t)}$, which determines how relevant each feature is to the underlying structure. Unlike traditional screening methods that rely solely on either supervised or unsupervised tests, we introduce a new composite statistic that adaptively combines both sources of information as follows:

$$S_j^{(t)} = \omega^{(t)}\Phi^{-1}(1-P_{F,j}^{(t)}) + (1-\omega^{(t)})\Phi^{-1}(1-P_{KS,j}). \quad (3)$$

Here, $T_j^{sup}(\ell^{(t-1)}) = \Phi(1 - P_{F,j}^{(t)})$ denotes the supervised test statistic and $T_j^{unsup} = \Phi^{-1}(1 - P_{KS,j})$ denotes the unsupervised test statistic. In detail, $P_{F,j}^{(t)}$ is the p-value of marginal F-statistic, using x_j and the current label estimates $\ell^{(t-1)}$. $P_{KS,j}$ is the p-value of the Kolmogorov-Smirnov (KS) statistic between the empirical distribution of x_j for feature j and a specified null distribution (Smirnov, 1939). While our framework is general and can accommodate any appropriate null distribution based on the specific domain, in this work we adopt the standard normal distribution as the null. This choice aligns with the rare and weak signal setting commonly assumed for high-dimensional genetics data (Jin and Wang, 2016). $P_{KS,j}$ remains static among iterations while $P_{F,j}^{(t)}$ depends on the pseudo-label in every iteration. Both are corrected to get rid of the gap between empirical null and theoretical null (Efron, 2004; Jin and Wang, 2016). The KS statistic $P_{KS,j}$ captures distributional deviation, while the F-statistic $P_{F,j}^{(t)}$ measures separation across pseudo-label clusters. Their associated p-values provide a statistically grounded measure of feature importance. Finally, we transform them into normal quantiles to calculate $S_j^{(t)}$, instead of using p-values (Wang et al., 2022) or the original statistics. Hence, the selected features are interpretable.

The **reliability weightage** $w^{(t)} \in [0, 1]$ is to evaluate our trust in the current estimated label $\hat{\ell}^{(t-1)}$. When the pseudo-label $\ell^{(t)}$ is more reliable, we tend to have a larger $w^{(t)}$ on the supervised statistic, with a higher power in feature selection. However, without the ground-truth of labels, the trust in $\ell^{(t)}$ is difficult to evaluate.

The idea is, if the selected features in previous step $l^{(t-1)}$ is further away from noises, then the predicted label $\ell^{(t)}$ based on $l^{(t-1)}$ should be more reliable. Consider the set of p-values from F-statistics $\{P_{F,j}; j \in$

$I^{(t-1)}\}$. We conduct a hypothesis testing on whether $P_{F,j}$ contains information or not, using this set. Let $p_1^{(t)} \in (0,1)$ denote the p-value of this test, where the details can be found in Appendix A.1. A smaller $p_1^{(t)}$ indicates a larger possibility that $I^{(t-1)}$ is informative. The definition of $w^{(t)}$ as follows:

$$w^{(t)} = 1 - p_1^{(t)}/(p_1^{(t)} + c). \tag{4}$$

The constant c gives the default importance of $P_{F,j}$, at $c/(1+c)$. Even when $p_1^{(t)} \to 1$, i.e., the beginning stage, we still hope $P_{F,j}$ to take part in. When $p_1^{(t)} \to 0$, the weight gradually approach 1, no matter what c is. A reasonable range for c is $0.35 \le c \le 0.6$. In numerical analysis, we consistently use 0.6, with an experiment on the effects of c in Section 5.

2.3 Feature Selection by HCT

Our novel score statistic $S_j^{(t)}$ ranks the importance of features, where a larger score $S_j^{(t)}$ indicates a larger potential for feature j to be influential. Hence, the selection follows that, for a threshold $\tau^{(t)}$,

$$I^{(t)} = \{1 \le j \le p \mid S_j^{(t)} \ge \tau^{(t)}\}. \tag{5}$$

The key is to decide $\tau^{(t)}$ in every iteration. We apply HCT, a data-driven threshold that optimizes the selection. First derive the p-values $\pi_j^{(t)} = 1 - \Phi(S_j^{(t)})$, then order the p-values in an increasing order, $\pi_{(1)}^{(t)} \le \pi_{(2)}^{(t)} \le \cdots \le \pi_{(p)}^{(t)}$. The HCT is defined as

$$\tau^{(t)} = S_{\hat{j}}^{(t)}, \tag{6}$$

where $\hat{j} = \arg \max_{\log p \le j \le p/2} (j/p - \pi_{(j)}^{(t)})/\sqrt{\pi_{(j)}^{(t)}(1-\pi_{(j)}^{(t)})}$

Using this HCT in (5), the $\hat{j}$ features with largest scores, i.e. smallest p-values, are selected. The selected features are considered as the influential features.

The corresponding post-selection data matrix follows that $X^{(t)} = X[:, I^{(t)}]$.

2.4 Embedding and Clustering

After selecting influential features, we normalize the corresponding submatrix $X^{(t)}$ to obtain $W(t)$, where each column has mean zero and unit variance. Despite feature selection, the high-dimensional data still contain redundant noise. Thus, we apply low-dimensional embedding techniques to extract underlying structure and enhance the signal-to-noise ratio.

We consider two embedding methods: Principal Component Analysis (PCA) and Laplacian Eigenmaps. PCA (Hotelling, 1933; Wang et al., 2022; Tong et al., 2025) projects the data onto directions of maximum variance and is widely used due to its simplicity and interpretability. However, it may fail to capture complex nonlinear structures inherent in many modern datasets. Laplacian Eigenmaps (Belkin and Niyogi, 2003), in contrast, construct a data dissimilarity graph and compute embeddings in to spectral space that preserve local geometry. This approach is particularly effective when data lie on a nonlinear manifold. Other embedding methods, such as UMAP (McInnes et al., 2020) and Autoencoder (Baldi, 2011), are also evaluated; their comparative results on real data sets can be found in the Appendix E.

For both methods on $W^{(t)}$, we consider the embeddings into $K+2$-dimensional space. Under the low-rank structure, separating K clusters only requires an embedding dimension larger than $K-1$ (Duda et al., 2001). In this work, we conservatively choose $K+2$ dimensions to ensure that the embedding preserves sufficient structural information. Therefore, we construct a spectral matrix $U^{(t)}$ from either $W^{(t)}$ or the data similarity matrix:

$$U^{(t)} = [u_1^{(t)}, u_2^{(t)}, \ldots, u_{K+2}^{(t)}] \in \mathbb{R}^{n \times (K+2)}. \tag{7}$$

Then we perform k-means on $U^{(t)}$, treating each row as a data point, to obtain pseudo-labels $\ell^{(t)} = k\text{-means}(U^{(t)}, K)$.

In real data analysis, we find that both embedding methods outperform clustering on raw features, with Laplacian Eigenmaps (i-IF-Lap) consistently yielding better results, highlighting the utility of nonlinear embeddings in complex data.

2.5 Stopping Criteria

To determine convergence, we monitor the stability of the selected influential feature sets across iterations. Let the relative change rate between iterations $t-1$ and t be $r^{(t)} = \left|I^{(t)}/I^{(t-1)}\right|/\left|I^{(t-1)}\right|$. If the change rate $r^{(t)} \le 10\%$ or the number of iterations exceeds a fixed limit (e.g., 10), we terminate the process.

3 MODEL ASSUMPTIONS AND THEORETICAL GUARANTEE

Consider the asymptotic clustering model where the signals are rare and weak. In detail, the data matrix $X_i \sim N(\mu_0 + \mu_{\ell_i}, \Sigma)$, where Σ is a diagonal matrix with diagonals σ_j^2. Let $M = [m_1, m_2, \cdots, m_K] \in \mathbb{R}^{p \times K}$, where $m_k = \Sigma^{-1/2}\mu_k$ be the normalized mean vectors.

Let M_j denote the j-th column of M. The influential features set is that $I = \{1 \le j \le p \mid \|M_j\| \neq 0\}$. Asymptotically, we assume that the signals are sparse, in the sense that when $p \to \infty$,

$$\epsilon = |I|/p \to 0. \tag{8}$$

The sparse setting causes a low signal-to-noise ratio, which is challenging in high-dimensional unsupervised learning.

While most theoretical papers focus on a uniform signal strength to decide the detection boundary, it is not the case in practice. The signal strength in individual features has severe heterogeneity. By an iterative framework, based on the extremely sparse features with relatively strong signals, we have an initial clustering label, and then recover the features with weak signals in iterations.

The following theorem explains the iterative feature selection effect. The technical conditions are not strict. In the simplified scenario that $K = 2$ with equal group size, as long as the accuracy rate is a constant larger than $1/2$, then the condition is satisfied.

Theorem 3.1. *Consider the estimated label $\hat{\ell}$, which can be the initial label $\hat{\ell}^{(0)}$ or the estimated label $\ell^{(t-1)}$ from last round. Denote $w_{ij} = E[\Sigma^{-1/2}X_{ij}]$ as the expectation for data point i on feature j, and the overall mean $\bar{w}_j = \frac{1}{n}\sum_{i=1}^n w_{ij}$. Denote $n_k(\hat{\ell}) = \sum_{i=1}^n 1\{\hat{\ell}_i = k\}$, $k \in [K]$. Suppose for the influential features in I, the community label satisfies that for a constant $c_0 > 0$,*

$$\min_{j\in I} U_j \ge c_0(\log p)^2, \tag{9}$$

$$\text{where } U_j := \sum\nolimits_{k\in[K]} \frac{1}{n_k(\hat{\ell})} \sum\nolimits_{\hat{\ell}_i=k} (w_{ij} - \bar{w}_j)^2.$$

Then our weight selection satisfies $P(w^{(t)} \ge 1 - p^{-2}) \ge 1 - p^{-2}$. Furthermore, with probability $1 - p^{-4}$, the $\hat{I}$ from IF step in i-IF-Learn satisfies that $I \subset \hat{I}$ and $|\hat{I}/I| \le C_0(\log p)^2$.

The correct recovery of I leads to a correct label recovery, as long as I contains sufficient information.

Theorem 3.2. *Suppose the assumptions of Theorem 3.1 hold. Let $\hat{I}$ denote the selected feature set and $M_{\hat{I}}$ denote the mean feature matrix M restricted on $\hat{I}$. Let $\tau_{\hat{I}}$ be the eigengap of matrix $M_{\hat{I}}$, then with a high probability, the K-means clustering error follows that*

$$Err(\hat{\ell}, \ell) \le C(n^{1/2} + |\hat{I}|^{1/2})^2/n\tau_{\hat{I}}^2. \tag{10}$$

Further, when the signal strength satisfies $\min_{j\in I}\|M_j\| \ge \log^2 p/\sqrt{n}$, then the clustering label by i-IF-Learn has that $Err(\hat{\ell}, \ell) \to 0$.

Our theorems suggest that when p is sufficiently large, one-step iteration could recover the correct labels and influential feature set from a random initialization. In practical data, since the constants are difficult to decide and p might be inadequate, we run iterations multiple times to ensure robustness and enhance practical performance.

4 REAL DATASETS

We evaluate our proposed i-IF-Learn method on a collection of 18 datasets, including 10 gene microarray datasets and 8 single-cell RNA sequencing (scRNA-seq) datasets. The datasets are publicly available at https://data.mendeley.com/datasets/cdsz2ddv3t/1 and https://data.mendeley.com/datasets/nv2x6kf5rd/1. Details and pre-processing can be found in Appendix D.2. These datasets are characterized by high dimensionality, sparse signal structure, and varying degrees of cluster separation, making them popular in literature for assessing the performance of high-dimensional clustering algorithms.

4.1 Gene Microarray Datasets

We benchmark our method on 10 gene microarray datasets. For each data set, we have a set of patients from different classes, and the gene expression levels of the same genes across all patients are recorded. These datasets have been widely used in prior clustering studies (Jin and Wang, 2016; Chen et al., 2023). They cover a range of cluster counts (from 2 to 5), with sample sizes ranging from 40 to 300 and number of genes typically in the thousands. Our goal is to tell the class from the gene expression levels data.

