A. Appendices

A.1. Details of ADMM Solution

We use the Alternating Direction Method of Multipliers (ADMM) [4]. First, the problem (1) is split as follows:

$$\min_{L,S,W} \|L\|_* + \lambda\|S\|_1 + \gamma\,\mathrm{tr}(W\Phi W^T)$$
$$\text{s.t. } X = L + S,\ L = W.$$

Second, the augmented Lagrangian and iterative scheme are introduced:

$$\begin{aligned}(L^{k+1}, S^{k+1}, W^{k+1}) = \underset{L,S,W}{\operatorname{argmin}} \|L\|_* + \lambda\|S\|_1 + \gamma\,\mathrm{tr}(W\Phi W^T) + \langle Z_1^k, X - L - S\rangle + \frac{r_1}{2}\|X - L - S\|_F^2 \\ + \langle Z_2^k, W - L\rangle + \frac{r_2}{2}\|W - L\|_F^2, \quad (2)\end{aligned}$$
$$Z_1^{k+1} = Z_1^k + r_1(X - L^{k+1} - S^{k+1}), \quad (3)$$
$$Z_2^{k+1} = Z_2^k + r_2(W^{k+1} - L^{k+1}), \quad (4)$$

where Z_1 and Z_2 are the lagrange multipliers (dual variables) and k is the iteration index.

In order to update the primal variables L, S and W we consider the definition of proximity operator [7]. Let $f \in \Gamma_0(\mathbb{R}^N)$, where $\Gamma_0(\mathbb{R}^N)$ is the class of lower semi-continuous convex functions from $\mathbb{R}^N$ to $]-\infty, +\infty]$ such that domain of f $\neq \varnothing$. Then, for every $x \in \mathbb{R}^N$, the following minimization problem admits a unique solution which is known as the **proximity operator of f** denoted by prox_f.

$$\mathrm{prox}_f(x) = \underset{y\in\mathbb{R}^N}{\operatorname{argmin}}\, f(y) + \frac{1}{2}\|x - y\|_2^2.$$

Using this definition, the updates for L, S and W at $(k+1)^{st}$ iteration using the previous iterates $L^k, S^k, W^k, Z_1^k, Z_2^k$ can be made using the proximity operators.

Update L

Keeping only the terms with L in Eq. (2) we get

$$\begin{aligned}L^{k+1} &= \underset{L}{\operatorname{argmin}} \|L\|_* + \langle Z_1^k, X - L - S^k\rangle + \frac{r_1}{2}\|X - L - S^k\|_F^2 + \langle Z_2^k, W^k - L\rangle + \frac{r_2}{2}\|W^k - L\|_F^2 \\ &= \underset{L}{\operatorname{argmin}} \|L\|_* + \frac{r_1}{2}\left\|L - \left(X - S^k + \frac{Z_1^k}{r_1}\right)\right\|_F^2 + \frac{r_2}{2}\left\|L - \left(W^k + \frac{Z_2^k}{r_2}\right)\right\|_F^2 \\ &= \underset{L}{\operatorname{argmin}} \|L\|_* + \frac{r_1 + r_2}{2}\left\|L - \frac{r_1H_1^k + r_2H_2^k}{r_1 + r_2}\right\|_F^2 \\ &= \mathrm{prox}_{\frac{1}{(r_1+r_2)}\|L\|_*}\left(\frac{r_1H_1^k + r_2H_2^k}{r_1 + r_2}\right),\end{aligned}$$

where $H_1^k = X - S^k + Z_1^k/r_1$ and $H_2^k = W^k + Z_2^k/r_2$. Let $\Omega_\tau : \mathbb{R}^N \longrightarrow \mathbb{R}^N$ denote the element-wise soft-thresholding operator $\Omega_\tau(x) = \mathrm{sgn}(x)\max(|x| - \tau, 0)$, then we can define $D_\tau(A) = P\Omega_\tau(\Sigma)Q^T$ as the singular value thresholding operator for matrix A, where $A = P\Sigma Q^T$ is any singular value decomposition of A. Let $A = \frac{r_1H_1^k + r_2H_2^k}{r_1+r_2}$ and $r = (r_1+r_2)/2$ then

$$L^{k+1} = D_{\frac{1}{r}}(A) = P\Omega_{\frac{1}{r}}(\Sigma)Q^T. \quad (5)$$

Update S

Following a similar procedure, we can write the update for S as.

$$\begin{aligned}S^{k+1} &= \mathrm{prox}_{\frac{\lambda}{r_1}\|S\|_1}\left(X - L^{k+1} + \frac{Z_1^k}{r_1}\right) \\ &= \Omega_{\frac{\lambda}{r_1}}\left(X - L^{k+1} + \frac{Z_1^k}{r_1}\right) \quad (6)\end{aligned}$$

Update W

By keeping only the terms with W in eq. (2) we get

$$W^{k+1} = \underset{W}{\operatorname{argmin}}\, \gamma \operatorname{tr}(W\Phi W^T) + \frac{r_2}{2}\left\|W - \left(L^{k+1} - \frac{Z_2^k}{2}\right)\right\|_F^2$$

which is a smooth function in W, so we can use the optimality condition to find a closed form solution for W.

$$W^{k+1} = r_2(\gamma\Phi + r_2 I)^{-1}\left(L^{k+1} - \frac{Z_2^k}{r_2}\right) \tag{7}$$

Projected Conjugate Gradient method was used to update W.

