

Unsupervised Feature Selection Through Group Discovery

Shira Lifshitz¹, Ofir Lindenbaum², Gal Mishne³, Ron Meir¹, Hadas Benisty¹

¹Technion – Israel Institute of Technology

²Bar Ilan University

³University of California San Diego

shiralif@campus.technion.ac.il, ofirlin@gmail.com, gmishne@ucsd.edu, rmeir@ee.technion.ac.il, hadasbe@technion.ac.il

Abstract

Unsupervised feature selection (FS) is essential for high-dimensional learning tasks where labels are not available. It helps reduce noise, improve generalization, and enhance interpretability. However, most existing unsupervised FS methods evaluate features in isolation, even though informative signals often emerge from groups of related features. For example, adjacent pixels, functionally connected brain regions, or correlated financial indicators tend to act together, making independent evaluation suboptimal. Although some methods attempt to capture group structure, they typically rely on predefined partitions or label supervision, limiting their applicability. We propose GroupFS, an end-to-end, fully differentiable framework that jointly discovers latent feature groups and selects the most informative groups among them, without relying on fixed a priori groups or label supervision. GroupFS enforces Laplacian smoothness on both feature and sample graphs and applies a group sparsity regularizer to learn a compact, structured representation. Across nine benchmarks spanning images, tabular data, and biological datasets, GroupFS consistently outperforms state-of-the-art unsupervised FS in clustering and selects groups of features that align with meaningful patterns.

Extended version — <https://arxiv.org/abs/2511.09166>

1 Introduction

Modern machine-learning systems routinely handle datasets with thousands to millions of features. Such high-dimensional data arise in neuroscience, finance, and computer vision (Fan and Lv 2010; Donoho et al. 2000). However, many of the observed features are nuisance, i.e., uninformative or noisy, and they obscure latent structure, inflate computational cost, and degrade generalization. Feature selection (FS) tackles this problem by retaining only the most relevant features, thereby discarding nuisance dimensions, reducing computational cost, and boosting downstream performance, e.g., clustering accuracy (Guyon and Elisseeff 2003). Because FS preserves the original measurements, the results remain interpretable and can enable domain-specific insights. In real-world applications such as neuroimaging, FS can lower acquisition costs by focusing on task-relevant

regions, thus saving time or enabling higher resolution. Similarly, in domains like behavioral research, feature acquisition (e.g., questionnaires) can be expensive or burdensome, making efficient feature selection especially valuable.

While many FS methods are supervised, even without labels, a well-chosen subset of features can uncover latent structure (Li et al. 2017). Yet selecting that subset is a complex combinatorial problem, and the challenge is amplified in the unsupervised setting where there are no labels to guide the selection process. Since obtaining annotations often requires costly expert effort, robust unsupervised FS is both challenging and essential (Solorio-Fernández, Carrasco-Ochoa, and Martínez-Trinidad 2020; Li et al. 2024).

Classical FS methods can be categorized into three families. *Filter* methods assign scores to features using model-agnostic criteria such as mutual information or graph smoothness (He, Cai, and Niyogi 2005; Battiti 1994). *Wrapper* methods search over subsets by repeatedly training a model, incurring high computational cost (Kohavi and John 1997). *Embedded* methods impose sparsity while training the model itself, e.g. LASSO (Tibshirani 1996) or stochastic-gating networks (Yamada et al. 2020; Sristi et al. 2023). Most of these approaches, however, score features independently and ignore relationships among them.

Real-world features often “act together”: spatially adjacent pixels, temporally co-varying sensors, or functionally coupled genes. Such relationships suggest that grouping features into meaningful subsets and selecting at the group level, rather than individually, can boost performance and provide clearer scientific insights. Existing group-aware FS methods either assume groups are known a priori or rely on supervision to form them (You et al. 2023; Imrie et al. 2022). However, in many applications, the group structure is unknown, and fixing groups in advance can bias selection. Jointly discovering groups, selecting which ones are informative and rejecting the rest, without labels, remains an open problem, referred to as unsupervised group feature selection.

In this paper, we address this gap by introducing **GroupFS**. It is a fully differentiable, end-to-end framework that simultaneously learns feature groups and selects the informative ones in a purely unsupervised manner. Our approach constructs two graphs: one over the sample space and

another over the feature space, enforcing Laplacian smoothness on both. A feature-grouping and gating mechanism, guided by sparse regularization, dynamically discovers relevant feature groups.

Our main contributions are as follows:

- We introduce GroupFS, the first end-to-end, fully differentiable framework for unsupervised feature selection that jointly discovers latent feature groups and selects informative groups from them.
- GroupFS automatically learns latent feature groups without relying on predefined partitions or supervision, thereby broadening its applicability to unlabeled, real-world data.
- Extensive experiments on diverse synthetic and real-world datasets demonstrate that GroupFS consistently outperforms state-of-the-art unsupervised FS baselines in clustering accuracy and identifies meaningful feature groups.

2 Related Work

Unsupervised FS. One line of research addresses the unsupervised FS problem by constructing a sample graph and selecting features that vary smoothly over the data manifold (He, Cai, and Niyogi 2005; Cai, Zhang, and He 2010; Lindenbaum et al. 2021; Miao et al. 2022; Luo et al. 2024). Autoencoder-based methods offer an alternative, ranking features by their contribution to reconstruction loss (Abid, Balin, and Zou 2019; Svirsky and Lindenbaum 2024). However, reconstruction objectives do not necessarily promote features that capture the relationships among samples, which are essential for downstream tasks such as clustering.

Group FS. Other work seeks to exploit feature groups, but most approaches assume the groups are fixed a priori, using heuristics or domain knowledge to define them (You et al. 2023; Zaharieva, Breiteneder, and Hudec 2017; Perera, Chan, and Karunasekera 2020; Wang et al. 2017; Park and Lee 2024). Although effective in special cases, predefined groups limit adaptability and can introduce bias. A more flexible strategy is to learn groups during training. Imrie et al. (2022) take a step in this direction by jointly inferring group structure and training a classifier, but they rely on label supervision. Sristi, Mishne, and Jaffe (2022) proposes a spectral approach to select groups of features; however, they assume a setting of differentiating between two or more given datasets.