For each dataset, we consider two variants of our method: i-IF-Lap and i-IF-PCA. Using i-IF-Lap as a pre-processing feature selection method, we further consider i-IF-Lap+DeepCluster, i-IF-Lap+UMAP, and i-IF-Lap+VAE, where we apply DeepCluster, UMAP, and VAE to the final influential features $\hat{I}$, respectively. The benchmark methods include: (1) classical methods, such as KMeans (Hartigan and Wong, 1979) and SpecGEM (Lee et al., 2010); (2) neural networks, including DeepCluster (Caron et al., 2018), DEC (Xie et al., 2016) and UMAP (McInnes et al., 2020); (3) feature selection and clustering methods, including IF-PCA (Jin and Wang, 2016) and IFVAE (Chen et al., 2023). Additionally, the experimental results for IDC (Svirsky and Lindenbaum, 2024) are provided in Appendix F. The compute resource is in Appendix D.1. Implementation details and hyperparameter selection for all algorithms are in Appendix C.

We assess clustering quality using two complementary metrics: **Accuracy**, which measures the best-matched

Table 1: Accuracy comparison of clustering methods across 10 gene microarray datasets.

Dataset	KMeans	SpecGEM	UMAP	DEC	DeepCluster	IFPCA	IFVAE	i-IF-PCA	i-IF-Lap	i-IF-Lap+UMAP	i-IF-Lap+DeepCluster	i-IF-Lap+VAE
Brain	0.667	**0.857**	0.676 (0.07)	0.638 (0.09)	0.721 (0.06)	0.738	0.500	0.691	0.738	**0.783** (0.02)	**0.783** (0.02)	0.612 (0.05)
Breast	0.562	0.562	0.556 (0.00)	**0.629** (0.02)	0.548 (0.00)	0.594	0.565	0.623	**0.630**	0.550 (0.01)	0.582 (0.02)	0.628 (0.00)
Colon	0.548	0.516	0.500 (0.00)	0.561 (0.09)	**0.635** (0.12)	0.597	0.597	**0.629**	0.597	0.570 (0.02)	0.594 (0.01)	0.597 (0.00)
Leukemia	**0.972**	0.708	0.778 (0.00)	0.910 (0.05)	0.969 (0.01)	0.931	0.722	0.861	**0.972**	**0.972** (0.00)	0.971 (0.01)	0.971 (0.01)
Lung1	0.901	0.878	0.899 (0.03)	0.832 (0.01)	0.834 (0.14)	0.967	0.967	**0.995**	**0.995**	0.962 (0.02)	0.890 (0.00)	0.989 (0.00)
Lung2	0.783	0.567	0.507 (0.00)	0.676 (0.02)	0.777 (0.01)	0.783	0.783	0.724	**0.803**	**0.788** (0.00)	0.783 (0.00)	0.783 (0.00)
Lymphoma	**0.984**	0.774	0.571 (0.02)	0.874 (0.12)	0.618 (0.08)	0.935	0.742	0.968	**0.936**	0.853 (0.18)	0.903 (0.02)	0.647 (0.12)
Prostate	0.578	0.578	0.555 (0.01)	0.568 (0.07)	0.578 (0.01)	**0.618**	0.588	0.588	0.569	0.568 (0.00)	0.575 (0.01)	**0.616** (0.00)
SRBCT	0.556	0.492	0.543 (0.01)	0.460 (0.04)	0.546 (0.08)	0.556	0.524	0.587	**0.984**	**0.984** (0.00)	0.981 (0.01)	0.975 (0.02)
SuCancer	0.523	0.511	**0.672** (0.00)	0.569 (0.01)	0.549 (0.05)	0.667	**0.672**	0.500	0.603	**0.687** (0.05)	0.609 (0.02)	0.605 (0.02)
Rank	6.000 (3.3)	8.800 (3.3)	9.500 (2.9)	8.600 (2.9)	7.700 (3.2)	**4.100** (1.6)	6.400 (3.6)	5.400 (3.5)	**3.400** (2.8)	5.000 (3.7)	5.100 (2.2)	4.800 (3.0)
Regret	0.109 (0.1)	0.172 (0.1)	0.191 (0.1)	0.145 (0.1)	0.139 (0.1)	0.078 (0.1)	0.151 (0.2)	0.100 (0.1)	**0.041** (0.0)	**0.045** (0.0)	0.051 (0.0)	0.074 (0.1)

Table 2: ARI comparison of clustering methods across 10 gene microarray datasets.

Dataset	KMeans	SpecGEM	UMAP	DEC	DeepCluster	IFPCA	IFVAE	i-IF-PCA	i-IF-Lap	i-IF-Lap+UMAP	i-IF-Lap+DeepCluster	i-IF-Lap+VAE
Brain	0.375	**0.567**	0.450 (0.03)	0.411	0.534 (0.10)	0.481	0.189	0.468	0.546	0.547 (0.03)	**0.552** (0.02)	0.344 (0.05)
Breast	**0.116**	0.006	0.005 (0.00)	0.010	0.006 (0.00)	0.004	0.007	0.017	**0.025**	-0.006 (0.00)	0.013 (0.01)	0.017 (0.00)
Colon	**0.090**	-0.010	-0.018 (0.00)	0.030	**0.207** (0.22)	0.009	0.013	0.045	0.018	0.003 (0.01)	0.016 (0.01)	0.018 (0.00)
Leukemia	0.889	0.212	0.300 (0.00)	0.568	0.642 (0.22)	0.734	0.211	0.515	**0.890**	**0.890** (0.00)	0.885 (0.04)	**0.896** (0.04)
Lung1	0.487	0.595	0.464 (0.18)	0.239	0.623 (0.05)	0.834	0.893	**0.973**	**0.973**	0.830 (0.01)	0.426 (0.01)	0.945 (0.01)
Lung2	0.254	-0.003	-0.005 (0.00)	-0.002	0.239 (0.01)	0.254	0.240	0.096	**0.314**	**0.281** (0.00)	0.254 (0.00)	0.245 (0.00)
Lymphoma	**0.947**	0.398	0.402 (0.02)	0.652	0.467 (0.17)	0.824	0.652	**0.893**	0.880	0.738 (0.25)	0.803 (0.02)	0.505 (0.16)
Prostate	0.016	0.009	0.003 (0.00)	0.009	0.022 (0.01)	0.050	0.022	**0.026**	0.009	0.009 (0.00)	0.013 (0.00)	**0.047** (0.00)
SRBCT	0.121	0.125	0.190 (0.01)	0.082	0.162 (0.08)	0.143	0.129	0.259	**0.946**	**0.946** (0.00)	**0.938** (0.02)	0.917 (0.06)
SuCancer	-0.003	0.092	0.115 (0.00)	0.011	-0.002 (0.01)	0.102	**0.115**	-0.002	0.046	**0.145** (0.07)	0.045 (0.02)	0.042 (0.02)
Rank	5.900 (4.3)	8.100 (3.4)	9.300 (3.2)	8.500 (2.5)	6.700 (2.8)	5.700 (3.0)	7.200 (3.3)	5.200 (3.3)	**3.300** (2.4)	**5.100** (3.9)	5.500 (2.6)	5.200 (3.3)
Regret	0.187 (0.3)	0.317 (0.3)	0.326 (0.2)	0.315 (0.3)	0.226 (0.2)	0.173 (0.2)	0.269 (0.3)	0.187 (0.2)	**0.051** (0.1)	**0.078** (0.1)	0.122 (0.2)	0.119 (0.1)

proportion of true labels, and **Adjusted Rand Index (ARI)** (Hubert and Arabie, 1985), which quantifies the similarity between predicted and true clusters for unbalanced data. When the standard deviation is smaller than 0.0001, we do not report it. To summarize performance across datasets, we further report the average rank to rank the algorithm among all methods and the average regret that captures how far a method's result deviates from the best-performer on each dataset. Lower rank and regret mean better performance.

The accuracy and ARI are summarized in Table 1 and 2, respectively. In both tables, our i-IF-Lap algorithm demonstrates the most consistent and superior performance across 10 microarray datasets. It outperforms all other methods in 5 out of 10 datasets in terms of accuracy. In both tables, i-IF-Lap achieves the lowest average rank and regrets, indicating that it consistently ranks among the top-performing algorithms and captures all the information across diverse datasets.

Our i-IF-Lap algorithm not only suggests consistent clustering labels $\hat{\ell}$, but also recovers an influential feature set $\hat{I}$. Based on $\hat{I}$, DeepCluster, UMAP and VAE all enjoy lower average ranks (7.7→5, 6.5→5.1, and 5.4→4.8, respectively), compared to their performance without i-IF-Lap pre-processing. It strongly supports the consistency of the i-IF-Lap feature selection.

4.2 Single-cell RNA Sequencing Datasets

We further evaluate i-IF-Learn on 8 single-cell RNA-seq (scRNA-seq) datasets, which measures gene expression levels at the resolution of individual cells. The number of cells ranges from a few hundred to several thousand, with gene dimensions ranging from 2,000 to 10,000. Due to dropout events, scRNA-seq data are usually more sparse and noisy than gene microarray datasets.

We consider two variants of our method, i-IF-Lap and i-IF-PCA, and further explore i-IF-Lap combined with DeepCluster, UMAP, or VAE to illustrate the effect of selected features. Benchmark methods include: (1) scRNA-seq clustering baselines, such as Seurat (Satija et al., 2015), SC3 (Kiselev et al., 2017), scAMF (Yao et al., 2024), and DESC (Li et al., 2020a); (2) neural network methods, including DeepCluster (Caron et al., 2018) and UMAP (McInnes et al., 2020); and (3) feature selection combined with clustering, including IFPCA (Jin and Wang, 2016) and IFVAE (Chen et al., 2023). Additionally, the experimental results for CLEAR (Han et al., 2022) are provided in Appendix F. Compute resources are listed in Appendix D.1, and implementation details with hyperparameter choices are in Appendix C.

The accuracy and ARI are demonstrated in Tables 3 and 4, respectively. Our i-IF-Lap algorithm achieves the lowest average rank and average regret in both tables. The second best performer is DeepCluster with i-IF-Lap selected $\hat{I}$. It proves the power of deep clustering methods with i-IF-Lap pre-processing. As a summary, our i-IF-Lap algorithm provides both reliable clustering labels and influential feature set.

4.3 Statistical Significance

To rigorously evaluate performance, we conducted non-parametric statistical testing across all 18 real-world datasets. The Friedman test on the clustering results

Table 3: Accuracy comparison of clustering methods across 8 scRNA-seq datasets.

Dataset	Seurat	SC3	scAMF	DESC	UMAP	DeepCluster	IFPCA	IFVAE	i-IF-PCA	i-IF-Lap	i-IF-Lap+UMAP	i-IF-Lap+DeepCluster	i-IF-Lap+VAE
Camp1	0.643	0.788	**0.882**	**0.799**	0.673 (0.03)	0.612 (0.02)	0.738	0.706	0.738	0.740	0.687 (0.07)	0.657 (0.03)	0.640 (0.00)
Camp2	0.654	**0.778**	**0.673**	0.656	0.615 (0.01)	0.608 (0.04)	0.660	0.690	0.617	0.605	0.573 (0.00)	0.656 (0.00)	0.577 (0.03)
Darmanis	0.779	0.736	0.766	0.609	0.628 (0.03)	0.583 (0.01)	0.789	0.540	0.783	**0.785**	0.718 (0.04)	**0.793** (0.05)	0.662 (0.06)
Deng	0.534	0.563	0.646	0.563	0.559 (0.09)	0.624 (0.08)	0.828	0.652	0.802	**0.869**	0.658 (0.01)	**0.857** (0.01)	0.830 (0.07)
Goolam	0.629	0.758	0.823	0.629	0.508 (0.00)	**0.847** (0.08)	0.721	0.492	0.629	0.758	0.500 (0.00)	0.835 (0.09)	**0.945** (0.02)
Grun	**0.993**	0.500	0.523	0.968	0.663 (0.00)	0.783 (0.02)	0.673	0.750	0.991	**0.994**	0.691 (0.00)	0.694 (0.01)	0.724 (0.01)
Li	**0.985**	0.919	0.804	0.827	0.896 (0.02)	0.871 (0.05)	0.909	0.852	**0.980**	0.966	0.955 (0.00)	0.970 (0.01)	0.797 (0.04)
Patel	0.653	**0.995**	**0.958**	0.939	0.927 (0.01)	0.859 (0.05)	0.940	0.569	0.788	0.942	0.945 (0.00)	0.954 (0.01)	0.875 (0.05)
Rank	7.125 (4.3)	5.750 (4.2)	5.625 (4.3)	7.250 (3.4)	9.750 (1.5)	10.750 (3.0)	5.625 (2.5)	8.875 (3.1)	5.750 (3.0)	**4.250** (3.2)	7.750 (2.6)	**4.375** (3.1)	8.250 (4.4)
Regret	0.171 (0.2)	0.150 (0.2)	0.146 (0.2)	0.156 (0.1)	0.222 (0.1)	0.363 (0.2)	0.123 (0.1)	0.249 (0.1)	0.114 (0.1)	**0.073** (0.1)	0.293 (0.2)	**0.103** (0.1)	0.149 (0.1)

Table 4: ARI comparison of clustering methods across 8 scRNA-seq datasets.