A.2. Algorithm

Algorithm 1 ADMM algorithm for Robust PCA on Graphs

procedure ROBUST PCA GRAPHS($X \in \mathbb{R}^{p\times n}, \Phi \in \mathbb{S}_+^n, \lambda, \gamma$) ▷ inputs
  $k \leftarrow 0$ ▷ iteration index k
  $L^k \leftarrow rand(n,p)$, $W^k \leftarrow rand(n,p)$, $S^k \leftarrow rand(n,p)$ ▷ Initialize primal variables
  $r_1 \leftarrow 1$, $r_2 \leftarrow 1$
  $Z_1^k \leftarrow X - L^k - S^k$, $Z_2^k \leftarrow W^k - L^k$ ▷ Initialize dual variables
  $P_1^k \leftarrow \|L^k\|_*$, $P_2^k \leftarrow \lambda\|S^k\|_1$, $P_3^k \leftarrow \gamma \operatorname{tr}(L^k\Phi {L^k}^T)$ ▷ Initialize primal objective
  while $\frac{\|P_1^{k+1}-P_1^k\|_F^2}{\|P_1^k\|_F^2} > \epsilon$ & $\frac{\|P_2^{k+1}-P_2^k\|_F^2}{\|P_2^k\|_F^2} > \epsilon$ & $\frac{\|P_3^{k+1}-P_3^k\|_F^2}{\|P_3^k\|_F^2} > \epsilon$ & $\frac{\|Z_1^{k+1}-Z_1^k\|_F^2}{\|Z_1^k\|_F^2} > \epsilon$ & $\frac{\|Z_2^{k+1}-Z_2^k\|_F^2}{\|Z_2^k\|_F^2} > \epsilon$ **do**
    Update L^{k+1} using eq. 5
    Update S^{k+1} using eq. 6
    Update W^{k+1} using eq. 7
    Update Z_1^{k+1} using eq. 2
    Update Z_2^{k+1} using eq. 3
    Update P_1^{k+1}, P_2^{k+1} and P_3^{k+1} using step 6.
    $k \leftarrow k+1$
  end while
  return L^{k+1}, S^{k+1}
end procedure

A.3. Convergence Analysis & Computational Complexity

Algorithm 1 is a special case of Alternating Directions method [23], [4]. These methods are a subset of more general class of methods known as Augmented Lagrange Multiplier methods. The convergence of these algorithms is well-studied [4], [14], [12]. This algorithm has been reported to perform reasonably well on a wide range of problems and small number of iterations are enough to achieve a good accuracy [6]. The complexity of nuclear norm proximal computation is $O(np^2 + p^3)$ for $n > p$ and the computational complexity of the Conjugate Gradient method for updating W is $O(np)$, per iteration. Thus, the dominant cost of each iteration corresponds to the computation of nuclear proximal operator. Our future work will be dedicated to reduce this cost by utilizing a partial SVD or an approximate SVD, as suggested in [9]. Further improvements can be made by using randomized algorithms for SVD [21] and exploiting the parallel processing capabilities [18]. Please refer to Section A.12 for a detailed comparison of computation time of this algorithm with other models considered in this work.

A.4. Properties of Various Models & Datasets

Table 2: A summary of the datasets used for the evaluation of various models. As NMF & GNMF require non-negative data so they are not evaluated for USPS, MFeat and BCI datasets.

Image / non-image	Data Type	Datasets	Evaluated Models
Image	Faces	CMU PIE (no pose changes)	all
		ORL (pose changes)	
		YALE (facial expressions)	
	Objects	COIL20 (pose changes)	
	Digits	MNIST (rotation)	
		USPS	all except NMF & GNMF
non-image	features	BCI (Brain Computer Interface)	
		MFeat (handwritten numerals)	