In contrast to these, we jointly discover feature groups and select the informative ones without any supervision, allowing structure to emerge directly from the data.

3 Preliminaries

3.1 Graphs and Spectral Analysis

Let $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d}$ be a data matrix, where row i (sample i) is $\mathbf{x}_i = X_{i,:} \in \mathbb{R}^d$, and column k (feature k) is $\mathbf{x}^{(k)} = X_{:,k} \in \mathbb{R}^N$. We assume the data lies on a low-dimensional manifold and capture its local geometry using an undirected, weighted graph $G = (V, E, W)$ (Von Luxburg 2007; Ng, Jordan, and Weiss

2001). Pairwise affinities are defined using the self-tuning kernel (Zelnik-Manor and Perona 2004) as

$$W_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\gamma_i \gamma_j}\right), \quad (1)$$

where γ_i is the distance from \mathbf{x}_i to its K -th nearest neighbor. This sample-dependent scaling adapts to local density, improving robustness in heterogeneous data. The degree matrix is defined as $D = \text{diag}(d_1, \dots, d_N)$, where $d_i = \sum_j W_{ij}$. Two standard graph operators are the normalized graph Laplacian $L_{\text{sym}} = I - D^{-1/2} W D^{-1/2}$ and the random walk matrix $P = D^{-1} W$. Though both are used in spectral analysis, they differ in interpretation: eigenvectors corresponding to low eigenvalues of L_{sym} capture smooth, low-frequency variations on the graph, while those associated with high eigenvalues of P capture similar directions. The matrix power P^t represents transition probabilities after t steps of the random walk (Spielman 2025).

Laplacian Score (LS). He, Cai, and Niyogi (2005) leveraged a sample graph structure for feature selection by ranking individual features based on their alignment with the graph’s smoothest modes. Features whose values vary minimally across strongly connected samples (i.e., along high-weight edges) are preferred. For a feature vector $\mathbf{x}^{(k)} \in \mathbb{R}^N$, the Laplacian Score is

$$\text{LS}(\mathbf{x}^{(k)}) = (\mathbf{x}^{(k)})^\top L_{\text{sym}} \mathbf{x}^{(k)} = \sum_{i=1}^N \lambda_i \langle \mathbf{v}_i, \mathbf{x}^{(k)} \rangle^2,$$

where $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^N$ are the eigenpairs of L_{sym} . A smaller score indicates that the feature varies smoothly over the sample manifold and is therefore considered more informative. Using the matrix trace operator tr , the total Laplacian score becomes

$$\sum_{k=1}^d (\mathbf{x}^{(k)})^\top L_{\text{sym}} \mathbf{x}^{(k)} = \text{tr}(X^\top L_{\text{sym}} X).$$

3.2 Gumbel-Softmax

The Gumbel-Softmax (Jang, Gu, and Poole 2016), also known as the Concrete distribution (Maddison, Mnih, and Teh 2016), provides a differentiable approximation to categorical sampling. It relaxes the discrete one-hot vector into a continuous distribution over C classes. Given class probabilities $\boldsymbol{\pi} = [\pi_1, \pi_2, \dots, \pi_C]$ and a temperature $T > 0$, we draw i.i.d. Gumbel noise variables $g_c \sim \text{Gumbel}(0, 1)$ and compute

$$m_c = \frac{\exp((\log \pi_c + g_c)/T)}{\sum_{h=1}^C \exp((\log \pi_h + g_h)/T)}.$$

As $T \rightarrow 0$, the distribution becomes increasingly peaked, and $\mathbf{m} \in \mathbb{R}^C$ approaches a one-hot sample. The reparameterization trick (Kingma, Welling et al. 2013) enables gradients to propagate through the sampling process, allowing Gumbel-Softmax to be trained using standard gradient-based optimizers (Battash, Wolf, and Lindenbaum 2024).

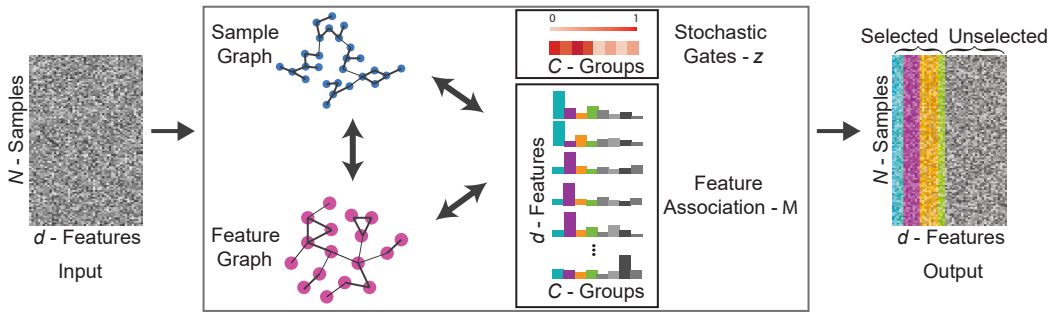


Figure 1: Illustration: GroupFS learns feature-to-group associations, enforces smoothness on the feature graph, infers the importance of each group, and reconstructs a smoother sample-similarity graph.

