Multi-modal Differentiable Unsupervised Feature Selection (Supplementary Material)

Junchen Yang[1] **Ofir Lindenbaum**[2] **Yuval Kluger**[1,4,5] **Ariel Jaffe**[3]

[1]Interdepartmental Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT, USA
[2]Faculty of Engineering, Bar-Ilan University, Israel
[3]Department of Statistics and Data Science, Hebrew University of Jerusalem, Israel
[4]Applied Math Program, Yale University, New Haven, CT, USA
[5]Department of Pathology, School of Medicine, Yale University, New Haven, CT, USA

A ADDITIONAL RESULTS

A.1 POINTS IN A 3D CUBE.

The data consists of points in a 3D cube $[0, l_s] \times [0, l_a] \times [0, l_b]$. The modality $\boldsymbol{X}$ includes the first two coordinates, and modality $\boldsymbol{Y}$ includes the first and third, as explained in Sec. 3. The upper row in Figure 1 shows the eigenvectors of $\boldsymbol{L}_x$. The eigenvectors change in both coordinates. The second row contain the eigenvectors of $\boldsymbol{P}_{\text{shared}}$. the leading eigenvectors change only with the first coordinate, as it is the only shared variable.

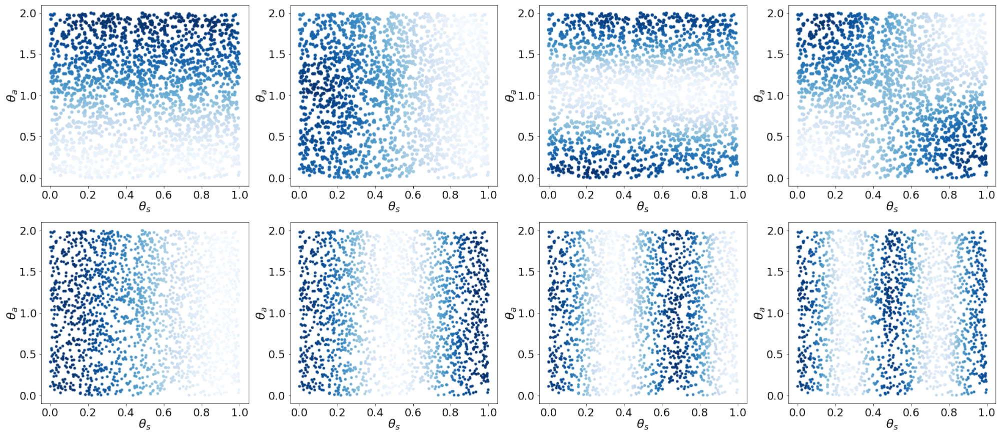


Figure 1: Data consists of points sampled uniformly at random in a 3D cube. The upper row shows a scatter plot of the points, located according to the first two coordinates a, b and colored by the leading eigenvectors of $\boldsymbol{L}_x$, the Laplacian matrix of modality $\boldsymbol{X}$. The bottom row shows the leading eigenvectors of $\boldsymbol{P}_{\text{shared}}$, the product of Laplacians as defined in Eq. 6.

A.2 RESCALED MNIST.

Here in Table 1, we compare mmDUFS to the baselines on the rescaled MNIST data with 3 modalities. We can see that mmDUFS outperforms all the baselines in terms of the F1-score, demonstrating its ability to identify informative features in multimodal scenarios accurately.

Modality	MC	mmKS	mmKP	mmDUFS
X	0.4012	0.6163	0.6163	**0.7035**
Y	0.5672	0.7562	0.7612	**0.8259**
Z	0.5333	0.7385	0.7385	**0.8154**

Table 1: F1-score of different methods on the rescaled MNIST data with 3 modalities

A.3 ROTATING DOLLS.

The two modalities include video frames taken simultaneously from two cameras, of three dolls rotating at different angular speeds. The first camera (modality $\boldsymbol{X}$) captures the left two dolls while the right camera (modality $\boldsymbol{Y}$) captures the right two dolls. Thus, the angle of the middle doll constitutes a shared variable $\boldsymbol{\theta}_s$. The angle of the left doll $\boldsymbol{\theta}_x$ is modality $\boldsymbol{X}$-specific latent variable, and the angle of the right doll $\boldsymbol{\theta}_y$ is modality $\boldsymbol{Y}$-specific latent variable.