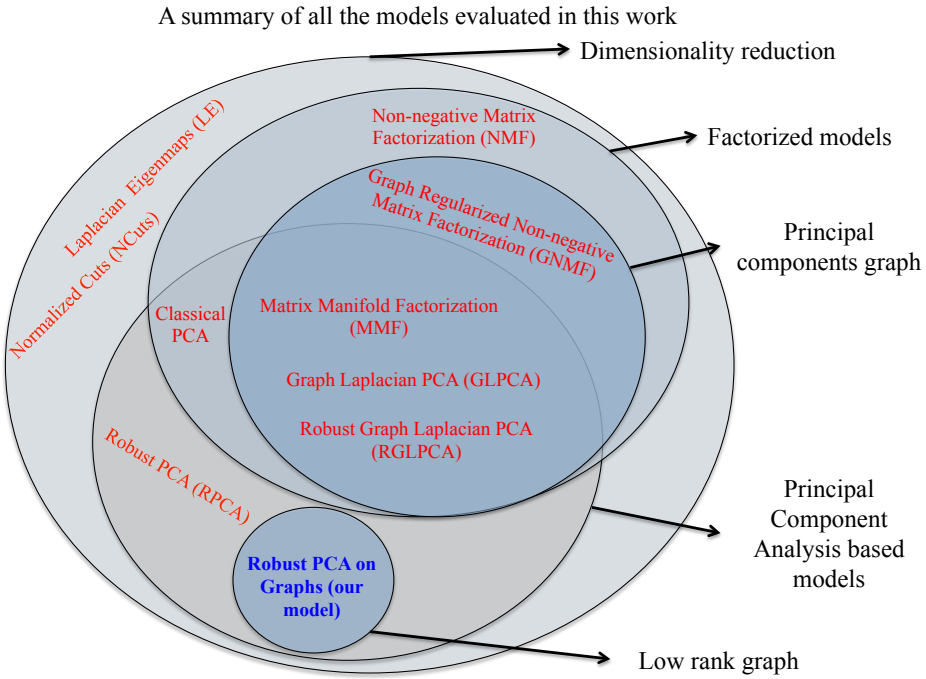


Figure 8: A venn diagram summarizing the properties of all the models evaluated in this work.

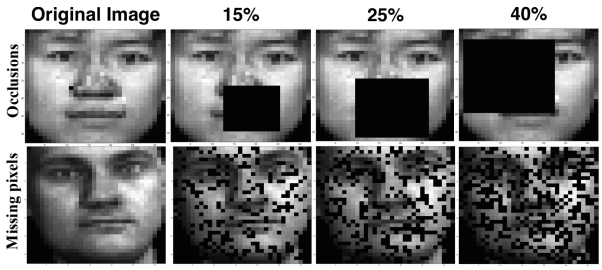


Figure 9: Sample images from CMU PIE dataset corrupted with occlusions (first row) and missing pixels (2nd row).