3.3 Stochastic Gates

The stochastic gates method (Yamada et al. 2020; Jana et al. 2023) provides a differentiable mechanism for feature selection by learning relaxed Bernoulli gates for the features. Each input feature $\mathbf{x}^{(k)}$, for $k \in \{1, \dots, d\}$, is multiplied by a stochastic gate (STG): $z_k = \max(0, \min(1, \mu_k + \varepsilon_k))$ where $\varepsilon_k \sim \mathcal{N}(0, \sigma^2)$ and μ_k are learnable parameters. This clipped Gaussian variable produces continuous approximations of binary gates. The expected number of selected features is

$$\mathbb{E}[\|\mathbf{z}\|_0] = \sum_{k=1}^d \mathbb{P}(z_k > 0) = \sum_{k=1}^d \Phi\left(\frac{\mu_k}{\sigma}\right),$$

where $\Phi(\cdot)$ denotes the standard Gaussian CDF and σ is the fixed gate noise. This relaxation enables training with standard gradient-based optimizers while implicitly encouraging sparsity through ℓ_0 -style regularization.

4 GroupFS

We tackle unsupervised group FS without assuming prior knowledge about the groups. Instead, we simultaneously learn feature groups and select the most relevant ones, yielding a compact and interpretable model (see Figure 1).

Problem setup. Let $X \in \mathbb{R}^{N \times d}$ be a data matrix with N samples and d raw features. We assume the features can be partitioned into C latent groups $\{\mathcal{G}_1, \dots, \mathcal{G}_C\}$, which are unknown a priori. Our method uncovers latent groups of related features, selects specific groups needed to preserve the data’s intrinsic geometry, and learns a low-dimensional embedding that faithfully reflects that geometry. Our model is guided by a composite loss function consisting of three components:

- **Sample-wise smoothness (\mathcal{L}_s):** Encourages feature values to vary smoothly across the sample manifold, promoting gradual transitions between nearby data points.
- **Feature-wise smoothness (\mathcal{L}_f):** Encourages consistent group assignments among high-affinity neighbors on the feature graph.
- **Group sparsity (\mathcal{L}_{reg}):** Promotes the selection of a small number of informative feature groups, resulting in a compact and interpretable model.

Overall Loss. Our objective combines the three components:

$$\mathcal{L} = \mathcal{L}_s + \lambda_1 \cdot \mathcal{L}_f + \lambda_2 \cdot \mathcal{L}_{\text{reg}},$$

where λ_1 and λ_2 weigh the relative importance of each term. This unified, end-to-end framework integrates differentiable grouping, stochastic gating, and Laplacian smoothness to discover informative feature groups while filtering out irrelevant or noisy features in an unsupervised setting. We describe each component below.

4.1 Sample-wise Smoothness Loss \mathcal{L}_s

Feature Association. Given a batch $X_B \in \mathbb{R}^{B \times d}$, we learn a feature-to-group assignment matrix $M \in \mathbb{R}^{d \times C}$ using the Gumbel-Softmax trick (see Sec. 3.2). Each row $M_{i \cdot}$ encodes the soft membership of feature i across the C latent groups,

$$M_{ij} = \frac{\exp((\log \pi_{ij} + g_{ij})/T)}{\sum_{k=1}^C \exp((\log \pi_{ik} + g_{ik})/T)},$$

where π_{ij} are learnable logits, $g_{ij} \sim \text{Gumbel}(0, 1)$ is i.i.d. noise, and T is a temperature parameter annealed during training. As $T \rightarrow 0$, each row approaches a one-hot vector, effectively assigning feature i to a single group. Since M is learned jointly with the group-importance gates (see following sub-section), the model can discover meaningful groupings directly from the data.

Group Importance. Following the STG formulation (see Sec. 3.3), we attach a stochastic gate z_j to each feature group, reducing the number of learnable gating parameters from d (features) to C (groups). We select features by sorting the groups according to their gate means and retaining those from the top-ranked groups. To compute feature-level weights, we aggregate the gated group assignments:

$$\hat{z}_i = \sum_{j=1}^C M_{ij} \cdot z_j, \quad i \in \{1, \dots, d\}, \quad j \in \{1, \dots, C\}.$$

Intuitively, \hat{z}_i measures the importance of feature i by averaging over its soft group memberships weighted by each group’s gate. Broadcasting $\hat{\mathbf{z}}$ across the batch yields $\hat{Z} \in \mathbb{R}^{B \times d}$ with identical rows. We apply these gates to the input using element-wise multiplication: $\tilde{X} = X_B \odot \hat{Z}$, effectively masking out less important features.

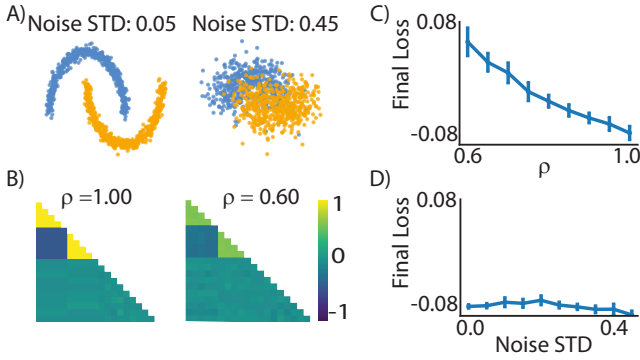


Figure 2: Two-moons synthetic data. (A) 2D visualization of the dataset under low and high Gaussian noise levels (STD = 0.05 and STD = 0.45). (B) Feature correlation matrices (20×20 , lower triangle) with two levels of correlation strength ($\rho = 1.00$ and $\rho = 0.60$). (C) Final training loss as a function of correlation strength ρ , showing lower loss for stronger correlations. (D) Final training loss as a function of noise standard deviation, showing robustness to moderate sample-level noise. Results in (C,D) are averaged over 10 random seeds; error bars denote standard error.

Smoothness Objective. We construct a dense random walk matrix $P_{\tilde{X}}$ (Sec. 3.1) over the batch-masked input \tilde{X} at each iteration, using the affinity matrix defined in eq. (1). To promote smooth variation over the gated sample manifold, we use

$$\mathcal{L}_s = -\frac{1}{Bd} \operatorname{tr} \left(\tilde{X}^\top P_{\tilde{X}}^t \tilde{X} \right).$$

Here, $P_{\tilde{X}}^t$ denotes the t -step diffusion operator, i.e., the t -th power of the random-walk matrix. Maximizing the trace aligns the retained features with the graph’s low-frequency directions. This encourages the model to assign higher importance to feature groups whose values vary smoothly across the sample manifold.