Dataset	Seurat	SC3	scAMF	DESC	UMAP	DeepCluster	IFPCA	IFVAE	i-IF-PCA	i-IF-Lap	i-IF-Lap+UMAP	i-IF-Lap+DeepCluster	i-IF-Lap+VAE
Camp1	0.519	**0.763**	**0.801**	0.729	0.521 (0.04)	0.516 (0.02)	0.629	0.639	0.635	0.650	0.624 (0.08)	0.616 (0.00)	0.597 (0.00)
Camp2	0.425	**0.594**	0.484	0.483	0.406 (0.01)	0.375 (0.03)	0.490	0.464	0.522	**0.524**	0.399 (0.00)	0.505 (0.01)	0.413 (0.04)
Darmanis	**0.719**	0.700	0.667	0.526	0.535 (0.05)	0.458 (0.01)	0.703	0.428	0.674	0.694	0.591 (0.03)	**0.703** (0.06)	0.558 (0.06)
Deng	0.427	0.541	0.561	0.426	0.440 (0.09)	0.393 (0.13)	0.848	0.431	0.810	**0.876**	0.567 (0.01)	**0.867** (0.01)	0.835 (0.10)
Goolam	0.544	0.687	**0.914**	0.543	0.423 (0.00)	0.766 (0.01)	0.537	0.205	0.582	0.687	0.427 (0.00)	0.801 (0.13)	**0.976** (0.01)
Grun	**0.969**	-0.060	-0.074	0.928	0.093 (0.00)	0.137 (0.07)	-0.096	0.244	0.955	**0.971**	0.145 (0.00)	0.150 (0.01)	0.198 (0.01)
Li	**0.971**	0.934	0.779	0.782	0.885 (0.02)	0.632 (0.13)	0.880	0.782	**0.985**	0.943	0.936 (0.00)	0.948 (0.01)	0.790 (0.04)
Patel	0.577	**0.989**	**0.905**	0.383	0.836 (0.01)	0.729 (0.09)	0.853	0.383	0.697	0.871	0.874 (0.00)	0.898 (0.02)	0.784 (0.04)
Rank	7.250 (4.7)	4.750 (3.5)	6.125 (4.4)	8.500 (3.6)	9.750 (1.9)	10.875 (3.3)	7.375 (3.2)	8.125 (3.8)	5.375 (3.1)	**3.375** (1.8)	7.500 (2.5)	**4.125** (2.5)	7.250 (3.3)
Regret	0.220 (0.2)	0.221 (0.1)	0.234 (0.3)	0.256 (0.2)	0.347 (0.2)	0.363 (0.2)	0.262 (0.3)	0.383 (0.3)	0.131 (0.1)	**0.087** (0.1)	0.271 (0.2)	**0.178** (0.3)	0.220 (0.2)

yielded $p = 3.6 \times 10^{-5}$, indicating a statistically significant difference in overall performance among the evaluated methods.

To further assess pairwise superiority, we conducted a Wilcoxon signed-rank test with Holm correction against the alternative hypothesis that i-IF-Lap achieves better clustering. Table 5 summarizes the results for methods evaluated across all 18 datasets. Our i-IF-Lap method significantly outperforms baselines such as UMAP, DeepCluster, and IFVAE, and shows statistically significant improvements over its VAE and UMAP pipeline variants.

Table 5: Summary of p-values from the Wilcoxon signed-rank test with Holm correction (Alternative hypothesis: i-IF-Lap works better). The methods compared are those evaluated across all 18 datasets.

Method	p-value	Method	p-value
UMAP	0.000	i-IF-Lap+UMAP	0.045
DeepCluster	0.011	i-IF-PCA	0.074
IFVAE	0.011	IFPCA	0.163
i-IF-Lap+VAE	0.045	i-IF-Lap+DeepCluster	0.290

5 SYNTHETIC DATASET

We conduct simulation studies to evaluate i-IF-Learn under controlled settings. We first compare the feature selection and clustering performance of i-IF-Learn methods under linear and non-linear settings, and then discuss the effects of the initial label and the constant c in reliability weightage.

Linear setting. Let $X_i \sim N(\ell_i \mu, \Sigma)$, where $\ell_i \in \{-1, 1\}$ with equal probability. The mean vector $\mu \in \mathbb{R}^p$ has $\mu_j = 0$ for $j \notin I$ and $\mu_j \neq 0$ for $j \in I$. Let $I = I_s \cup I_w$, where $\mu_j \sim \frac{1}{2}N(\tau_s, 0.01^2) + \frac{1}{2}N(-\tau_s, 0.01^2)$ for $j \in I_s$ and $\mu_j \sim \frac{1}{2}N(\tau_w, 0.01^2) + \frac{1}{2}N(-\tau_w, 0.01^2)$ for $j \in I_w$. Hence, I_s is the set of relatively strong signals and I_w is the set of weak signals. The covariance matrix Σ is a diagonal matrix with diagonals σ_j^2, where $\sigma_j \sim Unif(1, 3)$.

We set $n = 500$, $p = 5000$, $|I_s| = 4$ relatively strong signals and $|I_w| = 100$ weak signals. Let $\tau_s = 1.1$ and $\tau_w \in \{0.1, 0.15, 0.2, \cdots, 1.0\}$. A larger τ_w indicates a stronger signal-to-noise ratio.

Methods. We consider 1) i-IF-Lap and i-IF-PCA; 2) i-IF-Lap+DeepCluster/UMAP/VAE, where DeepCluster, UMAP, and VAE are applied to the influential features $\hat{I}$ from i-IF-Lap, respectively; 3) KMeans, SpecGEM, DeepCluster, DEC, UMAP, IFPCA and IFVAE as benchmark methods. For feature selection accuracy, we compare i-IF-Lap, i-IF-PCA, and IFPCA. For all methods, the input is the data points X_i's, without labels or influential feature information. The compute resource is in Appendix D.1. Implementation details and hyperparameter selection for all algorithms are in Appendix C.

Results. The accuracy rates versus τ_w over 100 repetitions in the right panel of Figure 3. As τ_w increases, all methods have a better accuracy from 0.5 to ~ 1. Our i-IF-Learn algorithms perform the best, especially when $0.4 \leq \tau_w \leq 0.8$. Methods using i-IF-Lap as preprocessing step also enjoy an outstanding performance, due to the reliable recovery of influential features.

To investigate the estimate of I, we summarize the False Discovery Rate (FDR) over 100 repetitions in the left panel of Figure 3. As τ_w increases, FDR for i-IF-Learn drops sharply, while IFPCA remains high across all settings. This highlights the benefit of iterative

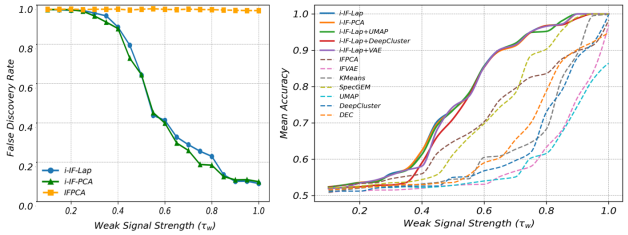

Figure 3: Left: FDR of feature selection step versus signal strength for signals in I_w. Right: the clustering accuracy versus signal strength for signals in I_w.

refinement: i-IF-Learn is able to recover weak signals with high precision. More figures about I can be found in Appendix D.3.

Nonlinear setting. Sample the underlying data points from a 2D manifold, and then project them to p-dimensional space. The observed data points are $x_i \in \mathbb{R}^p$. Let $n = 500$ and $K = 2$. We consider two experiments:

- **p-Sweep.** Let 20 features be strong signals with strength 1.0 and 60 features be weak signals with strength 0.2, while the remaining features are irrelevant. Vary p from 1500 to 6000.
- **μ-Power Sweep.** Let $p = 4000$, with 80 influential features. Each influential feature j has a signal strength at μ_j^a, where $\mu_j \sim_{i.i.d} U(0.2, 1)$ and the power $a \in \{1/4, 1/3, 1/2, 1, 2, 3, 4\}$.

Methods. We compare **i-IF-Lap**, **i-IF-PCA**, and **IFPCA**. Figure 4 shows that i-IF-Lap outperforms all other methods in both settings. It illustrates the power of non-linear embedding. Furthermore, the i-IF-PCA method outperforms IFPCA, indicating the power of iteration.

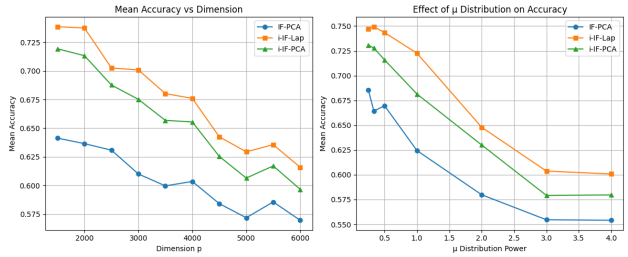

Figure 4: Left: clustering accuracy versus dimension p in the p-Sweep experiment. Right: clustering accuracy versus the power a in the μ-Sweep experiment.

Robustness. We examine the robustness of i-IF-Learn with respect to the choice of the constant c in Eq. (4) and the initialization scheme. Table 6 reports clustering accuracies for $c \in \{0.4, 0.5, 0.6\}$ under both linear and nonlinear settings. For each c, the top row corresponds to IFPCA initialization, while the bottom row corresponds to a random initialization. Results comparing different embedding methods (e.g., UMAP, Autoencoder, PCA, Laplacian Eigenmaps) can be found in Appendix 6. The results show that i-IF-Learn is stable across different constants c. Furthermore, even with a random initialization, our i-IF-Learn framework still enjoys some clustering improvements.

c	Init $\ell^{(0)}$	Linear		Non-linear	
		i-IF-Lap	i-IF-PCA	i-IF-Lap	i-IF-PCA
0.4	IFPCA	0.733 (0.22)	0.731 (0.22)	0.715 (0.08)	0.674 (0.11)
	Random	0.606 (0.17)	0.611 (0.17)	0.618 (0.11)	0.610 (0.11)
0.5	IFPCA	0.713 (0.22)	0.712 (0.22)	0.701 (0.09)	0.674 (0.10)
	Random	0.576 (0.12)	0.573 (0.12)	0.627 (0.11)	0.611 (0.09)
0.6	IFPCA	0.742 (0.22)	0.738 (0.22)	0.702 (0.09)	0.672 (0.12)
	Random	0.585 (0.14)	0.580 (0.14)	0.644 (0.11)	0.619 (0.11)

Table 6: Accuracy (mean (std)) of i-IF-Lap and i-IF-PCA across different constants and initializations.

6 DISCUSSION

We introduce i-IF-Learn, an iterative framework for high-dimensional clustering that integrates feature selection with low-dimensional embedding. By adaptively using supervised pseudo-labels and unsupervised statistics, our novel screening metric enables robust feature selection even when early clustering assignments are noisy. Unlike static pipelines, i-IF-Learn iteratively refines both feature sets and labels to effectively amplify weak signals. Beyond assigning cluster labels, i-IF-Learn outputs an interpretable set of influential features.