A.5. Evaluation Scheme and Parameter Selection for all Models

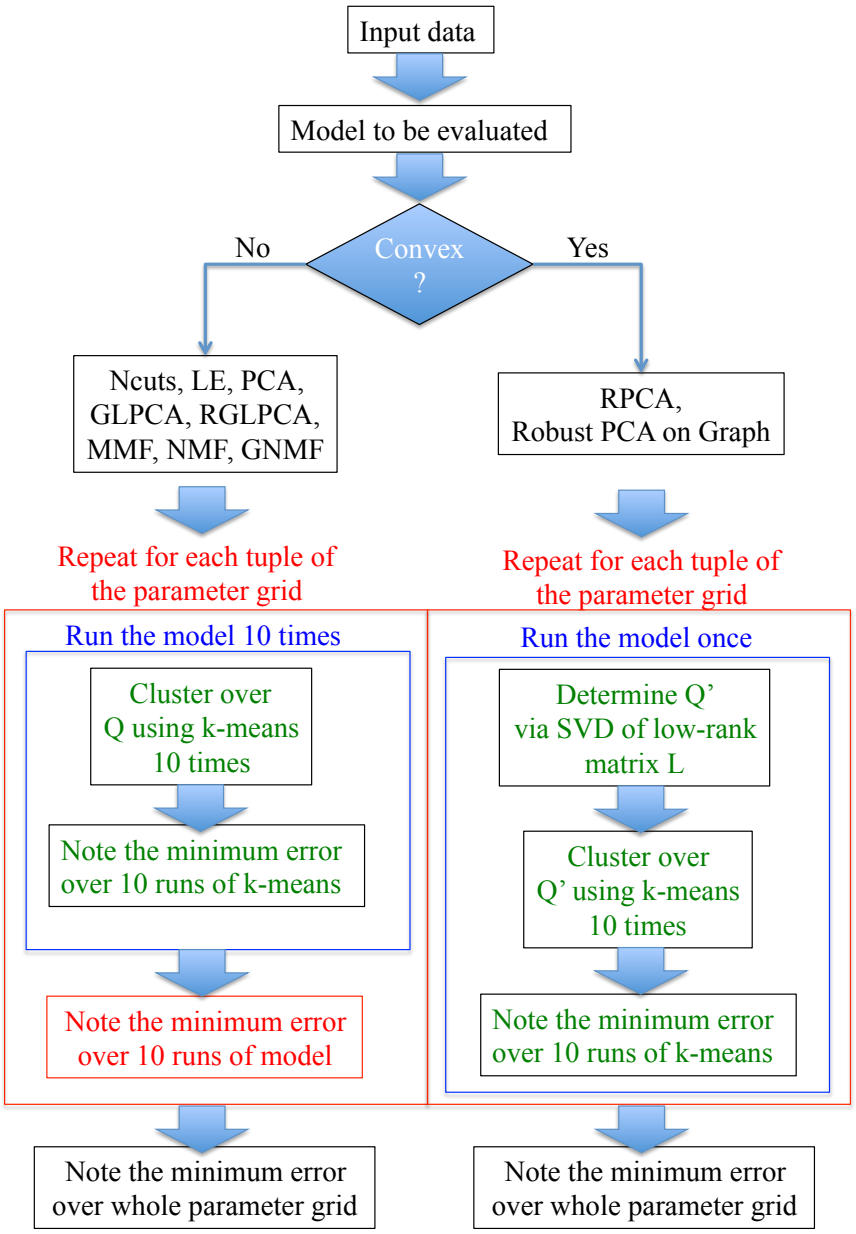


Figure 10: A flow chart describing the evaluation and parameter selection procedure for each of the models considered in this work. Each model has several parameters as described in Table 3. To perform a fair evaluation between all the models, they are run over the entire parameter range and the minimum error is reported. Further, the non-convex models are run 10 times (to determine a good local minimum) for every tuple of the parameter range and the minimum error is reported. The k-means clustering procedure for each of the models is also evaluated 10 times to avoid the bias introduced by its non-convexity.

Table 3: Range of parameter values for each of the models considered in this work. The clustering results reported in Tables 4, 5 & 6 correspond to the minimum error over this parameter range using the procedure of Fig. 10. $X \in \mathbb{R}^{p\times n}$ is the data matrix, $U \in \mathbb{R}^{p\times d}$ and $Q \in \mathbb{R}^{d\times n}$ are the principal directions and principal components in a d dimensional space (rank = d). $L = UQ \in \mathbb{R}^{p\times n}$ is the low-rank representation and $S \in \mathbb{R}^{p\times n}$ is the sparse matrix. D is the degree matrix and A is the adjacency matrix. $D - A$ is the unnormalized graph Laplacian and $\Phi = D^{-1/2}(D - A)D^{-1/2} \in \mathbb{R}^{n\times n}$ is the normalized graph Laplacian. $\|\cdot\|_F$, $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the Frobenius, nuclear and l_1 norms respectively.

	Model	**Objective**	**Constraints**	**Parameters**	**Parameter Range**	**no. of runs for each parameter value**
1	NCut [19]	$\min_Q \operatorname{tr}(Q\Phi Q^T)$	$QQ^T = I$	d	$d \in \{2^1, 2^2, \cdots, \min(n,p)\}$	10 (non-convex)
2	LE [2]	$\min_Q \operatorname{tr}(Q(D-A)Q^T)$	$QDQ^T = I$			
3	PCA	$\min_{U,Q} \|X - UQ\|_F^2$	$U^TU = I$			
4	GLPCA [10]	$\min_{U,Q} \|X - UQ\|_F^2 + \gamma \operatorname{tr}(Q\Phi Q^T)$	$QQ^T = I$	d, γ	$d \in \{2^1, 2^2, \cdots, \min(n,p)\}$	10 (non-convex)
5	RGLPCA [10]	$\min_{U,Q} \|X - UQ\|_{2,1} + \gamma \operatorname{tr}(Q\Phi Q^T)$			$\gamma \implies \beta$ using transformation in [10] $\beta \in \{0.1, 0.2, \cdots, 0.9\}$	
6	MMF [24]	$\min_{U,Q} \|X - UQ\|_F^2 + \gamma \operatorname{tr}(Q\Phi Q^T)$	$U^TU = I$	d, γ	$d \in \{2^1, 2^2, \cdots, \min(n,p)\}$ $\gamma \in \{2^{-3}, 2^{-2}, \cdots, 1000\}$	10 (non-convex)
7	NMF [13]	$\min_{U,Q} \|X - UQ\|_F^2$	$U \geq 0, Q \geq 0$	d		
8	GNMF [5]	$\min_{U,Q} \|X - UQ\|_F^2 + \gamma \operatorname{tr}(Q\Phi Q^T)$	$U \geq 0, Q \geq 0$	d, γ		
9	RPCA [6]	$\min_{L,S} \|L\|_* + \lambda\|S\|_1$	$X = L + S$	λ	$\lambda \in \{\frac{2^{-3}}{\sqrt{\max(n,p)}} : 0.01 : \frac{2^3}{\sqrt{\max(n,p)}}\}$ $\gamma \in \{2^{-3}, 2^{-2}, \cdots, 1000\}$	1 (convex)
10	Our model	$\min_{L,S} \|L\|_* + \lambda\|S\|_1 + \gamma \operatorname{tr}(L\Phi L^T)$	$X = L + S$	λ, γ		