\mathcal{L}_s combines the group assignments M with their gate importances \mathbf{z} to select a subset of groups, and then rebuilds the sample graph using only these features. On this newly constructed graph, the retained features vary smoothly, turning the raw data into a cleaner and more informative representation.

4.2 Feature-wise Smoothness Loss \mathcal{L}_f

This term encourages the learned cluster assignments to vary smoothly across features while remaining mutually distinct. We first embed each feature into a C -dimensional space: $F = MQ \in \mathbb{R}^{d \times C}$ where $M \in \mathbb{R}^{d \times C}$ is the soft assignment matrix from Section 4.1, and $Q \in \mathbb{R}^{C \times C}$ is a trainable linear projection that allows interactions between clusters.

Smoothness on the feature graph. Analogous to the sample graph in Section 3.1, we construct a feature similarity graph, where nodes represent features, and compute its normalized Laplacian $L_{\text{feat}} \in \mathbb{R}^{d \times d}$. The term $\operatorname{tr} (F^\top L_{\text{feat}} F)$ penalizes rapid changes of F across similar features, promoting alignment with the graph’s low-frequency directions.

Orthogonality regularization. To ensure diverse, non-redundant cluster embeddings, analogous to orthogonal eigenvectors in spectral clustering, we add an orthogonality penalty using the Frobenius norm: $\|F^\top F - I\|_F^2$. Combining both objectives yields:

$$\mathcal{L}_f = \frac{1}{dC} \left[\operatorname{tr} (F^\top L_{\text{feat}} F) + \beta \cdot \|F^\top F - I\|_F^2 \right],$$

where β is a hyper-parameter that weights the orthogonality penalty. Following each update step, we center the columns of F to have zero mean and renormalize them to unit ℓ_2 -norm. This avoids convergence to trivial solutions such as constant vectors or the zero vector.

Intuitively, \mathcal{L}_f term encourages similar features (or highly connected nodes in the feature graph) to have similar group assignments, ensuring that the learned groups respect the underlying structure of the feature space.

4.3 Group Sparsity Loss \mathcal{L}_{reg}

To encourage selection of the most informative groups, we penalize the expected number of active gates, weighted by each group’s relative size. Using the activation probability for STG gates (Sec. 3.3), we define the regularization term

$$\mathcal{L}_{\text{reg}} = \frac{1}{C} \sum_{j=1}^C \mathbb{P}(z_j > 0) \cdot \frac{1}{d} \sum_{i=1}^d M_{ij},$$

which increases with both the likelihood that group j is active and the proportion of features assigned to it. Minimizing \mathcal{L}_{reg} therefore encourages compactness and sparsity by keeping fewer and smaller groups active.

5 Experiments

We evaluate **GroupFS** across three complementary settings:

1. **Synthetic data.** We construct a synthetic dataset with features partitioned into known groups. This setup allows us to (i) verify whether GroupFS correctly recovers and selects the true groups, and (ii) study the effects of hyperparameters and intrinsic data properties.
2. **Real-world data.** To assess whether explicit feature grouping enhances or hinders FS, we compare GroupFS to state-of-the-art baselines across nine widely used datasets from image and biological domains.
3. **Interpretability.** We demonstrate that GroupFS discovers meaningful feature groups that align with domain knowledge.

5.1 Implementation Details

The model’s learnable parameters are: (i) $d \times C$ logits of the Gumbel-Softmax assignment matrix M ; (ii) C gate means $\{\mu_j\}$ for the STG-based group importances; and (iii) $C \times C$ transformation matrix Q . A heuristic for selecting the number of groups C is described in App. D.

Initialization. Gates are initialized to $\mu_j = 0.5$ (an unbiased prior) following (Yamada et al. 2020). We warm-start the logits using spectral clustering assignments based on L_{sym} (Von Luxburg 2007): for a feature i assigned to cluster j^* , we set

$$\log(\pi_{ij}) = \begin{cases} \Delta & \text{if } j = j^* \\ 0 & \text{otherwise} \end{cases}, \quad \Delta = \log\left(\frac{p_{\text{main}}}{p_{\text{rest}}}\right),$$

with $p_{\text{rest}} = \frac{1-p_{\text{main}}}{C-1}$ and $p_{\text{main}} = 0.7$ in all experiments. We initialize Q as a random orthonormal matrix and scale each row inversely to the feature-cluster sizes estimated from the spectral-clustering logits initialization, ensuring balanced influence on F . Note that these initial group assignments are not fixed; they are gradually overwritten during training as the model adapts the gates and groupings to minimize the total loss. Further information regarding hyperparameters is detailed in App. B.1.

5.2 Synthetic Data

We construct a 20-dimensional synthetic dataset by extending the classic *two-moons* dataset (Fig. 2A). Features 1-5 are noisy linear transformations of the moons’ first coordinate, and features 6-10 of the second, each generated as $x' = \sqrt{\rho}x + \sqrt{1-\rho}\epsilon$, where x is an original coordinate and $\epsilon \sim \mathcal{N}(0, 1)$. The remaining features (11-20) are i.i.d. Gaussian noise with zero mean and unit variance. The correlation strength $\rho \in [0.6, 1]$ controls how tightly each group follows its base coordinate. Figure 2B shows two examples for the sample correlation matrices (lower triangle) across features for $\rho=1.0$ and $\rho=0.6$ (a detailed description regarding the construction of the synthetic dataset is in App. A.1). The goal is to identify $G_1=\{1:5\}$ and $G_2=\{6:10\}$ as two separated groups, while the rest (11-20) are assigned to other groups, activate the gates attached to G_1 and G_2 , and deactivate the other gates attached to the rest. Unless noted otherwise, we use $\rho=0.95$, additive Gaussian noise with std. 0.05, 500 training epochs, batch size 100, and determine group count via the heuristic described in App. D.