From the left video, we cut the frames such that it includes only the middle doll (the shared component). From these images we computed a graph Laplacian matrix and its leading eigenvectors denoted $\boldsymbol{\phi}_i^s$. As explained in Sec. 3, we expect the eigenvectors of the shared operator, denoted $\boldsymbol{v}_i^s$ to be similar to $\boldsymbol{\phi}_i^s$, as both are associated with the latent variable $\boldsymbol{\theta}_s$. Figure 2 shows $\boldsymbol{v}_i^s$ as a function of $\boldsymbol{\phi}_i^s$ for $i = 1, 2, 3$. The three vectors are clearly highly correlated.

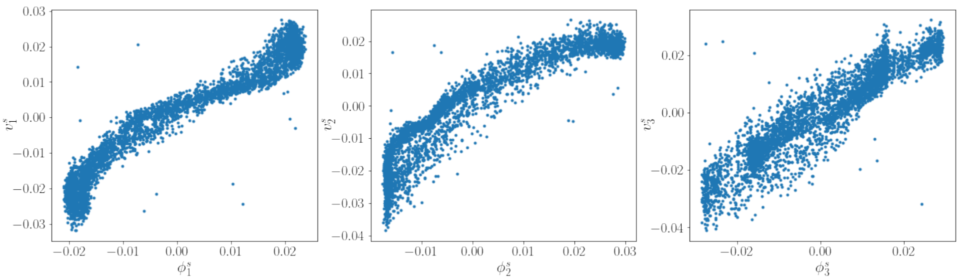


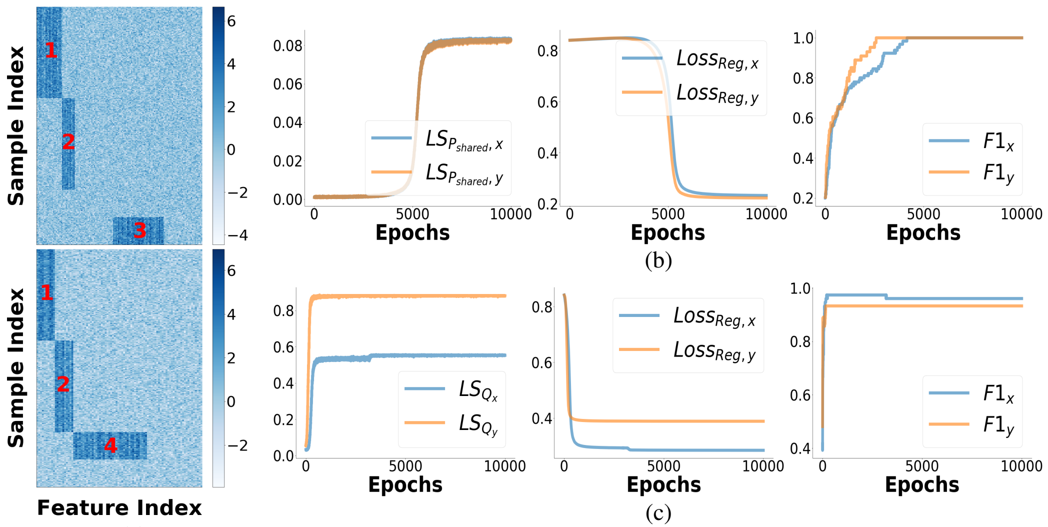
Figure 2: The figure shows a scatter plot of $\boldsymbol{v}_i^s$, the leading eigenvectors of $\boldsymbol{P}_{\text{shared}}$ as a function of $\boldsymbol{\phi}_i^s$, the estimated leading vectors of the shared component in the rotating doll dataset.

A.4 CITE-SEQ DATASET.

To demonstrate the feature selection performance of mmDUFS on the shared structures, we focus on the CITE-seq data and analyze four cell types: B cells, CD8 T cells, CD16+ Monocytes, and Naive CD4 T cells. This subset has $2,101$ cells for both RNA and protein modalities. We select the top 500 variable genes as the informative features in the RNA modality and add $1,500$ nuisance features generated according to a Gaussian distribution. Then, we apply different baseline methods to select the informative features in the RNA modality and compare their performance using F1-score. As shown in Table 2, mmDUFS outperforms other baseline methods in terms of selecting the correct informative features.