As an exploration of the framework's flexibility, we replaced the IF step with a supervised Lasso penalty; however, this yielded sub-optimal performance (see Appendix G).

While demonstrating strong empirical results, i-IF-Learn presents several avenues for future research. First, the current marginal screening step evaluates features individually, potentially ignoring pairwise or block interactions. Future work could incorporate a supervised recovery step (e.g., using CIFE (Lin and Tang, 2006), or JMI (Yang and Moody, 1999)) on the full feature set, guided by the generated pseudo-labels, to capture joint effects and eliminate redundancies. Second, i-IF-Learn currently identifies a single global set of influential features. Developing high-resolution methods to detect distinct, cluster-specific feature subsets is a highly relevant next step. More HDLSS datasets can be found in the publicly available repository provided by Li et al. (2017)(`https://jundongl.github.io/scikit-feature/datasets`).

Acknowledgements

This research was supported by the Singapore Ministry of Education Academic Research Fund Tier 1 under Grant A-8001451-00-00. C. Ma gratefully acknowledges the financial support from the Southern University of Science and Technology (SUSTech) during the exchange program at the National University of Singapore (NUS), and the travel support provided by the Wallinska resestipendiet and the Swedish Association for Medical Statistics (FMS) travel scholarship. The authors also thank the anonymous reviewers for their valuable comments and suggestions, which led to the addition of new experiments in this work.

References

Pierre Baldi. Autoencoders, unsupervised learning and deep architectures. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, page 37–50. JMLR.org, 2011.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

Sabri Boutemedjet, Djemel Ziou, and Nizar Bouguila. Unsupervised feature selection for accurate recommendation of high-dimensional image data. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/073b00ab99487b74b63c9a6d2b962ddc-Paper.pdf.

Christos Boutsidis, Petros Drineas, and Michael W Mahoney. Unsupervised feature selection for the k-means clustering problem. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/c51ce410c124a10e0db5e4b97fc2af39-Paper.pdf.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Dieyi Chen, Jiashun Jin, and Zheng Tracy Ke. Subject clustering by IF-PCA and several recent methods. *Frontiers in Genetics*, 14:1166404, 2023.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

David Donoho and Jiashun Jin. Higher criticism for detecting sparse heterogeneous mixtures. *The Annals of Statistics*, pages 962–994, 2004.

David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, 1(2000):32, 2000.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*, chapter 5. Wiley-Interscience, 2 edition, 2001. ISBN 0-471-05669-3.

Bradley Efron. Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99(465):96–104, 2004.

Ran Eisenberg, Jonathan Svirsky, and Ofir Lindenbaum. Coper: Correlation-based permutations for multi-view clustering. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 46154–46179, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/720719dbe00dcd5210cd5ec3a7399476\-Paper-Conference.pdf.

Miriam R Elman, Jessica Minnier, Xiaohui Chang, and Dongseok Choi. Noise accumulation in high dimensional classification and total signal index. *Journal of Machine Learning Research*, 21(36):1–23, 2020.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

Wenkai Han, Yuqi Cheng, Jiayang Chen, Huawen Zhong, Zhihang Hu, Siyuan Chen, Licheng Zong, Liang Hong, Ting-Fung Chan, Irwin King, Xin Gao, and Yu Li. Self-supervised contrastive learning for integrative single cell rna-seq data analysis. *Briefings in Bioinformatics*, 23(5):bbac377, 09 2022. ISSN 1477-4054. doi: 10.1093/bib/bbac377. URL https://doi.org/10.1093/bib/bbac377.

John A Hartigan and Manchek A Wong. Algorithm AS 136: A K-means clustering algorithm. *Journal of the Royal Statistical Society. series c (Applied Statistics)*, 28(1):100–108, 1979.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985. ISSN 1432-1343. doi: 10.1007/BF01908075. URL https://doi.org/10.1007/BF01908075.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 1965–1972. AAAI Press, 2017. ISBN 9780999241103.

Jiashun Jin and Wanjie Wang. Influential features PCA for high dimensional clustering. *The Annals of Statistics*, 44(6):2323–2359, 2016.

Jiashun Jin, Zheng Tracy Ke, and Wanjie Wang. Phase transitions for high dimensional clustering and related problems. *The Annals of Statistics*, 45(5):2151–2189, 2017.

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

Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, et al. SC3: consensus clustering of single-cell RNA-seq data. *Nature Methods*, 14(5):483–486, 2017.

Ann B Lee, Diana Luca, and Kathryn Roeder. A spectral graph approach to discovering genetic ancestry. *The Annals of Statistics*, 4(1):179, 2010.

Changhee Lee, Fergus Imrie, and Mihaela van der Schaar. Self-supervision enhanced feature selection with correlated gates. In *International Conference on Learning Representations*, 2022.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017.

Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature Communications*, 11(1):2338, 2020a.

Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P. Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature Communications*, 11(1):2338, 2020b. doi: 10.1038/s41467-020-15851-3. URL https://doi.org/10.1038/s41467-020-15851-3.

Dahua Lin and Xiaoou Tang. Conditional infomax learning: An integrated framework for feature extraction and fusion. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 68–82, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33833-8.

Ofir Lindenbaum, Uri Shaham, Erez Peterfreund, Jonathan Svirsky, Nicolas Casey, and Yuval Kluger. Differentiable unsupervised feature selection based on a gated laplacian. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1530–1542. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0bc10d8a74dbafbf242e30433e83aa56-Paper.pdf.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2020.

Yushan Qiu, Lingfei Yang, Hao Jiang, and Quan Zou. sctpc: a novel semisupervised deep clustering model for scrna-seq data. *Bioinformatics*, 40(5):btae293, 04 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae293. URL https://doi.org/10.1093/bioinformatics/btae293.

Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, 2015.

Nikolai V Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2 (2):3–14, 1939.

Jonathan Svirsky and Ofir Lindenbaum. Interpretable deep clustering for tabular data. In *International Conference on Machine Learning*. PMLR, 2024.

Xin T Tong, Wanjie Wang, and Yuguan Wang. Uniform error bound for PCA matrix denoising. *Bernoulli*, 31(3):2251–2275, 2025.

Nakul Upadhya and Eldan Cohen. Neurcam: Interpretable neural clustering via additive models, 2024. URL https://arxiv.org/abs/2408.13361.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Jianqiao Wang, Wanjie Wang, and Hongzhe Li. Sparse block signal detection and identification for shared cross-trait association analysis. *The Annals of Applied Statistics*, 16(2):866–886, 2022.

Daniela M Witten and Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

Xinxing Wu and Qiang Cheng. Algorithmic stability and generalization of an unsupervised feature

selection algorithm. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19860–19875. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a546203962b88771bb06faf8d6ec065e-Paper.pdf.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487. PMLR, 2016.

Howard Yang and John Moody. Data visualization and feature selection: New algorithms for nongaussian data. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/8c01a75941549a705cf7275e41b21f0d-Paper.pdf.

Zhigang Yao, Bingjie Li, Yukun Lu, and Shing-Tung Yau. Single-cell analysis via manifold fitting: A framework for RNA clustering and beyond. *Proceedings of the National Academy of Sciences*, 121(37): e2400002121, 2024.

Zhiyue Zhang, Kenneth Lange, and Jason Xu. Simple and scalable sparse k-means clustering via feature ranking. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10148–10160. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/735ddec196a9ca5745c05bec0eaa4bf9-Paper.pdf.

Zheng Zhao, Lei Wang, and Xiaofeng Du. Exploring feature selection with limited labels: A comprehensive survey of semi-supervised and unsupervised approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1769–1787, 2019.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] We provide the mathematical setting and assumptions in Section 3, and describe the algorithms in Section 2 and provide pseudocodes for each algorithms in Appendix B.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] The computational complexity is analyzed in Section 2.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] An anonymized implementation with dependencies is included in the supplemental file `i-IF-Lear.zip` and `NumericalStudy.zip`.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] All assumptions are clearly stated in Section 3.
   (b) Complete proofs of all theoretical results. [Yes] Full proofs are provided in Appendix A.
   (c) Clear explanations of any assumptions. [Yes] Explanations are given alongside the assumptions in Section 3.
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] Code is included in the supplemental material, and datasets are public with download links provided in Section 4 and more details about in D.2.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Detailed implementation settings and hyperparameters are listed in Appendix C.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] We report standard deviations for all results, and omit them only when the variance is negligible (less than 0.0001).
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] As described in Appendix D.1, our experiments are lightweight and reproducible on a standard PC without specialized hardware.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] All datasets and existing codes are properly cited (see Section 4 and Appendix D.2).
   (b) The license information of the assets, if applicable. [Yes] License information for the datasets is listed in Appendix.

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes] We provide an anonymized implementation of our proposed method in `i-IF-Learn.zip`.

(d) Information about consent from data providers/curators. [Not Applicable] All datasets used are publicly available.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] The datasets do not contain sensitive or personally identifiable content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable] Our work does not involve crowdsourcing or human subjects.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable] Our work does not involve crowdsourcing or human subjects.

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable] Our work does not involve crowdsourcing or human subjects.

i-IF-Learn: Iterative Feature Selection and Unsupervised Learning for High-Dimensional Complex Data: Supplement Material

A TTECHNICAL DETAILS AND PROOFS

In section A.1, we explain the details of the weight selection. With the weight selection, we prove the main theorems and corresponding lemmas in the following subsections.

A.1 Dynamic Weigh

To decide the reliability constant $w^{(t)}$, we want to test our trust in $P_{F,j}^{(t)}, j \in I^{(t-1)}$. In other words, we wonder whether these statistics follow a null distribution or not.

Let $\pi_I^{(t-1)} = \{P_{F,j}^{(t)}, j \in I^{(t-1)}\}$ be the set of p-values from corrected marginal F-statistics, restricted on the features in $I^{(t-1)}$. An unreliable feature set $I^{(t-1)}$ will have uniformly distributed p-values $\pi_I^{(t-1)}$. Therefore, it is the hypothesis testing problem that $\pi_I^{(t-1)} \sim Unif(0,1)$.

We construct a test for the hypothesis

$$H_0 : \pi_{(j)}^{(t-1)} \sim Unif(0,1) \quad vs \quad H_1 : \pi_{(j)}^{(t-1)} \sim (1-\epsilon)Unif(0,1) + \epsilon G,$$

where G is some other distribution that focuses on small p-values.

We conduct the Higher Criticism statistic Donoho and Jin (2004) for this test. Denote $s^{(t-1)} = |I^{(t-1)}|$ and $\pi_{(j)}^{(t-1)}$ as the j-th smallest value in $\pi_I^{(t-1)}$. The p-value is

$$p_1^{(t)} = 1 - \exp(-e^{c-bT}), \text{ where } T = \max_{1 \le j \le 2s^{(t-1)}/3} \sqrt{s^{(t-1)}} \frac{j/s^{(t-1)} - \pi_{(j)}^{(t-1)}}{\sqrt{\pi_{(j)}^{(t-1)}(1-\pi_{(j)}^{(t-1)})}}, \tag{11}$$

and $b = \sqrt{2\log(\log(s^{(t-1)}))}$, $c = 2\log(\log(s^{(t-1)})) + \log(\log(\log(s^{(t-1)})))/2 - \log(4\pi)/2$. Here, the p-value $p_1^{(t)} \in (0,1)$, and a smaller $p_1^{(t)}$ indicates a larger possibility that $I^{(t-1)}$ is informative.

A.2 Proof of Theorem 3.1

Given a label $\hat{\ell}$, we use it as a pseudo-label to select the influential features for the next step. The selection is based on our new score $S_j^{(t)} = S_j(\hat{\ell})$, defined in (3), where the weight is defined in (4). Therefore, to show that our IF step is powerful, we need the score $S_j(\hat{\ell})$ to efficiently evaluate the dependency between feature x_j and $\hat{\ell}$, and then the data-driven threshold HCT to select a proper threshold.

The following lemma explains the power of our statistic: when the initial label delivers some information, then the weight will be close to 1, so that the $S_j(\hat{\ell})$ highly depends on the F-statistics. Further, even with noisy labels, our statistic $S_j(\hat{\ell})$ clearly separates I and I^c. The proof of the lemma can be found in Section A.3.