Effect of correlation strength. Fig. 2C shows how the final loss varies with correlation strength ρ . For each ρ , we select the best model from a grid search over λ_1 and λ_2 , choosing the combination that yields the lowest final loss averaged over 10 runs with different seeds (details in App. B.1). As ρ increases, the loss decreases, reaching a minimum at $\rho=1$. This trend aligns with the more transparent block structure in the correlation matrices (Fig. 2B), confirming that GroupFS favors stronger intra-group coherence.

Effect of additive noise. In Fig. 2D, we plot the final loss as a function of the noise std. This curve remains essentially flat, indicating that increasing the standard deviation of the additive Gaussian noise up to 0.45 has little to no effect on the final loss. This suggests that GroupFS is robust to moderate sample-level noise, a desirable property in real-world applications where such noise is common.

Across all tested noise levels and ρ , the model isolates the informative groups $\{1:5\}$ and $\{6:10\}$ while the ten noisy dimensions are placed in unselected clusters.

Effect of feature and group numbers. We assess performance along two axes: (i) *feature grouping* and (ii) *feature selection*. In this controlled setting, the true group structure is known ($G_1=\{1:5\}$, $G_2=\{6:10\}$), enabling quantitative evaluation. We adapt the Group Similarity metric (Imrie et al. 2022) to compute *Relevant-Group Similarity* (\mathbf{RG}_{sim}). Let $\mathcal{G} = \{G_1, G_2\}$ denote the ground-truth groups and $\hat{\mathcal{G}} = \{\hat{G}_1, \dots, \hat{G}_C\}$ the predicted groups. We retain only predicted groups that overlap with at least one informative group: $\tilde{\mathcal{G}} = \{\hat{G}_j \mid \hat{G}_j \cap G_1 \neq \emptyset \text{ or } \hat{G}_j \cap G_2 \neq \emptyset\}$. Then we define

$$RG_{\text{sim}} = \frac{1}{\max(|\mathcal{G}|, |\tilde{\mathcal{G}}|)} \sum_{i=1}^2 \max_{\hat{G}_j \in \tilde{\mathcal{G}}} \mathcal{J}(G_i, \hat{G}_j),$$

where $\mathcal{J}(A, B) = |A \cap B|/|A \cup B|$ is the Jaccard index. This score lies in $[0, 1]$ and achieves a value of 1 only when both informative groups are perfectly recovered.

For feature selection, we report two common metrics. *True Positive Rate* (**TPR**) is the fraction of informative features $\{1:10\}$ that are selected (preferred: TPR=1). The *False Discovery Rate* (**FDR**) is the fraction of selected features from the noise set $\{11:20\}$ (preferred: FDR=0).

We vary the total number of features d (the last $d-10$ are nuisance) and the number of groups C , then evaluate the best-loss model over 10 runs with the three metrics (hyperparameters in App. B.1). Importantly, ground-truth groups are used only to compute evaluation metrics; the model is trained without access to this information. We retain the top-ranked groups by gate mean until at least 10 features are covered. The overall results are summarized in Fig. 3. We note that the effective number of groups that is required to fully separate signal and noise is: $C=2+(d-10)$, one group per informative feature cluster plus one per nuisance feature. Indeed, we observe that for $C \leq 2+(d-10)$, the model nearly always achieves $RG_{\text{sim}}=1$, TPR=1 and FDR=0, indicating it cleanly recovers both informative groups and assigns noise to separate groups. An exception is $C=2$: with too few groups, noise features are merged with informative ones, lowering RG_{sim} . TPR stays high, but FDR rises due to unwanted noise selection. When $C > 2+(d-10)$, performance remains strong in TPR and FDR, but RG_{sim} gradually declines. In this case, the model breaks informative clusters into smaller groups, a reasonable outcome given the surplus of available groups.

Overall, GroupFS performs well across a range of C values, as long as C is neither too small to separate informative from noisy features, nor too large to over-fragment the groups. In practice, setting C slightly above the expected number of informative groups is effective, especially for high-dimensional data.

5.3 Real-world Data

Our evaluation spans nine widely used datasets drawn from two domains (see Table 1 for sizes). The biomedical set: ALLAML (Golub et al. 1999), Lung500 (Lee et al. 2010), METABRIC (Pereira et al. 2016; Curtis et al. 2012), and HeartDisease (Janosi et al. 1989) contains gene-expression or clinical profiles. The vision set in-

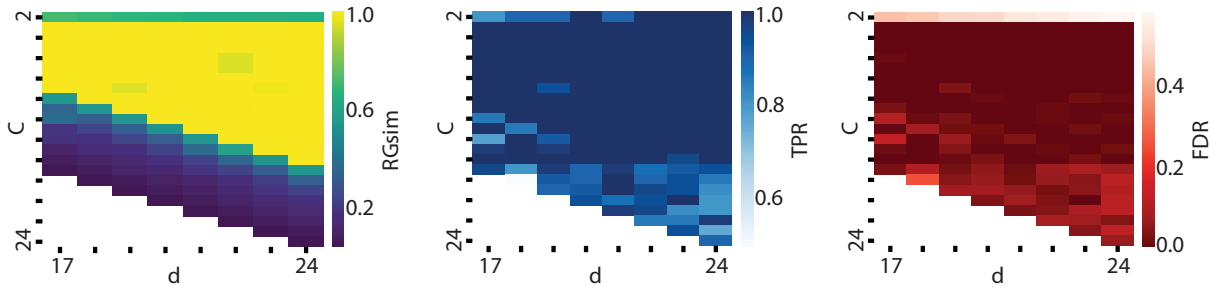


Figure 3: Two-moons: Effect of feature dimension d and group count C . Mean RG_{sim} TPR and FDR of the best-loss model over 10 random seeds. Complementary std results are in App. E.1.