A.6. Adjacency Matrix Construction from Corrupted & Uncorrupted Data

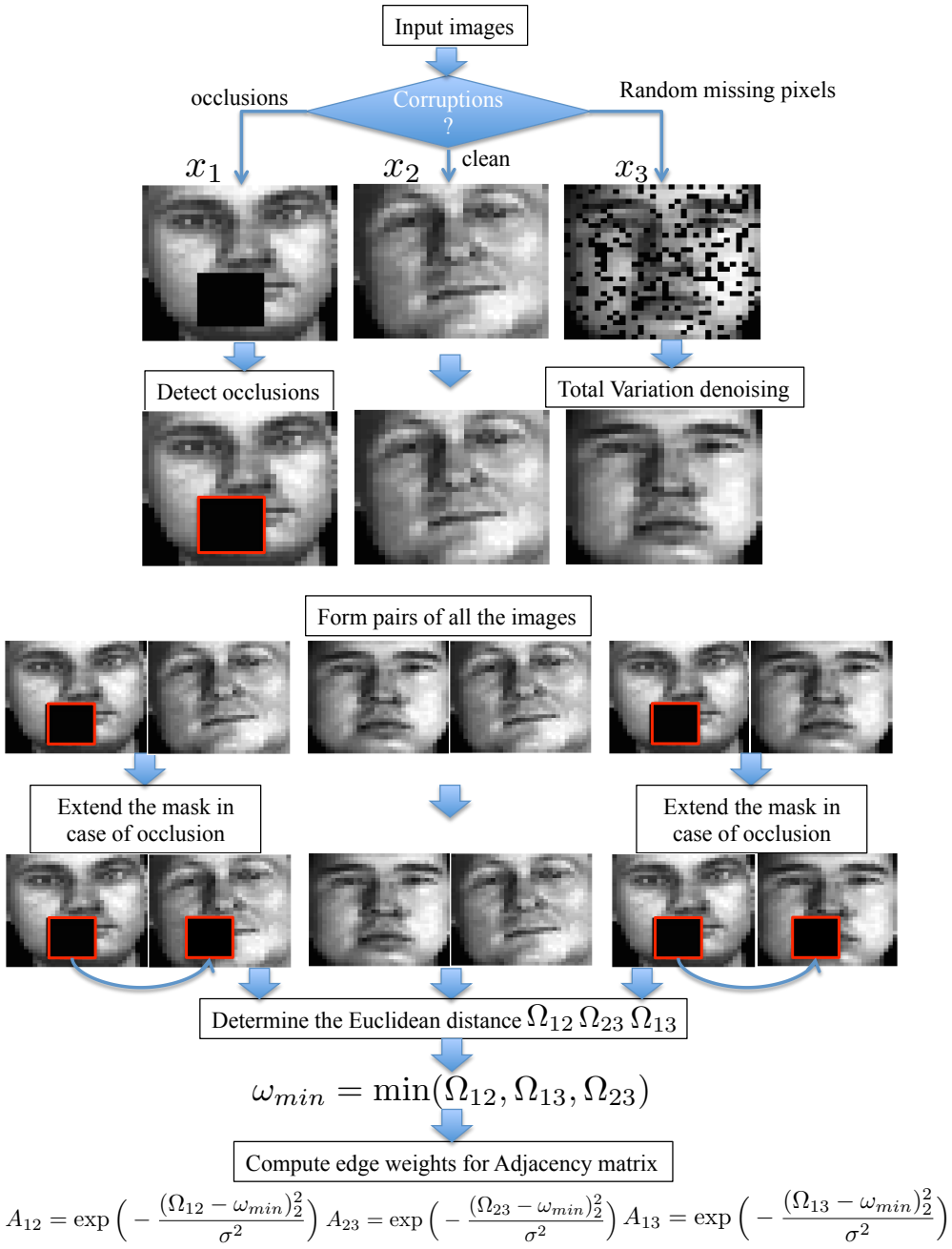


Figure 11: A procedure for the construction of Adjacency matrix A from clean and corrupted samples X. We assume that the block occlusions can be detected and the mask can be extended to the other image of the pair to remove the effect of occlusion in the calculation of Euclidean distance. The case of random missing pixels can be handled by using inpainting techniques, such as Total Variation denoising.

A.7. Clustering Results on all Databases

Table 4: A comparison of clustering error of our model with various dimensionality reduction models using the procedure of Fig. 10. The image data sets include: 1) ORL 2) CMU PIE 3) YALE and 4) COIL20. The compared models are: 1) k-means 2) Normalized Cuts (NCuts) 3) Laplacian Eigenmaps (LE) [2] 4) Standard Principal Component Analysis (PCA) 5) Graph Laplacian PCA (GLPCA) [10] 6) Robust Graph Laplacian PCA (RGLPCA) [10] 7) Manifold Regularized Matrix Factorization (MMF) [24] 8) Non-negative Matrix Factorization [13] 9) Graph Regularized Non-negative Matrix Factorization (GNMF) [5] and 10) Robust PCA (RPCA) [6]. Two types of full and partial corruptions were introduced in the data: 1) Block occlusions and 2) Random missing values. The best results are highlighted in bold. This table provides the numerical errors for the bar plots shown in Fig. 1.