Lemma A.1. *Fix a constant $q > 4$. Denote $w_{ij} = E[\Sigma^{-1/2}X_{ij}]$ as the expectation for data point i on feature j, and the overall mean $\bar{w}_j = \frac{1}{n}\sum_{i=1}^n w_{ij}$. Denote $n_k(\hat{\ell}) = \sum_{i=1}^n 1\{\hat{\ell}_i = k\}$, $k \in [K]$. Suppose for the influential features in I, the community label satisfies that for a constant $c_0 > 0$,*

$$\min_{j \in I} U_j \ge c_0(\log p)^2, \quad \textit{where } U_j := \sum\nolimits_{k \in [K]} \frac{1}{n_k(\hat{\ell})} \sum\nolimits_{\hat{\ell}_i = k} (w_{ij} - \bar{w}_j)^2.$$

Then we have $P(w \ge 1 - p^{-q/2}) \ge 1 - p^{-q/2}$, and further

- *If* $j \in I$, $P(S_j(\hat{\ell}) > p^{-q}) \le p^{-q}$.
- *If* $j \in I^c$, *for any* $u \in (0,1)$, $P(S_j(\hat{\ell}) < u) \le u - \exp(-p)$.

For notation simplicity, we write $S_j = S_j(\hat{\ell})$ when there is no misunderstanding. The proof consists of three steps. We first show that S_j for $j \in I$ and $j \notin I$ has a clear division with high probability. Then we show that Higher Criticism Threshold (HCT) almost achieves the optimal division, in the sense that $HC(j)$ for all $j \in I^c$ is smaller than the selected HCT and no more than $C \log p$ feature in I have lower $HC(j)$ than HCT.

Step 1: Clear division with high probability. We define a good event that the statistics from $j \in I$ and $j \notin I$ are clearly divided, that

$$B = \{\max_{j \notin I^c} S_j < 1 + q \log p < \min_{j \in I} S_j\}.$$

By Lemma A.1, the probability of B bounded by the union bound

$$1 - P(B) \quad \le \quad |S| p^{-q} + p p^{-q} = o(1).$$

Step 2: Under B, all indices from I will be selected by HCT. By the algorithm, $\hat{I}$ is achieved by selecting all the features with p-values smaller than the threshold τ. In other words, if we order the features in the way that the p-value of S_j is decreasing, then the first $|\hat{I}|$ features are selected.

According to the definition of B in Step 1, features from I always have smaller p-values than those in I^c. Therefore, the first $|I|$ features are from I and the following features are from I^c. As long as we set the threshold as the p-value of the $|I|$-th smallest one, then we exactly recover I. Note that we select $|\hat{I}|$ according to where the HC scores $HC(j)$ achieves maximum. If $HC(j)$ achieves maximum around $j = |I|$, then the cutoff is correct. It suffices to show that for any $j < |I|$,

$$HC(j) \le HC(|I|). \tag{12}$$

Under B, $\max_{j \in I} S_j \le p^{-q}$. Introduce it into $HC(|I|)$, we have

$$HC(|I|) \ge \frac{|I|/p - p^{-q}}{\sqrt{|I|/p(1-|I|/p)}} \ge \frac{|I|/p - p^{-q}}{\sqrt{|I|/p(1-|I|/p)}}. \tag{13}$$

Meanwhile, we have $HC(j) \le \frac{\sqrt{(|I|-1)/p}}{\sqrt{1-(|I|-1)/p}}$ for all $j \le |I| - 1$.

Plug these upper bounds and (13) into (12). We have

$$\begin{aligned}
[HC(j)]^2 \le [HC(|I|)]^2 \Leftrightarrow \\
\frac{(|I|-1)/p}{1-(|I|-1)/p} \le \frac{(|I|/p - p^{-q})^2}{|I|/p(1-|I|/p)} \Leftrightarrow \\
(|I|-1)|I|(p-|I|) \le (|I| - p^{-q})^2 (p - |I| + 1).
\end{aligned}$$

Rearrange the terms and it suffices to show (12) if we can show that

$$|I|(p-|I|)(1-2p^{-2q}) + |I|(|I| - 2p^{-q}) + p^{-2q}(p - |I| + 1) \ge 0.$$

When p is sufficiently large, it holds. So (12) holds for sufficiently large p.

Step 3: Under B, only $C \log^2 p$ features from I^c will be selected by HCT. According to previous analysis, it suffices to show $HC(|I| + k) < HC(|I|)$ when $k > C \log^2 p$.

Consider a sequence of threshold $v_k = k/|I^c|$ for $k > C_1 \log^2 p$. Note that $P(S_j \le u) \le u - \exp(-p)$ for all $j \in I^c$. By the Bernstein inequality with $\delta_k = 4\sqrt{v_k \log p / p}$, when $C_1 > 2(p/|I^c|)^2$ and $|I^c|/p > 1/2$, we have

$$\begin{aligned}
P\big(\big|\frac{1}{|I^c|} \sum_{j \in I^c} 1_{S_j < v_k} - v_k\big| > \delta_k\big) \quad &\le \quad \exp\left(-\frac{|I^c|\delta_k^2}{2(v_k(1-v_k) + \delta_k/3)}\right) \\
&\le \quad \exp\left(-\frac{|I^c|\delta_k^2}{4v_k(1-v_k)}\right) + e^{-3|I^c|\delta_k/4} \\
&\le \quad 2p^{-2}.
\end{aligned} \tag{14}$$

Naturally, the union bound follows, which is

$$P\big(|\frac{1}{|I^c|}\sum_{j\in I^c} 1_{S_j<v_k} - v_k| > \delta_k, \text{ for any } k \geq C_1\log^2 p\big) \leq \frac{1}{p}.$$

Hence, the complementary event happens with probability $1 - O(p^{-1})$. Define it as

$$C := \left\{ |\frac{1}{|I^c|}\sum_{j\in I^c} 1_{S_j<v_k} - v_k| < \delta_k, \text{ for any } k \geq C_1\log^2 p \right\}.$$

Denote the ordered S_i as $S_{(1)} \leq S_{(2)} \leq \ldots \leq S_{(p)}$. Under the event B, $S_{(|I|+k)} = S^0_{(k)}$ where $S^0_{(\cdot)}$ are the ordered p-values of $i \in I^c$. Now we want to derive a lower bound of $S_{(k)}$.

Under C, recall that $v_j = j/|I^c|$ and $\delta_j = 4\sqrt{v_j \log p/p}$, when $j > C_1\log^2 p$,

$$\sum_{i\in I^c} 1\{S_i < v_j\} < |I^c|(v_j + \delta_j) < j + 4\sqrt{j\log p}.$$

It means $S^0_{(j+4\sqrt{j\log p})} \geq v_j$. Now let k be the smallest integer so that $k \geq j + 4\sqrt{j\log p}$, then it follows $j \geq k - 1 - 4\sqrt{k\log p}$. Hence, under $C \cap B$, for $k \geq 3C_1\log^2 p$,

$$S_{(|I|+j+4\sqrt{j\log p})} \geq v_j \Longrightarrow S_{(|I|+k)} \geq (k-1)/p - 4\sqrt{k\log p}/p,$$

This further leads to

$$HC(|I|+k) = \frac{\frac{|I|+k}{p} - S_{(|I|+k)}}{\sqrt{\frac{(|I|+k)}{p}(1-\frac{|I|+k}{p})}} \leq \frac{\frac{|I|+k}{p} - \frac{k-1}{p} + \frac{4\sqrt{k\log p}}{p}}{\sqrt{\frac{(|I|+k)}{p}(1-\frac{|I|+k}{p})}}.$$

Combine it with (13). To conclude our claim, we need to show that if $0.25p > k \geq 3C_1\log^2 p$, there is

$$\frac{(|I|-p^{-c})^2}{|I|(p-|I|)} > \frac{(|I|+1+4\sqrt{k\log p})^2}{(|I|+k)(p-|I|-k)}.$$

This can be shown if

$$(|I|-p^{-c})^2(|I|+k)(p-|I|-k) > (|I|+1+4\sqrt{k\log p})^2(|I|+k)(p-|I|-k).$$

When $|I| \gg \log p$, it can further be simplified as

$$p|I|^3 + kp|I|^2 > p|I|^3 + 8p\sqrt{k\log p}|I|^2 + \text{lower order terms}.$$

It holds when $k \geq 3C_1\log^2 p$ and p large enough. In other words, $HC(|I|+k) < HC(|I|)$ for $k > C\log^2 p$ where $C = 3C_1$. This gives our second bound.

Combining the results, Theorem 3.1 is proved.

A.3 Proof of Lemma A.1

In this section, we prove Lemma A.1used in the previous section. First we discuss the required condition that $\min_{j\in I} U_j \geq c_0(\log p)^2$, which is about the estimated labels $\hat{\ell}$ and the influential features.

Consider the $K = 2$ case where both classes have size $n/2$. For feature $j \in I$, since the overall sum is $m = 0$, for a constant signal strength κ, there is $m_1 = \kappa$ and $m_2 = -\kappa$. Now, consider the estimated label $\hat{\ell}$. Suppose it classifies $r_{11}n/2$ with true label 1 and $r_{21}n/2$ with true label 2 as Class 1. Then we have

$$\begin{aligned}\sum_{k\in[K]}\frac{1}{n_k}(\sum_{i\in[n_k]} m_{i,k})^2 =& \frac{1}{(r_{11}+r_{21})n/2}(\kappa r_{11}n/2 - \kappa r_{21}n/2)^2 \\ &+ \frac{1}{(2-r_{11}-r_{21})n/2}(\kappa(1-r_{11})n/2 - \kappa(1-r_{21})n/2)^2 \\ =& \frac{n\kappa^2(r_{11}-r_{21})^2}{(r_{11}+r_{21})(2-r_{11}-r_{21})}.\end{aligned}$$

Hence, as long as $r_{11} \neq r_{21}$, i.e. there is a constant difference between the proportion the two classes that $\hat{\ell}$ classifies into class 1, then the whole term is at an order of $Cn\kappa^2$. The condition follows that

$$Cn\kappa^2 \geq (\log p)^2 \Leftrightarrow \kappa^2 \geq \frac{(\log p)^2}{n}.$$

It means a constant error rate is accepted in the initial label.

Now we come to the proof. We will suppress the appearance of feature index j in the subscript for notational simplicity. We also remove $\hat{\ell}$. Denote $J_k = \{i \mid \hat{\ell} = k\}$, $k \in [K]$. Let $n_k = |J_k|$ for $k \in [K]$.

When $j \notin I^c$, the nominator of the F-statistics is given by

$$N_0 = \frac{1}{K-1} \sum_{k\in[K]} n_k (\frac{1}{n_k} \sum_{i\in J_k} \xi_i - \frac{1}{n} \sum_{i\in[n]} \xi_i)^2 \sim \frac{\chi^2_{K-1}}{K-1},$$

where ξ_i are some $N(0,1)$ distributed random variables. While the denominator

$$D_0 = \frac{1}{n-K} \sum_{k\in[K]} \sum_{i\in J_k} (\xi_i - \frac{1}{n_k} \sum_{j\in J_k} \xi_j)^2 \sim \frac{\chi^2_{n-K}}{n-K}.$$

Using concentration inequality, we find that there is a constant C_0 so that

$$P(N_0 > 1 + C_0 q \log p) \leq p^{-q}, \quad P(D_0 < 1 - C_0\sqrt{q \log p/n}) \leq p^{-q}.$$

So

$$P(N_0/D_0 > 1 + 2C_0 q \log p) \leq 2p^{-q}.$$

And if $j \in I$, denote $\Delta m_i = m_i - \bar{m}, \Delta\xi_i = \xi_i - \bar{\xi}$

$$N_1 = \frac{1}{K-1} \sum_{k\in[K]} n_k (\frac{1}{n_k} \sum_{i\in J_k} (\Delta\xi_i + \Delta m_i))^2 \sim \frac{\chi^2_{K-1}}{K-1} + Q + U_j$$

Where

$$Q = \frac{1}{K-1} \sum_{k\in[K]} \frac{1}{n_k} \left(\sum_{i\in J_k} \Delta\xi_i + \Delta m_i \right)^2$$

Using concentration inequality of Gaussian, $P(|Q| + \frac{\chi^2_{K-1}}{K-1} > q \log p) \leq p^{-q}$. So $P(N_1 < U_j - q \log p) \leq p^{-q}$.