	MC	mmKS	mmKP	mmDUFS
F1-score	0	0.664	0.778	**0.808**

Table 2: Comparison of F1-score between different methods on the CITE-seq data (RNA modality)



(a)

(b)

(c)

Figure 3: Synthetic Gaussian mixture cluster example. (a): Data matrix of modality $\boldsymbol{X}$ (top) and $\boldsymbol{Y}$ (bottom). Rows are samples, and columns are features. Each modality has 3 clusters (labeled in red). Clusters 1 and 2 are shared between modalities, and cluster 3 and 4 are specific to each modality. (b): Change of the Shared Laplacian Scores, regularization loss, and the F1-score of the selected features concerning the number of epochs (x-axis) for mmDUFS with the shared operator. (c): Change of the Differential Laplacian Scores, regularization loss, and the F1-score of the selected features concerning the number of epochs (x-axis) for mmDUFS with the differential operator.

A.5 SYNTHETIC GAUSSIAN MIXTURES.

Here we apply mmDUFS to uncover the informative features of the shared clusters and the modality-specific clusters. Fig. 3b and Fig. 3c show the change of the average Shared/Differential Laplacian Scores across features, the regularization loss, and the F1-score of the selected features from mmDUFS with respect to the number of epochs, where we can see that mmDUFS gradually selects the correct features corresponding to high scores while sparsifying the number of features.

We also apply DUFS to each modality on this data and compare its performance to mmDUFS in terms of F1-score, as shown below in Table 3.

Dataset	Modality	DUFS	mmDUFS
Original Gaussian	X	0.300	**1**
	Y	0	**1**
Gaussian + 10 Noisy Feats	X	0.2667	**1**
	Y	0	**1**
Gaussian + 30 Noisy Feats	X	0.100	**1**
	Y	0	**1**
Gaussian + 50 Noisy Feats	X	0.033	**0.9667**
	Y	0	**0.8500**

Table 3: Comparison of F1-score on the synthetic Gaussian mixture data between DUFS and mmDUFS

DUFS is suboptimal for this task because it recovers the most informative features in a single modality. It does not, however, distinguish between modality-specific and modality-shared features.

B EXPERIMENT DETAILS

In the following subsections, we provide additional experimental details required for the reproduction of the experiments provided in the main text. The CPU model used for the experiments is Intel(R) Xeon(R) Gold 6150 CPU @ 2.70GHz (72 cores total). GPU model is NVIDIA GeForce RTX 2080 Ti.

Below in Table 4 and 5, we list the parameters we used on each experiment for mmDUFS with the shared operator and the differential operator. Paramter c is a regularization constant for mmDUFS with the differential operator, as mentioned in the main text. Parameter b is a scaling factor to the operators to balance between the Shared/Differential Laplacian Scores with respect to the regularization term. We used normalized Laplacian Matrix throughout the experiments except for the CITE-seq example where we found the performance was satisfactory with the un-normalized Laplacian Matrix.

Datasets	learning rate	epochs	λ_x	λ_y	b
Rescaled MNIST	2	10000	$1e-1$	$1e-1$	$1e2$
Synthetic Tree	2	25000	$1e-1$	$1e-1$	$1e3$
Gaussian Mixture	2	10000	$1e-4$	$1e-4$	1
Gaussian Mixture (10 Noisy Features)	2	20000	$1e-8$	$1e-6$	1
Gaussian Mixture (30 Noisy Features)	2	40000	$1e-4$	$1e-4$	1
Gaussian Mixture (50 Noisy Features)	2	10000	$1e-2$	$1e-3$	$1e2$
Rotating Dolls	2	10000	0.2	0.2	$1e3$

Table 4: Parameters for mmDUFS with the shared operator across different datasets.

Datasets	learning rate	epochs	λ_x	λ_y	c	b
Rescaled MNIST	1	10000	0.5	0.5	$1e-3$	$1e-4$
Synthetic Tree	2	10000	4	2	$1e-3$	$1e-3$
Gaussian Mixture	1	10000	0.4	0.4	$1e-1$	$1e-1$
Rotating Dolls	2	10000	2	2	3	$1e3$
CITE-seq	2	5000	3		2	1

Table 5: Parameters for mmDUFS with the differential operator across different datasets.