Dataset	ALL	LS	MCFS	CAE	DUFS	MGAGR	CompFS	GroupFS	#Feat	Dim/Samp/Class
ALLAML	65.1±8.4	70.6±1.4	69.7±4.4	67.4±3.2	66.1±5.2	66.4±4.8	57.2±6.3	70.6±1.4	274	7129 / 72 / 2
Lung500	86.1±10.9	83.4±4.3	84.8±7.1	91.3±6.7	88.6±7.3	82.0±8.9	81.3±8.8	93.0±6.8	234	500 / 56 / 4
METABRIC	65.7±6.4	64.2±5.0	65.3±5.7	70.5±8.2	60.4±8.1	70.6±6.0	63.8±4.0	68.0±3.2	226	489 / 1904 / 2
HeartDisease	82.5±0.7	81.9±1.3	82.6±0.4	78.5±1.2	77.1±8.1	75.1±7.4	82.0±0.4	83.1±0.5	10	13 / 297 / 2
Yale	46.6±3.5	41.5±2.2	37.8±3.5	46.0±3.9	43.3±4.9	37.6±5.0	44.1±6.1	42.1±1.4	341	1024 / 165 / 15
AR10P	24.7±4.5	22.5±1.1	22.7±2.8	19.6±2.1	23.7±1.9	24.9±2.9	24.7±3.2	32.5±4.1	362	2400 / 130 / 10
PIE10P	29.0±2.7	22.8±1.0	34.0±2.0	24.4±1.5	35.0±3.6	34.4±2.8	29.0±1.8	38.4±2.5	49	2420 / 210 / 10
NMNIST 3-8	72.6±7.6	77.3±1.0	67.2±0.2	57.1±0.3	51.1±0.7	52.3±10.7	64.6±0.9	83.3±0.1	51	784 / 1000 / 2
NMNIST	49.5±2.7	46.1±2.4	48.9±6.9	44.1±3.0	20.1±5.9	–	45.9±1.4	48.9±2.7	184	784 / 12000 / 10

Table 1: Scenario 1 - Fixed budget, unsupervised setting. k -means accuracy (mean \pm std over 10 runs). All methods use the same feature budget (#Feat). The last column shows the original feature dimension, sample count, and number of classes. Bold marks the best score per dataset.

cludes Yale (Cai et al. 2007), AR10P (Martinez and Benavente 1998), PIE10P (Sim, Baker, and Bsat 2002), and two noisy MNIST variants (Larochelle et al. 2007; LeCun et al. 2002), all of which utilize image data. We adopt a random-background version of noisy MNIST (NMNIST), and a “3-8” subset comprising 500 images of digit 3 and 500 images of digit 8 (NMNIST 3-8). All datasets are z -scored feature-wise. Full download links and details appear in App. A.2. We compare GroupFS against a diverse set of FS methods:

- **LS** and **MCFS**: Classical graph-based selectors (He, Cai, and Niyogi 2005; Cai, Zhang, and He 2010).
- **CAE**: Concrete Autoencoder based on the Gumbel-Softmax relaxation (Abid, Balin, and Zou 2019).
- **DUFS**: A stochastic gating approach that learns a sample graph during training (Lindenbaum et al. 2021).
- **MGAGR**: A recent unsupervised method that leverages pre-defined feature groups (You et al. 2023).
- **CompFS**: A supervised baseline that jointly learns feature groups and a classifier (Imrie et al. 2022).
- **ALL**: A trivial baseline using all features.

We assess FS quality via k -means clustering, with k set to the number of ground-truth classes. To reduce sensitivity to initialization, we run k -means ten times with different seeds and report mean clustering accuracy \pm standard deviation. To ensure a fair comparison, we adapt each baseline as needed: Graph-based methods (LS, MCFS) use the self-tuning kernel from eq. (1) to avoid manual bandwidth

tuning. CAE and CompFS are trained with a 90/10 train-validation split, and the best model is chosen based on the lowest reconstruction loss (CAE) or the highest accuracy (CompFS) on the validation set. Because CompFS lacks a global feature budget, we aggregate all learned scores and retain the top-ranked features. MGAGR follows the authors’ recommended grouping but is skipped on NMNIST due to impractical runtime. For full hyperparameter settings and implementation details, see App. B.2.

Scenario 1: Fixed budget, unsupervised model choice (Table 1). All methods use the same feature budget as GroupFS (i.e., number of selected features). We determine this budget by gradually adding feature groups, ordered by the mean of their gates, and choosing the number of groups (and corresponding number of features) that yields a local maximum in model accuracy. For all models, hyperparameters are set based on the lowest loss, without supervision (i.e., without using labels), except for CompFS, which uses validation accuracy due to its supervised nature. Our GroupFS ranks first or tied for first on 6 out of 9 datasets, outperforming the next-best method by an average of +3.84%. On two of the remaining three datasets, GroupFS still ranks in the top three. Notably, Yale and NMNIST appear especially challenging for feature selection, as using all features yields the best results on both.

Scenario 2: Adaptive budget, accuracy-guided model choice (Table 2). We ran each baseline with multiple feature budgets: $\{50, 100, 200, 400\}$ (or $\{2, 4, 8, 10\}$ for the Heart Disease dataset) and reported the result with the highest k -