Meanwhile, if $l(i) = k$, we denote $\nabla m_i = m_i - \frac{1}{n_k} \sum_{j\in J_k} m_j, \nabla\xi_i = \xi_i - \frac{1}{n_k} \sum_{j\in J_k} \xi_j$

$$D_1 = \frac{1}{n-K} \sum_{i\in[n]} (\nabla m_i + \nabla\xi_i)^2 \sim \frac{\chi^2_{n-K} + 2\sum_{i\in[n]} \nabla m_i \nabla\xi_i + (\nabla m_i)^2}{n-K}$$

Using Gaussian concentration, there is a constant C, so that $P(D_1 > C) \leq p^{-q}$. In combine, we find $P(D_1/N_1 \leq q \log p) \leq P(D_1/N_1 < U_j/C) \leq p^{-q}$. Therefore,

$$P(P_{F,j} \geq p^{-q}) \leq P(D_1/N_1 \leq q \log p) \leq p^{-q}.$$

The next step would be considering the reliability constant. We note that

$$T \geq \sqrt{|I|} \frac{1/|I| - P_{F,(1)}}{\sqrt{P_{F,(1)}}} \geq \frac{p^{-1} - p^{-q}}{p^{-q/2}} \geq p^{q/2-1} \geq 2p$$

Then

$$p_1 \geq 1 - \exp(-\exp(\log p - p)) \geq \exp(-p).$$

And we get $w = 1 - \frac{p_1}{p_1 + 0.6} \geq 1 - \exp(-p)$.

Finally, for $j \in I$ we have

$$P(S_j > 2p^{-q}) \leq P(P_{F,j} > 2p^{-q} - w + 1) \leq p^{-q}.$$

For $j \in I^c$, we have

$$P(S_j < u) \leq P(P_{F,j} w < u) \leq u - \exp(-p).$$

The result is proved.

A.4 Proof of Theorem 3.2

With a correct selection of I, k-means on the low-dimensional embeddings will give a nice clustering result. The clustering error rate is evaluated by the Hamming error, which is the proportion of unmatched labels under the best scenario. In detail, for estimated label $\hat{\ell}$, let $\pi : [K] \to [K]$ be any permutation of $\{1, \cdots, K\}$, then

$$Err(\hat{\ell}, \ell) = \min_{\pi:[K]\to[K]} \sum_{i=1}^{n} 1\{\hat{\ell}_{\pi(i)} \neq \ell_i\}.$$

For notation simplicity, denote X as the post-selection data matrix $X^{(t)}$. Denote s_I be the number of informative features in $I^{(t)}$ and s be the total number.

The normalized data matrix can be written as $W = LM_I + E$, where E is the noise matrix, $L \in \{0,1\}^{n\times K}$ is the label matrix and $M_I \in \mathbb{R}^{s\times K}$ be the mean matrix on $I^{(t)}$ among all classes. Denote τ_I be the eigengap of M_I, which is no smaller than $\tau\sqrt{s_I}$.

According to random matrix theory Vershynin (2010), with high probability

$$\|E\| \leq 2(\sqrt{n} + \sqrt{s})$$

Let $\hat{U}$ denote the top K left singular vectors of X and U denote the top K left singular vectors of LM_I. By Davis-Kahan Theorem Davis and Kahan (1970), there exists an orthogonal matrix O, so that

$$\|\hat{U} - UO\| \leq \frac{\sqrt{n} + \sqrt{s}}{\text{eigengap}(LM_I)} \leq C\frac{\sqrt{n} + \sqrt{s}}{\sqrt{n}\tau_I} := \delta.$$

Next we examine the performance of k-means on $\hat{U}$. For any estimated label $\hat{\ell}$ and centers $\hat{u}$, define the within-cluster distance $L(\hat{\ell}, \hat{u}) = \sum_{i=1}^{n} \|z_i - \hat{u}_{\hat{\ell}(i)}\|^2$. The algorithm k-means is to find $\hat{\ell}$ that minimizes $L(\hat{\ell}, \hat{u})$.

Suppose the singular value decomposition $LM_I = U\Lambda V'$, then $U = LM_IV\Lambda^{-1}$. For two nodes i and j in the same group, there is $\ell_i = \ell_j$ and the i-th row and j-th row in L are the same. Therefore, the i-th row and j-th row of U are the same. This will be the basis of clustering.

Denote the i-th row of $\hat{U}$ as z_i. Denote the k-th row of $M_IV\Lambda^{-1}$ as u_k. For the true labels ℓ and centers u_k's, there is $L(\ell, u) = \sum_{i=1}^{n} \|z_i - u_{\ell(i)}\|^2$. Let $\hat{\ell}$ and $\hat{u}$ be the labels and centers identified by k-means. Hence, for centers $\hat{u}_k$ and labels $\hat{\ell}$,

$$L(\hat{\ell}, \hat{u}) \leq L(\ell, u), \tag{15}$$

For any community k, let the permutation $\pi(k) = \arg\min_{1\leq j\leq K} \|u_k - \hat{u}_j\|$. Hence, $\pi(k)$ is the community where the estimated center is closest to u_k. We want to control the distance between u_k and $\hat{u}_{\pi(k)}$. According to k-means, for community k, let n_k be the size of community k, then

$$\begin{aligned}
L(\hat{\ell}, \hat{u}) &\geq \sum_{\ell(i)=k} \|z_i - \hat{u}_{\hat{\ell}(i)}\|^2 \\
&\geq \sum_{\ell(i)=k} (-\|z_i - u_kO\|^2 + \frac{1}{2}\|u_kO - \hat{u}_{\hat{\ell}(i)}\|^2) \\
&\geq -\sum_{i=1}^{n} \|z_i - u_kO\|^2 + \frac{1}{2}\sum_{\ell(i)=k} \|u_kO - \hat{u}_{\hat{\ell}(i)}\|^2 \\
&\geq -L(\ell, u) + \frac{1}{2}n_k\|u_kO - \hat{u}_{\pi(k)}\|^2.
\end{aligned} \tag{16}$$

Combining (15) and (16), there is

$$\|u_k - \hat{u}_{\pi(k)}\|^2 \leq 4L(\ell, u)/n_k \leq 4L(\ell, u)/cn.$$

Recall that the centers $\{u_i, i = 1, \ldots, K\}$ are $1/\sqrt{n}$-distance apart. If $L(\ell, u) < c_0$ for a constant c_0 small enough, then each u_k is paired with a unique $\hat{u}_{\pi(k)}$ such that $\|\hat{u}_{\pi(k)} - u_k\| \leq 2\sqrt{L(\ell, u)/cn}$.

Furthermore, since the data points come from $\hat{U}$ and the centers are from U, where the distance is controlled by the Davis-Kahan Theorem. Therefore, we control the loss in the idean case, where

$$L(\ell, uO) = \sum_{i=1}^{n} \|z_i O' - u_{\ell(i)} OO'\|^2 \leq K\delta^2. \tag{17}$$

Finally, we consider the mis-classification rate. To simplify the notations, we assume $\pi(k) = k$ without loss of generality. Then the misclassified nodes are $S = \{i : \ell_i \neq \hat{\ell}_i\}$. The misclassification rate is $Err(\hat{\ell}, \ell) = |S|/n$.

$$\begin{aligned} L(\hat{\ell}, \hat{u}) = \sum_{i=1}^{n} \|z_i - \hat{u}_{\hat{\ell}_i}\|^2 &\geq -\sum_{i=1}^{n} \|z_i O' - u_{\ell_i}\|^2 + \frac{1}{2}\sum_{i=1}^{n} \|u_{\ell_i} - \hat{u}_{\hat{\ell}_i} O'\|^2 \\ &\geq -L(\ell, uO) + \frac{1}{2}\sum_{i\in S} \|u_{\ell_i} - \hat{u}_{\hat{\ell}_i} O'\|^2 \\ &\geq -L(\ell, uO) + \frac{1}{2}|S|/n. \end{aligned}$$

Combining it with $L(\hat{\ell}, \hat{u}) \leq L(\ell, uO)$ in (15) and $L(\ell, uO) \leq \delta^2$ in (17), then we have $Err(\hat{\ell}, \ell) = |S|/n \leq n\delta^2$. The theorem is proved.

B PSEUDO-CODE FOR ALGORITHMS

In this section, we present the pseudo-code for our algorithm and some other algorithms without existing packages. Here is the list of algorithms we have discussed:

- IFPCA, the initialization step and a comparison algorithm, in Algorithm 3
- i-IF-Learn, our algorithm, in Algorithm 4
- DeepCluster in Algorithm 5
- Deep Embedding Clustering (DEC) in Algorithm 6
- Uniform Manifold Approximation and Projection (UMAP) in Algorithm 7
- Variational Autoencoder (VAE) in Algorithm 8

In this section, we only present the pseudo-code. Hyper-parameter selections and implementation details can be found in Section C.

Algorithm 3 IFPCA Initialization Procedure

Require: Data matrix $X \in \mathbb{R}^{n\times p}$, number of clusters K
Ensure: Initial cluster labels $\ell^{(0)}$, initial influential feature set $I^{(0)}$, p-values of KS test $P_{KS,p}$

Normalized data matrix X, denoted it as W.
Step 1.1 Compute Kolmogorove-Smirnov scores
for $j = 1$ to p **do**
    $\psi_{n,j} \leftarrow \sqrt{n} \cdot \sup_t |F_{n,j}(t) - \Phi(t)|$, where $F_{n,j}(t)$ is the empirical cumulative density function of w_j and Φ is standard normal distribution.
end for
Normalize scores: $\psi^*_{n,j} \leftarrow \dfrac{\psi_{n,j} - \text{mean}(\psi_{n,\cdot})}{\text{std}(\psi_{n,\cdot})}$
Step 1.2: HCT and feature selection:
for $j = 1$ to p **do**
    $P_{KS,j} \leftarrow 1 - F_0(\psi^*_{n,j})$, where F_0 is the null distribution.
end for
Sort p-values: $P_{KS,1} \le P_{KS,2} \le \cdots \le P_{KS,p}$
for $j = 1$ to $p/2$ **where** $P_{KS,j} > \log(p)/p$ **do**
    $HC_{p,j} \leftarrow \dfrac{\sqrt{p}(j/p - \pi_{(j)})}{\sqrt{\max\{\sqrt{n}(j/p - \pi_{(j)}), 0\} + j/p}}$
end for
$\hat{j} \leftarrow \arg\max_j HC_{p,j}$, $t_p^{\text{HC}} \leftarrow \psi^*_{n,\hat{j}}$
$I^{(0)} \leftarrow \{1 \le j \le p \mid \psi^*_{n,j} > t_p^{\text{HC}}\}$
Step 1.3: PCA embedding and k-means clustering:
Apply PCA to post-selection data, retain top $K-1$ components. Denote it as U.
$\ell^{(0)} \leftarrow k\text{-means}(U, K)$

Algorithm 4 i-IF-Learn

Require: Data matrix $X \in \mathbb{R}^{n\times p}$, number of clusters K
Ensure: Predicted cluster labels ℓ, influential feature set I

Step 1: Initialization with IFPCA

Require: Data matrix $X \in \mathbb{R}^{n\times p}$, number of clusters K
Ensure: Initial cluster labels $\ell^{(0)}$, initial influential feature set $I^{(0)}$, p-values of KS test $P_{KS,p}$

Detail for this part, please see in Alg 3

Step 2: Iterative Loop

Require: Data matrix $X \in \mathbb{R}^{n\times p}$, number of clusters K, initial cluster labels $\ell^{(0)}$, initial influential feature set $I^{(0)}$, p-values of KS test $P_{KS,p}$
Ensure: Predicted cluster labels ℓ, influential feature set I

Sample F_{random} as F statistics under random selected features.
for $t = 1, 2, \ldots,$ max_iter **do**