Data Set	**Model**	**No Corruptions**	**Sample specific corruptions**		**Full Corruptions**						
			Occlusions (25% of data)	**Missing values (25% of data)**	**Occlusions (% of image size)**			**Missing (% of image pixels)**			
			25% image size	**25% image pixels**	**15%**	**25%**	**40%**	**15%**	**25%**	**35%**	**50%**
CMU PIE [1] (faces)	k-means	72.2	72.5	73.4	78.5	79.4	81.0	72.3	76.1	75.7	80.3
	NCuts	41.7	36.0	40.0	42.0	40.0	44.0	35.0	41.3	40.7	46.7
	LE	41.7	40.0	43.3	49.3	54.0	51.3	46.3	46.0	48.3	55.0
	PCA	13.0	25.7	34.0	42.7	45.0	61.0	35.0	50.0	56.3	63.7
	GLPCA	29.8	27.7	38.3	38.0	40.2	41.3	39.3	37.5	42.7	48.2
	RGLPCA	28.0	28.3	31.7	35.0	41.3	39.0	34.3	33.7	28.7	42.0
	MMF	53.7	50.0	55.0	54.0	58.0	60.7	59.7	60.3	60.3	62.7
	NMF	49.0	68.0	70.3	76.0	80.0	79.3	49.0	76.0	80.0	79.3
	GNMF	52.7	66.0	68.3	71.3	75.0	75.3	52.7	71.3	75.0	75.3
	RPCA	3.3	**4.3**	11.0	6.7	13.3	23.3	9.7	8.3	17.0	37.3
	Our model	**0.0**	4.7	**6.9**	**5.0**	**7.7**	**17.7**	**7.7**	**7.5**	**15.7**	**34.0**
ORL[2] (faces)	k-means	35.4	57.6	50.8	64.3	68.4	77.5	36.0	42.4	49.4	57.1
	NCuts	47.0	35.7	43.0	42.3	53.0	61.0	49.7	45.7	42.7	46.7
	LE	45.3	38.0	34.7	39.0	45.3	54.3	42.7	45.0	44.0	42.7
	PCA	28.0	46.0	37.3	53.7	65.0	72.6	34.3	39.0	40.3	40.0
	GLPCA	27.6	28.3	29.7	33.6	34.1	44.3	30.3	27.9	27.6	33.3
	RGLPCA	28.3	28.7	29.0	28.3	36.3	37.3	26.0	29.3	26.7	26.3
	MMF	20.3	29.7	24.3	30.0	34.7	38.3	24.7	28.3	28.3	27.3
	NMF	29.0	31.7	27.7	78.0	79.0	81.3	29.7	39.3	50.3	53.3
	GNMF	22.7	29.0	25.3	35.0	37.0	39.3	25.3	26.7	28.0	26.7
	RPCA	18.6	27.7	20.3	26.0	35.0	71.4	24.7	26.0	27.0	36.0
	Our model	**15.7**	**20.0**	**14.3**	**23.7**	**24.7**	**31.7**	**20.0**	**18.3**	**20.3**	**21.0**
YALE [3] (faces)	k-means	53.4	54.9	64.5	71.6	73.4	73.1	54.3	56.4	58.9	63.5
	NCuts	54.5	57.6	64.4	66.7	64.2	68.5	56.4	56.9	61.2	61.8
	LE	57.6	57.0	61.8	63.6	68.5	69.1	57.8	59.4	60.6	66.7
	PCA	53.9	49.7	58.8	61.8	66.1	70.9	55.7	55.1	62.4	61.2
	GLPCA	49.1	50.9	53.9	54.5	61.2	58.2	50.9	52.7	48.5	56.9
	RGLPCA	48.5	50.0	50.9	54.5	55.2	58.8	50.3	53.9	49.7	50.9
	MMF	38.8	37.6	46.7	55.7	55.2	53.9	38.2	38.2	44.2	49.1
	NMF	63.0	63.6	70.3	72.1	71.5	72.1	64.8	61.2	61.8	66.7
	GNMF	56.9	55.8	57.6	60.6	64.8	60.6	56.9	56.9	58.2	59.4
	RPCA	39.4	40.6	45.5	61.8	67.9	63.0	42.4	39.4	43.6	63.6
	Our model	**35.1**	**35.8**	**40.0**	**43.6**	**46.1**	**50.9**	**35.2**	**35.8**	**39.4**	**50.3**
COIL20 [4] (objects)	k-means	32.0	55.2	30.2	34.9	45.8	57.8	36.3	34.8	37.8	42.8
	NCuts	44.5	30.5	39.5	38.5	38.5	43.5	38.0	50.0	48.0	47.5
	LE	38.0	31.5	27.0	34.5	36.5	40.0	31.0	37.5	40.0	39.0
	PCA	30.5	43.5	28.5	28.0	37.0	50.0	24.0	30.0	33.5	31.0
	GLPCA	20.5	22.0	17.0	25.0	25.5	25.3	21.7	21.5	21.5	23.3
	RGLPCA	18.5	23.5	17.5	20.5	22.0	23.5	22.0	20.0	22.0	21.0
	MMF	18.0	19.0	11.5	19.0	18.0	25.0	19.0	18.0	19.0	18.5
	NMF	20.0	30.0	19.0	32.5	47.0	60.5	17.5	26.0	24.0	26.0
	GNMF	**15.5**	18.5	9.5	20.0	19.5	30.0	14.5	13.5	**11.0**	17.0
	RPCA	19.0	33.0	13.5	19.0	28.0	50.5	15.0	16.0	16.5	27.0
	Our model	**15.5**	**18.0**	**9.0**	**18.0**	**17.5**	**19.5**	**8.5**	**12.0**	16.5	**15.0**