Dataset	LS	MCFS	CAE	DUFS	MGAGR	CompFS	GroupFS
ALLAML	72.2 (200)	71.8 (200)	70.4 (100)	71.5 (400)	66.3 (200)	67.4 (100)	72.8 (302)
Lung500	82.2 (200)	87.5 (200)	92.3 (400)	95.0 (50)	93.0 (100)	94.5 (100)	96.1 (361)
METABRIC	68.0 (200)	69.3 (100)	72.8 (50)	72.4 (100)	71.6 (200)	73.2 (50)	73.4 (159)
HeartDisease	81.9 (10)	82.7 (8)	82.8 (8)	83.5 (8)	84.3 (10)	84.3 (8)	83.1 (10)
Yale	43.0 (400)	45.0 (100)	45.6 (400)	42.6 (200)	40.7 (100)	46.7 (50)	48.3 (398)
AR10P	32.9 (50)	29.5 (100)	30.1 (100)	34.2 (200)	32.6 (50)	29.8 (50)	34.7 (363)
PIE10P	26.3 (400)	34.1 (100)	26.6 (400)	42.1 (50)	36.0 (50)	29.1 (50)	40.8 (370)
NMNIST 3-8	76.6 (200)	68.1 (400)	76.0 (400)	80.2 (200)	77.4 (400)	80.4 (100)	84.1 (288)
NMNIST	49.4 (400)	50.4 (200)	50.5 (400)	42.0 (400)	–	55.0 (200)	48.9 (184)

Table 2: Scenario 2 - Adaptive budget, accuracy-guided setting. k -means accuracy (mean over 10 runs). Each method selects its own feature budget (numbers in parentheses). Bold marks the best score per dataset. See App. E.2 for mean \pm std results.

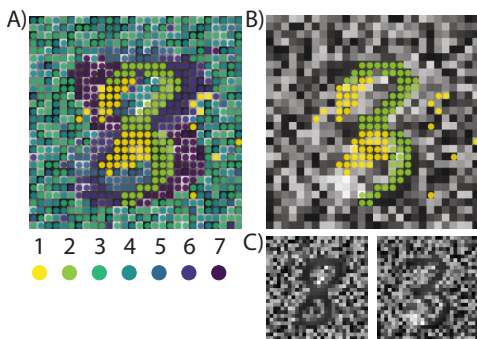


Figure 4: GroupFS on MNIST (3 vs. 8). (A) Pixel groups discovered by GroupFS, colored by group ID and ranked by importance (1 = highest, 7 = lowest). (B) The top two groups align with class-relevant regions. (C) Noisy image examples of digits ‘8’ and ‘3’.

means accuracy. Since GroupFS naturally outputs variable-sized groups, we retain the parameter choice that achieves the best accuracy while using no more than 400 features (or 10 for the Heart Disease dataset). In this setting, GroupFS achieves the highest accuracy on 6 out of 9 datasets, demonstrating strong performance even when methods are allowed to adapt their feature count.

5.4 Interpretability Experiment

We evaluate the interpretability benefits of GroupFS on two datasets from distinct domains: vision and education. In both cases, our goal is to highlight how unsupervised group discovery reveals meaningful feature groups aligned with domain knowledge. The first is the MNIST 3-8 subset, and the second is the UCI Student Performance dataset (Cortez 2008), with 395 samples and 30 features covering academic, demographic, and behavioral variables. For the UCI dataset, we focus on predicting math exam pass/fail outcomes (a detailed description of this dataset is App. A.2).

MNIST 3-8. GroupFS discovers seven spatially coherent pixel groups, visualized in Figure 4. Pixels are colored by group, with the legend ranking importance from 1 (most important) to 7 (least). Although learned with no supervision, group 1 (yellow) highlights regions that differentiate 3s from 8s, such as the upper-left loop of 8s, which is typi-

cally absent in 3s. Lower-ranked groups correspond mainly to background pixels with little discriminative value. This shows that GroupFS segments the image into functionally meaningful regions that are spatially localized and consistent across the dataset.

Student Performance. For this tabular dataset, GroupFS discovers seven interpretable feature groups. Taking the top three groups leads to clustering accuracy of $61.3 \pm 2.6\%$. The highest-ranked group includes features related to alcohol consumption (daily and weekly). The second group includes features associated with motivation, such as the number of school absences, past academic failures, romantic relationships, and intention to pursue higher education. The third group relates to parents, including their education and the mother’s job. Each group displays strong semantic coherence, reinforcing the idea that GroupFS can uncover meaningful structure even in unlabeled tabular data. A full breakdown of feature rankings for this and other baselines appears in App. E.3.

6 Conclusion

We address unsupervised group feature selection, which involves the discovery of informative groups of features in an entirely data-driven manner, without relying on labels or prior knowledge. We introduce **GroupFS**, an end-to-end differentiable framework that simultaneously learns group assignments and sparsely selects them. Across a variety of image, tabular, and biomedical datasets, GroupFS matches or surpasses state-of-the-art unsupervised feature-selection methods on downstream clustering tasks under both fixed and variable feature budgets. Qualitative inspection further shows that the discovered groups align with meaningful domain structure. GroupFS has limitations: the sample and feature graphs rely on Euclidean distances, which can misrepresent data on curved, non-Euclidean manifolds. It also learns a single global notion of group importance, overlooking condition, or time-dependent relevance, where both groups and their importance may evolve. Future work will include smooth, differentiable manifold-aware distances and a dynamic formulation with condition-adaptive grouping and ranking. Overall, GroupFS provides a practical step toward combining feature-structure discovery with sparse selection in a purely unsupervised setting, and it can be used modularly as a building block for downstream learning tasks.

Acknowledgements

This research was partially supported by the NSF (CCF-2403452), by the ISF (2418/24) and (1693/22), and by the Skillman chair (RM). OL was supported by the MOST grant No. 0007341.