Step 2.1: Compute F statistic

  for $j = 1$ to p **do**
    Computer $F^{(t)}(j)$ under $\ell^{(0)}$
    Normalize the F statistics with quantiles: $F^{(t)}_{\text{adj}}(j) = \frac{F(j)-Q^d_2}{Q^d_3-Q^d_1} * (Q^t_3 - Q^t_1) + Q^t_2$, where Q^d_q and Q^t_q are empirical and theoretical q-th quantiles of $F^{(t)}(j)$ and the null F-distribution F_0, respectively.
    $P^{(t)}_{F,j} \leftarrow 1 - F_0(F^{(t)}_{\text{adj}}(j))$
  end for

Step 2.2 Calculate weight for F-test

  For $\{1 \le m \le p \mid m \in I^{(t-1)}\}$, $\pi^{(t-1)}_m = \text{mean}(F^{(t)}_{\text{adj}}(m) < F_{\text{random}})$
  Sort $\pi^{(t-1)}$: $\pi^{(t-1)}_{(1)} \le \pi^{(t-1)}_{(2)} \le \cdots \le \pi^{(t-1)}_{(s^{(t-1)})}$, where $s^{(t-1)} = |I^{(t-1)}|$
  $p^{(t)}_1 = 1 - \exp(-e^{c-bT})$, where $T = \max_{1\le j\le 2s^{(t-1)}/3} \sqrt{s^{(t-1)}} \frac{j/s^{(t-1)} - \pi^{(t-1)}_{(j)}}{\sqrt{\pi^{(t-1)}_{(j)}(1-\pi^{(t-1)}_{(j)})}}$
  Weight $w^{(t)} = 1 - p^{(t)}_1/(p^{(t)}_1 + 0.6)$

Step 2.3: Compute core statistic

  For each feature j, core statistic is $S^{(t)}_j = w^{(t)}\Phi^{-1}(1 - P^{(t)}_{F,j}) + (1 - w^{(t)})\Phi^{-1}(1 - P_{KS,j})$, where Φ^{-1} is inverse standard normal distribution

Step 2.4: Calculate threshold

  For each feature j, $\pi^{(t)}_j = \Phi(1 - S^{(t)}_j/\sqrt{(w^{(t)})^2 + (1-w^{(t)})^2})$, where Φ is standard normal distribution
  Sort the p-values as $\pi^{(t)}_{(1)} \le \pi^{(t)}_{(2)} \le \cdots \le \pi^{(t)}_{(p)}$
  The HCT can be found as $\tau^{(t)} = S^{(t)}_{\hat{j}}$, where $\hat{j} = \arg\max_{\log p\le j\le p/2} \frac{j/p - \pi^{(t)}_{(j)}}{\sqrt{\pi^{(t)}_{(j)}(1-\pi^{(t)}_{(j)})}}$
  $I^{(t)} = \{1 \le j \le p \mid S^{(t)}_j \ge \tau^{(t)}\}$

Step 2.5: Reduce dimensions and cluster

  Apply Laplacian Eignmap or PCA on $W[:, I^{(t)}]$, retain the top $K+2$ eigenvectors to form a spectral matrix $U^{(t)}$
  Perform k-means on $U^{(t)}$, then $\ell^{(t)} = k\text{-means}(U^{(t)}, K)$
  if $r^{(t)} = \frac{|I^{(t)}/I^{(t-1)}|}{|I^{(t-1)}|} < 10\%$ **then**
    break
  end if
end for

Algorithm 5 DeepCluster with Autoencoder and Hyperparameter Optimization

Require: Input data X, number of clusters K
Ensure: Predicted cluster labels ℓ
Initialize an Optuna study to maximize clustering performance
for each trial in Optuna **do**
  Sample hyperparameters: hidden size h, epochs E, iterations T
  Initialize an autoencoder model with encoder, decoder, and a classification head $\ell^{(0)}$
  for each iteration $t = 1$ to T **do**
    Encode input W to get low-dimensional features z
    Normalize z and apply k-means with K clusters to obtain pseudo-labels
    for each epoch $e = 1$ to E **do**
      Decode z to reconstruct input, and classify using pseudo-labels
      Compute total loss: reconstruction loss + classification loss
      Update model parameters via backpropagation
    end for
  end for
  Compute silhouette score s based on final cluster assignments
  Define objective score as: $s - 0.5 \cdot \text{final loss}$
end for
Retrieve the best hyperparameters from Optuna
Train the model again using the best settings, obtion predicted cluster labels ℓ

Algorithm 6 Clustering with DEC

Require: Input data X, number of clusters K
Ensure: Predicted cluster labels ℓ
Define an autoencoder: encoder $f_\phi(x) = z$, decoder $g_\theta(z) = \hat{x}$
for epoch $= 1$ to N_{pretrain} **do**
  Compute reconstruction: $\hat{x} = g_\theta(f_\phi(x))$
  Minimize MSE loss: $\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|^2$
end for
Encode all data: $z = f_\phi(x)$
Apply k-means on z to obtain cluster centers $\{\mu_k\}_{k=1}^K$
Initialize cluster layer with these centers
for epoch $= 1$ to N_{DEC} **do**
  Encode $z = f_\phi(x)$ and compute soft assignments q_{ik}: $q_{ik} = \frac{(1+\|z_i-\mu_k\|^2)^{-1}}{\sum_j (1+\|z_i-\mu_j\|^2)^{-1}}$
  Compute target distribution p: $p_{ik} = \frac{q_{ik}^2 / \sum_i q_{ik}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})}$
  Minimize KL divergence loss: $\mathcal{L}_{\text{KL}} = \text{KL}(P\|Q) = \sum_i \sum_k p_{ik} \log \frac{p_{ik}}{q_{ik}}$
  Total loss: $\mathcal{L} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{recon}}$
  Update model parameters via gradient descent
end for
Assign cluster label $\ell_i = \arg\max_k q_{ik}$ for each sample

Algorithm 7 Clustering with UMAP

Require: Input data X, number of clusters K
Ensure: Predicted labels ℓ
Obtain top $K + 2$ eigenvectors for XX^T, and use them as initialization for UMAP optimizer
Apply UMAP on X, retain the top $K + 2$ eigenvectors to form a spectral matrix U
Perform k-means on U, then predicted labels $\ell = k\text{-means}(U, K)$

Algorithm 8 Clustering with VAE

Require: Input data X, number of clusters K
Ensure: Predicted labels ℓ
1: Define encoder network $q_\phi(z|x)$ that maps input X to latent mean μ and log-variance $\log \sigma^2$
2: Use reparameterization trick: $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$
3: Define decoder network $p_\theta(w|x)$ to reconstruct input from latent vector
4: Train the VAE by minimizing the loss: $\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta \cdot D_{\mathrm{KL}}(q_\phi(z|x)\|p(z))$
5: Use a warm-up schedule to gradually increase β from 0 to 1 during training
6: Encode all samples to latent space: $Z = \mu(x)$ for each x
7: Apply k-means clustering on Z with K clusters, obtain predicted labels ℓ

C IMPLEMENTATION DETAILS AND PARAMETERS

In the following subsections, we present the implementation details and hyperparameter settings of all methods involved in our study. Specifically, i-IF-Lap is our proposed iterative feature selection algorithm that incorporates Laplacian embedding to guide low-dimensional clustering. It constructs a cosine-based affinity graph and applies spectral embedding for representation learning. DeepCluster is a self-supervised deep clustering method that alternates between clustering with K-means and updating a feature encoder, which we adapt to our tabular datasets using lightweight autoencoders. IFVAE and i-IF-Lap+VAE are based on the Variational Autoencoder framework, where clustering is performed in the learned latent space, and i-IF-Lap+VAE further integrates our iterative feature selection procedure. DEC (Deep Embedded Clustering) jointly optimizes a clustering loss and a deep autoencoder, and has been used as a strong baseline for representation-based clustering. UMAP is a non-linear dimensionality reduction technique that preserves both local and global structure of the data, and we use it both as a baseline and as part of i-IF-Lap+UMAP. Each method is implemented with reasonable default parameters or carefully tuned hyperparameters, as detailed in the sections below.

C.1 i-IF-Lap

In i-IF-Lap, implementation details for Laplacian Eignmap are:

Cosine distance is computed between all pairs of feature vectors in $W[:, I^{(t)}]$.

The affinity matrix $A \in \mathbb{R}^{n \times s^{(t)}}$ is constructed using a Gaussian kernel applied to the cosine distances:

$$A_{ij} = \exp(-\gamma \cdot d_{ij}^2),$$

where d_{ij} is the cosine distance between feature vectors i and j, and γ is a scaling parameter.

For parameters:

- **Gamma (γ):** 1
- **Affinity type:** Precomputed affinity matrix
- **Number of nearest neighbors:** 8
- **Output dimensionality:** $K + 2$
- **Implementation:** `SpectralEmbedding` from `scikit-learn`

The resulting low-dimensional representation $U \in \mathcal{R}^{n \times (K+2)}$ is subsequently used for clustering.

C.2 DeepCluster

Our dataset is relatively small and low-dimensional compared to image dataset, and therefore does not require a deep or complex neural network architecture.

For both Deepcluster and i-IF-Lap+DeepCluster, we use `optuna` to obtain optimal parameters. The parameters tuned include:

- h: the size of the Autoencoder's hidden layer, selected from {64, 128, 256}.
- E: the number of training epochs per clustering iteration, ranging from 5 to 15.
- T: the total number of clustering-training iterations, ranging from 3 to 10.
- **learning_rate**: 1×10^{-3}

C.3 VAE

We use the IFVAE implementation provided under the GNU GPL by Chen et al. (2023) Chen et al. (2023), as instructed in the repository license. The citation to the original paper is included in our manuscript. For both IFVAE and i-IF-Lap+VAE, parameters are:

- **latent_dim**: Dimensionality of the latent space in the VAE, set to 25.
- **batch_size**: Mini-batch size during training, set to 50.
- **epochs**: Total number of training epochs, set to 100.
- **learning_rate**: Learning rate used in the optimizer, set to 0.0005).
- **kappa**: Warm-up increment per epoch for β in the KL divergence term, set to 1.

C.4 DEC

We use the publicly available DEC implementation released under the MIT License by Junyuan Xie (2015) Xie et al. (2016). The license permits use, modification, and redistribution with appropriate credit. For parameters in DEC:

- **Hidden layer dimensions:** A list specifying the number of neurons in the encoder and decoder layers, set to `[500, 10]` for an encoder of size $p \to 500 \to 10$ and a mirrored decoder.
- **Pretraining epochs (N_{pretrain}):** Number of epochs for unsupervised autoencoder pretraining, set to 10.
- **DEC training epochs (N_{DEC}):** Number of epochs for joint clustering optimization, set to 100.
- **Batch size:** Number of samples per training batch, set to 256.
- **Learning rate:** Learning rate for the optimizer, set to 1×10^{-3}.

C.5 UMAP

For both UMAP and i-IF-Lap+UMAP, parameters are:

- **Number of neighbors:** 8
- **Metric:** Cosine distance
- **Embedding dimensionality:** $K + 2$
- **Angular random projection forest:** Enabled (`angular_rp_forest=True`)
- **Implementation:** `umap.UMAP` from the `umap` Python package

D DETAILS ABOUT NUMERICAL EXPERIMENTS

D.1 Computer Resources

All experiments were conducted on Amazon Web Services (AWS) using m5.large instances. The key specifications of the compute environment are as follows:

- **Instance type:** AWS m5.large
- **CPU:** Intel(R) Xeon(R) Platinum 8175M CPU @ 2.50GHz
- **Cores/Threads:** 2 cores, 4 threads (Hyperthreading enabled)
- **Memory:** 8 GB RAM
- **GPU:** None (CPU-only setup)
- **Virtualization:** KVM hypervisor

D.2 Datasets

To facilitate comparative analysis, following tables summarize the key characteristics of the benchmark data sets used in this study. For each data set, we report three key quantities: the number of samples n, the number of features p, and the number of cluster K.

We use a set of publicly available gene microarray datasets in our study. Download datasets by following links: https://data.mendeley.com/datasets/nv2x6kf5rd/1, https://data.mendeley.com/datasets/cdsz2ddv3t/1. These datasets are licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0), as indicated on the data hosting platform. Table 8 includes the 8 scRNA-seq data sets, while Table 7 contains the 10 microarray data sets. This alignment enables consistent assessment of algorithmic performance across diverse biological contexts.