For the baseline methods, k features with the highest Laplacian Scores are selected. When evaluating f1-score on the synthetic datasets, we set k to be the correct number of informative features. To make a fair comparison, we also let mmDUFS to select k features by sorting the raw gates (μ_d for feature d). For other datasets, we define selected features by mmDUFS as features whose gates converged to 1 ($z_d = 1$ for feature d).

For the image datasets (rescaled MNIST, rotating dolls), we add small Gaussian noise drawn from $N(0, \sigma^2)$ to the pixels to stabilize feature selection of mmDUFS. For the rescaled MNIST dataset, $\sigma = 0.1$ and we add noise to the non-informative pixels before standardizing the pixels via z-scoring. For the rotating dolls data, $\sigma = 5e-3$ and we add noise to all pixels before standardizing the pixels via z-scoring.

B.1 TUNING OF THE REGULARIZATION PARAMETER

mmDUFS has tunable regularization parameters λ_x and λ_y that control the sparsity of the number of selected features. For synthetic datasets, one can tune these parameters to select features such that the selected number is close to the prescribed number s. However, it can still be time and resource consuming to optimize these parameters. Also, for real data, one might not know how many features to select and what λ_x and λ_y to choose.

To alleviate this issue, we propose a "warm-up" procedure similar to [Lindenbaum et al., 2021] to optimize λ_x and λ_y. Specifically, we evaluate the mean Shared Laplacian Scores $S_{\text{shared}} = \frac{1}{2n}(\text{Tr}[\tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{P}}_{\text{shared}} \tilde{\boldsymbol{X}}]/m + \text{Tr}[\tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{P}}_{\text{shared}} \tilde{\boldsymbol{Y}}]/d)$ and the mean Differential Laplacian Scores $S_{\text{x}} = \text{Tr}[\tilde{\boldsymbol{X}}^T \boldsymbol{Q}_{\tilde{x}} \tilde{\boldsymbol{X}}]/(d \times n)$, $S_{\text{y}} = \text{Tr}[\tilde{\boldsymbol{Y}}^T \boldsymbol{Q}_{\tilde{y}} \tilde{\boldsymbol{Y}}]/(m \times n)$ over a grid of λ_x and λ_y at the early stage of training (e.g., first 1000 epochs), and pick the parameters that maximize the Scores. Here n is the number of samples in the batch, and m and d are the number of selected features on each modality for real data, or the number of pre-specified features for synthetic data.

To demonstrate this procedure, we use the synthetic Gaussian mixture dataset as the example, and we evaluate λ_x and λ_y over $\{1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2\}$ using mmDUFS with the shared operator. For illustration purpose, we set $\lambda_x = \lambda_y$ Fig. 4 shows the mean Shared Laplacian Scores over different λ values. We can see that $\{1e-6, 1e-5, 1e-4, 1e-3\}$ are the best candidates that give the highest Shared Laplacian Scores that also correspond to the highest F1-score.

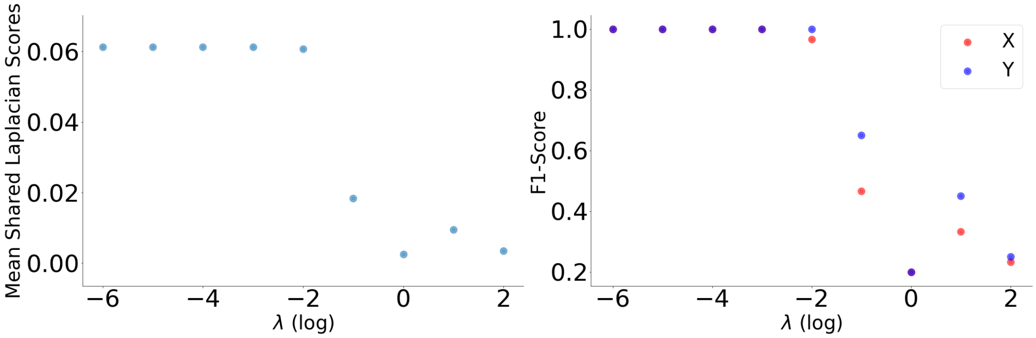


Figure 4: Evaluation of the mean Shared Laplacian Scores (left) and the corresponding F1-scores (right) over a grid of λs on the synthetic Gaussian mixture dataset. y-axis shows the mean Shared Laplacian Scores (left) and F1-scores (right) whereas the x-axis shows the values of λ.