References

- Abid, A.; Balin, M. F.; and Zou, J. 2019. Concrete autoencoders for differentiable feature selection and reconstruction. *arXiv preprint arXiv:1901.09346*.
- Battash, B.; Wolf, L.; and Lindenbaum, O. 2024. Revisiting the noise model of stochastic gradient descent. In *International Conference on Artificial Intelligence and Statistics*, 4780–4788. PMLR.
- Battiti, R. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4): 537–550.
- Cai, D.; He, X.; Hu, Y.; Han, J.; and Huang, T. 2007. Learning a spatially smooth subspace for face recognition. In *2007 IEEE conference on computer vision and pattern recognition*, 1–7. IEEE.
- Cai, D.; Zhang, C.; and He, X. 2010. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 333–342.
- Cortez, P. 2008. Student Performance. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5TG7T>.
- Curtis, C.; Shah, S. P.; Chin, S.-F.; Turashvili, G.; Rueda, O. M.; Dunning, M. J.; Speed, D.; Lynch, A. G.; Samarajiwa, S.; Yuan, Y.; et al. 2012. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403): 346–352.
- Donoho, D. L.; et al. 2000. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000): 32.
- Fan, J.; and Lv, J. 2010. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1): 101.
- Golub, T. R.; Slonim, D. K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J. P.; Coller, H.; Loh, M. L.; Downing, J. R.; Caligiuri, M. A.; et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439): 531–537.
- Guyon, I.; and Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar): 1157–1182.
- He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. *Advances in neural information processing systems*, 18.
- Imrie, F.; Norcliffe, A.; Liò, P.; and van der Schaar, M. 2022. Composite feature selection using deep ensembles. *Advances in Neural Information Processing Systems*, 35: 36142–36160.
- Jana, S.; Li, H.; Yamada, Y.; and Lindenbaum, O. 2023. Support recovery with projected stochastic gates: Theory and application for linear models. *Signal Processing*, 213: 109193.
- Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.
- Janosi, A.; Steinbrunn, W.; Pfisterer, M.; and Detrano, R. 1989. Heart Disease. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C52P4X>.
- Kingma, D. P.; Welling, M.; et al. 2013. Auto-encoding variational bayes.
- Kohavi, R.; and John, G. H. 1997. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2): 273–324.
- Larochelle, H.; Erhan, D.; Courville, A.; Bergstra, J.; and Bengio, Y. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, 473–480.
- LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 2002. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.
- Lee, M.; Shen, H.; Huang, J. Z.; and Marron, J. S. 2010. Biclustering via sparse singular value decomposition. *Bio-metrics*, 66(4): 1087–1095.
- Li, G.; Yu, Z.; Yang, K.; Lin, M.; and Chen, C. P. 2024. Exploring feature selection with limited labels: A comprehensive survey of semi-supervised and unsupervised approaches. *IEEE Transactions on Knowledge and Data Engineering*.
- Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6): 1–45.
- Lindenbaum, O.; Shaham, U.; Peterfreund, E.; Svirsky, J.; Casey, N.; and Kluger, Y. 2021. Differentiable unsupervised feature selection based on a gated laplacian. *Advances in neural information processing systems*, 34: 1530–1542.
- Luo, H.; Xu, C.; Ji, Z.; and Yuan, A. 2024. Adaptive Graph Learning for Multi-View Unsupervised Feature Selection. In *2024 8th Asian Conference on Artificial Intelligence Technology (ACAIT)*, 255–262. IEEE.
- Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.
- Martinez, A.; and Benavente, R. 1998. The ar face database: Cvc technical report, 24. *CVC Technical Report*.
- Miao, J.; Yang, T.; Sun, L.; Fei, X.; Niu, L.; and Shi, Y. 2022. Graph regularized locally linear embedding for unsupervised feature selection. *Pattern Recognition*, 122: 108299.
- Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.
- Park, H.; and Lee, C. 2024. Feature Selection With Group-Sparse Stochastic Gates. *IEEE Access*.
- Pereira, B.; Chin, S.-F.; Rueda, O. M.; Vollan, H.-K. M.; Provenzano, E.; Bardwell, H. A.; Pugh, M.; Jones, L.; Russell, R.; Sammut, S.-J.; et al. 2016. The somatic mutation

- profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature communications*, 7(1): 11479.
- Perera, K.; Chan, J.; and Karunasekera, S. 2020. Group Based Unsupervised Feature Selection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 805–817. Springer.
- Sim, T.; Baker, S.; and Bsat, M. 2002. The CMU pose, illumination, and expression (PIE) database. In *Proceedings of fifth IEEE international conference on automatic face gesture recognition*, 53–58. IEEE.
- Solorio-Fernández, S.; Carrasco-Ochoa, J. A.; and Martínez-Trinidad, J. F. 2020. A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2): 907–948.
- Spielman, D. A. 2025. Spectral and Algebraic Graph Theory. Lecture notes (incomplete draft). Available at <http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf>.
- Sristi, R. D.; Lindenbaum, O.; Lifshitz, S.; Lavzin, M.; Schiller, J.; Mishne, G.; and Benisty, H. 2023. Contextual feature selection with conditional stochastic gates. *arXiv preprint arXiv:2312.14254*.
- Sristi, R. D.; Mishne, G.; and Jaffe, A. 2022. Disc: Differential spectral clustering of features. *Advances in Neural Information Processing Systems*, 35: 26269–26282.
- Svirsky, J.; and Lindenbaum, O. 2024. Interpretable Deep Clustering for Tabular Data. In *International Conference on Machine Learning*, 47314–47330. PMLR.
- Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1): 267–288.
- Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing*, 17: 395–416.
- Wang, S.; Wang, Y.; Tang, J.; Aggarwal, C.; Ranganath, S.; and Liu, H. 2017. Exploiting hierarchical structures for unsupervised feature selection. In *Proceedings of the 2017 siam international conference on data mining*, 507–515. SIAM.
- Yamada, Y.; Lindenbaum, O.; Negahban, S.; and Kluger, Y. 2020. Feature selection using stochastic gates. In *International conference on machine learning*, 10648–10659. PMLR.
- You, M.; Yuan, A.; Zou, M.; He, D.; and Li, X. 2023. Robust Unsupervised Feature Selection via Multi-Group Adaptive Graph Representation. *IEEE Transactions on Knowledge and Data Engineering*, 35(3): 3030–3044.
- Zaharieva, M.; Breiteneder, C.; and Hudec, M. 2017. Unsupervised group feature selection for media classification. *International Journal of Multimedia Information Retrieval*, 6: 233–249.
- Zelnik-Manor, L.; and Perona, P. 2004. Self-tuning spectral clustering. *Advances in neural information processing systems*, 17.