	Data name	Source	K	n	p
1	Brain	Pomeroy (02)	5	42	5,597
2	Breast cancer	Wang et al. (05)	2	276	22,215
3	Colon cancer	Alon et al. (99)	2	62	2,000
4	Leukemia	Golub et al. (99)	2	72	3,571
5	Lung cancer (1)	Gordon et al. (02)	2	181	12,533
6	Lung cancer (2)	Bhattacharjee et al. (01)	2	203	12,600
7	Lymphoma	Alizadeh et al. (00)	3	62	4,026
8	Prostate cancer	Singh et al. (02)	2	102	6,033
9	SRBCT	Kahn (01)	4	63	2,308
10	Su cancer	Su et al. (01)	2	174	7,909

Table 7: Summary of microarray datasets with K (number of clusters), n (samples), and p (features).

For scRNA-seq data sets except Patel, we add log transformation ($X = log(X + 1)$) on data matrices.

	Data set	K	n	p
1	Camp1	7	777	13,111
2	Camp2	6	734	11,233
3	Darmanis	9	466	13,400
4	Deng	6	268	16,347
5	Goolam	5	124	21,199
6	Grun	2	1,502	5,547
7	Li	9	561	25,369
8	Patel	5	430	5,948

Table 8: Summary of scRNA-seq datasets with K (number of clusters), n (samples), and p (features).

D.3 Additional Simulation Results in Synthetic Datasets

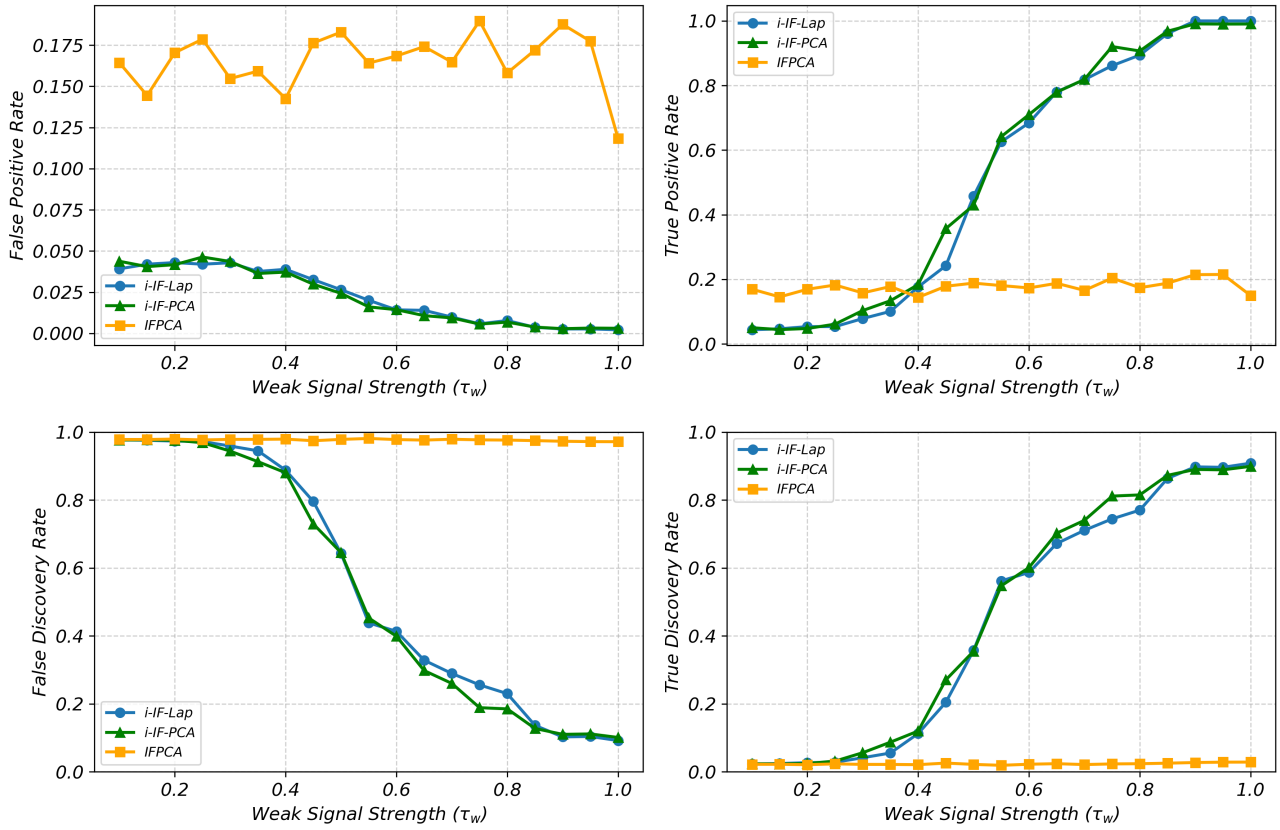

Figure 5: Comparison of feature selection performance under increasing weak signal strength (τ_w). Each subplot reports a different metric: (a) False Positive Rate (FPR), (b) True Positive Rate (TPR), (c) False Discovery Rate (FDR), and (d) True Discovery Rate (TDR).

We compare three feature selection methods: baseline IFPCA, our proposed i-IF-PCA and i-IF-Lap. As τ_w increases, we provide detailed insights into each subplot of Figure 5:

- **(a) False Positive Rate (FPR).** As τ_w increases, both i-IF-PCA and i-IF-Lap significantly reduce FPR, while IF-PCA maintains a consistently high FPR across all levels. This indicates that our proposed iterative methods are much more effective in suppressing noise features, especially as signal strength grows.
- **(b) True Positive Rate (TPR).** The TPR for IFPCA remains stagnant, failing to recover true signals

under increasing τ_w. In contrast, both i-IF-PCA and i-IF-Lap demonstrate a clear transition from low to high TPR as τ_w increases, indicating their capacity to adaptively extract true features. Notably, i-IF-Lap and i-IF-PCA achieve near-perfect TPR when $\tau_w > 0.7$.

- **(c) False Discovery Rate (FDR).** IFPCA shows extremely high FDR (close to 1), suggesting that nearly all selected features are false discoveries. Both i-IF variants show decreasing FDR as τ_w increases, with i-IF-PCA slightly outperforming i-IF-Lap under high signal strengths.
- **(d) True Discovery Rate (TDR).** TDR follows a similar trend to TPR. The iterative methods rapidly increase TDR with growing τ_w, again highlighting their adaptability. i-IF-Lap and i-IF-PCA reach near-perfect TDR beyond $\tau_w = 0.8$, indicating very high fidelity in recovering true features.

Conclusion. These results further confirm that iterative frameworks (i-IF-PCA and i-IF-Lap) significantly outperform the static IFPCA in both reducing false selections and recovering weak signals, particularly when weak signals become stronger.

E ADDITIONAL EXPERIMENT FOR EMBEDDING

We applied our *i-IF-Learn* framework with different embedding methods on scRNA-seq datasets. We compared four popular dimensionality reduction techniques: UMAP, Autoencoder, Laplacian Eigenmap, and PCA. The clustering accuracy is reported in the form of mean (standard deviation) over 30 repetitions.

Data	UMAP	Autoencoder	Laplacian	PCA
camp1	**0.804 (0.0012)**	0.671 (0.0418)	0.740 (0.0000)	0.738 (0.0000)
camp2	0.546 (0.0041)	**0.630 (0.0137)**	0.605 (0.0000)	0.617 (0.0000)
darmanis	0.720 (0.0227)	0.781 (0.0222)	**0.785 (0.0000)**	0.783 (0.0000)
deng	0.626 (0.0083)	0.861 (0.0026)	**0.869 (0.0000)**	0.802 (0.0000)
goolam	0.665 (0.0109)	**0.916 (0.0750)**	0.758 (0.0000)	0.629 (0.0000)
grun	0.672 (0.0027)	0.692 (0.0087)	**0.994 (0.0000)**	0.991 (0.0000)
li	0.943 (0.0135)	0.897 (0.0019)	0.966 (0.0000)	**0.980 (0.0000)**
patel	0.946 (0.0028)	0.771 (0.0284)	**0.942 (0.0000)**	0.788 (0.0000)

Table 9: Clustering accuracy of different embedding methods across scRNA-seq datasets using i-IF-Learn.

Overall, the results show that Laplacian Eigenmap achieves the best performance across multiple datasets, demonstrating both high accuracy and stability. While UMAP and Autoencoder sometimes achieve the highest accuracy for specific datasets, their performance is less stable, with larger standard deviations. PCA performs well on datasets with linear structure, but is generally outperformed by Laplacian Eigenmap in most other cases.

F ADDITIONAL BASELINE COMPARISONS: IDC AND CLEAR

To further comprehensively evaluate our proposed method, we conducted additional experiments comparing i-IF-Lap against two other baseline methods: IDC and CLEAR.

First, we applied IDC to our datasets. However, we encountered out-of-memory constraints on the larger datasets, limiting its successful application to 6 datasets. As shown in Table 10, our i-IF-Lap method outperforms IDC on 5 out of the 6 evaluated datasets.

Additionally, we compared i-IF-Lap with CLEAR across 8 scRNA-seq datasets. As detailed in Table 11, i-IF-Lap demonstrates superior clustering accuracy, consistently outperforming CLEAR across all 8 datasets.

Table 10: Clustering accuracy comparison between IDC and i-IF-Lap on 6 datasets. IDC encountered out-of-memory errors on the remaining larger datasets.

Method	Brain	Colon	Lymphoma	Leukemia	Prostate	SRBCT
IDC	0.238	**0.645**	0.645	0.653	0.510	0.524
i-IF-Lap	**0.783**	0.635	**0.936**	**0.972**	**0.569**	**0.984**

Table 11: Clustering accuracy comparison between CLEAR and i-IF-Lap on 8 scRNA-seq datasets.

Method	Camp1	Camp2	Darmanis	Deng	Goolam	Grun	Li	Patel
CLEAR	0.597	0.426	0.487	0.549	0.742	0.540	0.886	0.540
i-IF-Lap	**0.740**	**0.605**	**0.785**	**0.869**	**0.758**	**0.994**	**0.966**	**0.942**

G ADDITIONAL EXPERIMENT: LASSO WITH PSEUDO-LABELS

The reviewer suggested comparing our approach with supervised feature selection methods such as Lasso. While these methods are powerful when reliable labels are available, our problem is inherently unsupervised and relies on pseudo-labels generated during the iterative procedure. As discussed in the main text, pseudo-labels in early iterations can be noisy, and supervised feature selection methods may treat these labels as ground truth, potentially leading to error propagation.

To further examine this issue, we conducted an additional experiment using Lasso-based feature selection on the microarray datasets. Specifically, we applied Lasso to the standardized data using the pseudo-labels produced by the iterative procedure, and evaluated the classification accuracy using the selected features.

Methods	Brain	Breast	Colon	Leukemia	Lung1	Lung2	Lymphoma	Prostate	SRBCT	SuCancer
Lasso	0.643	0.627	0.597	0.931	0.967	0.783	0.468	**0.618**	0.460	0.500
i-if-lap	**0.738**	**0.630**	**0.597**	**0.972**	**0.995**	**0.803**	**0.936**	0.569	**0.984**	**0.603**

Table 12: Comparison between Lasso-based feature selection and the i-IF-Lap on microarray datasets.

As shown in Table 12, our proposed i-IF-Lap method outperforms the Lasso-based approach on 9 out of the 10 evaluated microarray datasets. Notably, in datasets such as *Lymphoma* and *SRBCT*, the accuracy of Lasso drops drastically compared to our framework. This substantial performance gap empirically validates our hypothesis: explicitly treating early-stage, noisy pseudo-labels as absolute ground truth—as standard supervised methods like Lasso inherently do—leads to severe error propagation. In contrast, our adaptive screening metric successfully mitigates this risk by dynamically balancing pseudo-label supervision with unsupervised signals, demonstrating the necessity and superiority of our tailored unsupervised framework.