Table 5: A comparison of clustering error of our model with various dimensionality reduction models. The image data sets include: 1) MNIST and 2) USPS. The compared models are: 1) k-means 2) Normalized Cuts (NCuts) [19] 3) Laplacian Eigenmaps (LE) [2] 4) Standard Principal Component Analysis (PCA) 5) Graph Laplacian PCA (GLPCA) [10] 6) Robust Graph Laplacian PCA (RGLPCA) [10] 7) Manifold Regularized Matrix Factorization (MMF) [24] 8) Non-negative Matrix Factorization [13] 9) Graph Regularized Non-negative Matrix Factorization (GNMF) [5] and 10) Robust PCA (RPCA) [6]. Two types of full and partial corruptions were introduced in the data: 1) Block occlusions and 2) Random missing values. The best results are highlighted in bold.

Data Set	**Model**	**No Corruptions**	**Sample specific corruptions**		**Full Corruptions**						
			Occlusions (25% of data)	**Missing values (25% of data)**	**Occlusions (% of image size)**			**Missing (% of image pixels)**			
			25% image size	**25% image pixels**	**15%**	**25%**	**40%**	**15%**	**25%**	**35%**	**50%**
MNIST[5] (digits)	k-means	62.2	71.2	59.6	68.2	79.2	82.5	62.0	60.5	60.6	65.9
	NCuts	69.3	86.7	59.0	85.3	87.3	86.0	77.7	78.7	83.7	86.7
	LE	71.0	88.0	55.3	88.0	89.3	89.0	68.7	55.3	86.3	89.3
	PCA	52.0	60.0	51.3	68.3	77.0	77.3	46.0	57.3	61.0	57.0
	GLPCA	49.3	63.7	44.7	59.3	71.7	79.0	45.3	51.7	56.3	57.1
	RGLPCA	46.3	64.7	51.0	55.3	65.3	73.7	52.7	53.0	53.7	44.3
	MMF	45.7	65.7	45.0	52.7	65.3	77.3	52.7	44.7	53.3	55.7
	NMF	46.3	67.3	33.7	60.0	72.3	72.0	52.0	47.7	49.7	47.0
	GNMF	49.7	72.7	35.3	51.7	71.0	83.7	50.7	42.7	44.7	40.7
	RPCA	39.0	53.3	43.0	64.7	73.7	78.0	44.7	44.7	52.7	70.7
	Our model	**31.7**	**46.0**	**32.3**	**53.0**	**62.7**	**69.7**	**29.7**	**33.7**	**35.3**	**37.7**
USPS (digits)	k-means	51.0	45.3	54.0	55.4	66.2	73.8	41.2	41.7	40.1	38.9
	NCuts	54.0	49.3	61.3	68.3	72.7	75.7	53.0	47.7	55.7	55.3
	LE	64.7	52.7	67.7	74.3	81.7	83.0	65.0	71.7	66.0	63.3
	PCA	49.7	42.0	50.3	55.0	66.0	71.7	43.3	36.0	39.3	48.0
	GLPCA	45.8	26.7	45.0	46.9	56.7	68.9	39.7	38.3	33.5	32.5
	RGLPCA	42.0	33.3	46.7	45.0	59.0	63.7	33.3	26.7	35.7	37.0
	MMF	33.0	22.3	40.7	**37.7**	50.3	62.6	**21.5**	21.6	20.7	**20.7**
	RPCA	28.3	29.0	46.7	56.0	67.7	69.7	29.0	32.3	32.6	38.0
	Our model	**21.3**	**21.7**	**35.3**	42.0	**48.0**	**56.0**	**21.5**	**21.3**	**20.5**	27.3

Table 6: A comparison of clustering error of PCA models and simple k-means for MFeat and BCI data sets. Each of the data sets was corrupted with two types of outliers: 1) Block occlusions and 2) Missing values. Block occlusions in non-image databases correspond to an unrealistic assumption so such experiments were no performed for these datasets. Furthermore, NMF and GNMF were not evaluated because they require non-negative data whereas these datasets are negative as well. The best results are highlighted in bold.