B.2 SYNTHETIC GAUSSIAN MIXTURES

We simulate 2 modalities $\boldsymbol{X}$ and $\boldsymbol{Y}$, where modality $\boldsymbol{X}$ has 260 samples with 130 features and modality $\boldsymbol{Y}$ has 260 samples with 90 features. Both modalities have 3 clusters in the data ($\boldsymbol{X}$ has cluster 1, 2, 3 and $\boldsymbol{Y}$ has cluster 1, 2, 4, all labeled in red in Fig. 3a), and each cluster has a set of informative features denoted as $\boldsymbol{f}_{x,i}$ and $\boldsymbol{f}_{x,i}$ (i = 1, 2, 3, 4) with length m_i (i = 1, 2, 3, 4). Each set of these informative features is drawn from $N(\boldsymbol{\mu_i}, \boldsymbol{I})$ independently for each sample, where $\boldsymbol{\mu_i}$ is a vector of length m_i drawn from $U(2, 4)$ and $\boldsymbol{I}$ is an $m_i \times m_i$ identity matrix.

By design, cluster 1 and 2 are shared between modalities with $m_1 = 20$ and $m_2 = 10$ in modality $\boldsymbol{X}$, and $m_1 = 10$ and $m_2 = 10$ in modality $\boldsymbol{Y}$. On the other hand, cluster 3 is specific to modality $\boldsymbol{X}$ with $m_3 = 40$, and cluster 4 is specific to modality $\boldsymbol{Y}$ with $m_4 = 40$. The remaining features are considered noisy features and are drawn from $N(0, 1)$.

B.3 SYNTHETIC DEVELOPMENTAL TREE

We use *generate_data()* function from dyntoy [1],a tree simulator package, to generate a dataset $\boldsymbol{X}_0$ with 1000 samples and 100 features. Specifically, the parameter *num_branchpoints* is set to 1, *num_cells* is set to 1000, *num_features* is set to 100, *sample_mean_count* is set to 10, *sample_dispersion_count* is set to 50, *differentailly_expressed_rate* is set to 4, and *dropout_probability_factor* is set to 0.

This step yields an initial data matrix $\boldsymbol{X}_0 \in \mathbb{R}^{1000\times100}$, and these 1000 samples are initially partitioned into 4 groups: G_1 and G_2, G_3 and G_4, G_5, G_6 shown in Fig. 3c. For $\boldsymbol{X}_0$, we further divide it into two halves, resulting in 2 data matrices $\boldsymbol{X} \in \mathbb{R}^{1000\times50}$ and $\boldsymbol{Y} \in \mathbb{R}^{1000\times50}$. We regard $\boldsymbol{X}$ and $\boldsymbol{Y}$ as 2 data modalities and these features as informative features contributing to the shared tree structure.

We further add 50 features to each modality that are drawn from negative binomial distributions to construct the differential structures between modalities. Specifically, for modality $\boldsymbol{X}$, the 50 features of G_1 are drawn from $NB(\mu = 4, \alpha = 0.1)$ where μ and α are the mean and dispersion parameter of the negative binomial distribution, whereas the 50 features of the other groups of samples are drawn from $NB(\mu = 20, \alpha = 0.1)$. Similarly, for modality $\boldsymbol{Y}$, the 50 features of G_3 are drawn from $NB(\mu = 4, \alpha = 0.1)$ while the 50 features of the other groups of samples are drawn from $NB(\mu = 20, \alpha = 0.1)$. Therefore, G_1 is bifurcated from G_2 and this structure is only observed in $\boldsymbol{X}$, and G_3 is bifurcated from G_4 and this structure is only observed in $\boldsymbol{Y}$.

Next, we row normalize each data matrix multipled by a scaling factor $1e4$, and log1p transform the data. Then we standardize the features by z-scoring. At the end, we add 200 features drawn from $N(0, 1)$ to each modality as the noisy features.

[1] https://github.com/dynverse/dyntoy

B.4 CITE-SEQ

The human cord blood mononuclear cells (CBMCs) CITE-seq data was generated by [Stoeckius et al., 2017], where the expression levels of both RNA and protein are measured for the same cells. We analyze 3 cell types: Erythoid cells, CD 34+ cells, and Murine cells. We row normalize each data matrix for both modalities. For the gene expression matrix (RNA), we filter the genes by standard deviation and keep the top 500 variable genes. Then for both matrices, we standardize the features by z-scoring.