Data Set	**Model**	**No Corruptions**	**Sample specific Missing values (25% of data) 25% features per sample**	**Full Corruptions Missing values (% features per sample)**			
				15%	**25%**	**35%**	**50%**
MFeat [6]	k-means	32.4	30.8	25.5	29.9	32.2	33.9
	NCuts	39.8	47.0	26.3	28.3	33.5	37.8
	LE	38.0	47.3	35.3	33.0	33.5	50.8
	PCA	7.3	7.3	10.0	11.8	9.0	13.8
	GLPCA	6.0	12.5	6.0	7.8	8.0	10.0
	RGLPCA	9.5	15.0	15.0	17.3	8.3	14.8
	MMF	7.3	6.3	6.0	5.3	7.0	7.0
	RPCA	5.0	4.3	3.8	6.3	9.5	15.0
	Our model	**3.3**	**2.0**	**3.5**	**4.3**	**6.8**	**7.0**
BCI [7]	k-means	47.8	47.2	47.8	47.8	48.0	48.2
	NCuts	47.0	47.0	47.3	47.5	47.3	48.3
	LE	46.5	49.7	49.8	49.8	49.8	45.8
	PCA	45.3	45.2	42.0	43.0	43.0	45.5
	GLPCA	46.5	46.0	45.8	45.8	46.5	45.0
	RGLPCA	45.5	43.7	46.5	44.3	43.8	42.8
	MMF	47.25	47.2	47.0	47.5	47.3	48.3
	RPCA	**39.8**	43.0	40.0	42.8	43.5	43.0
	Our model	40.3	**37.7**	**39.0**	**42.3**	**42.0**	**41.0**

A.8. A Comparison of the Principal Directions U Learned with Low-Rank and Principal Components Graph

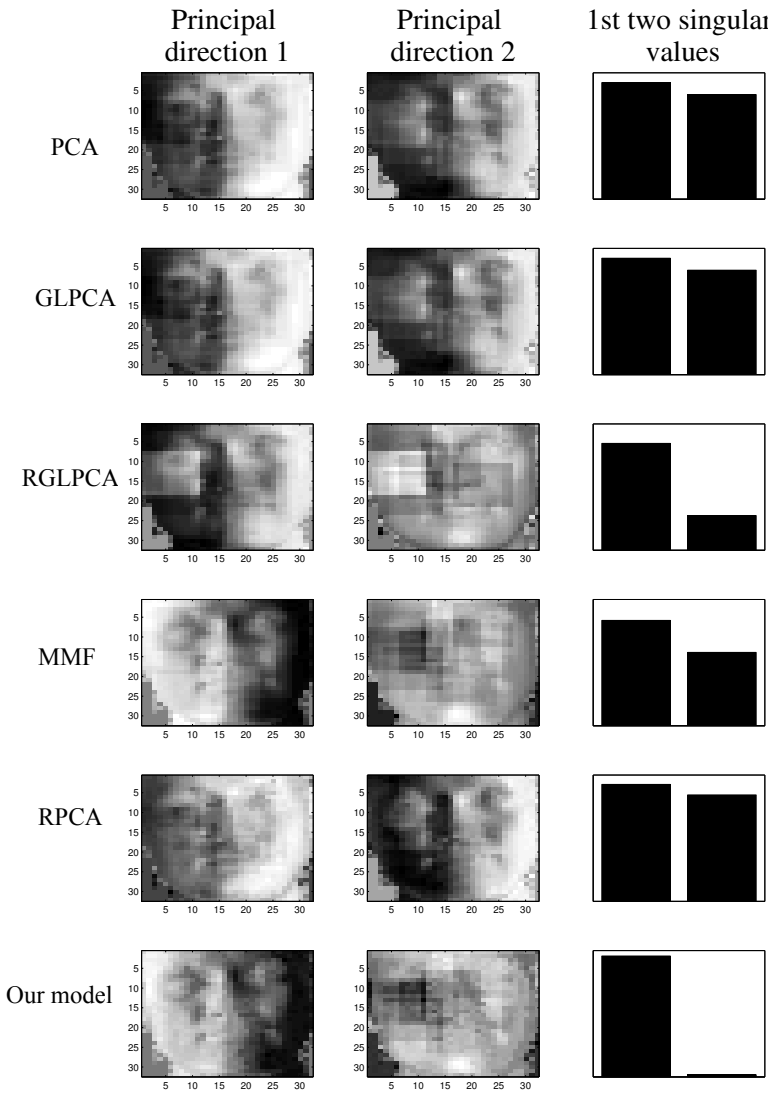


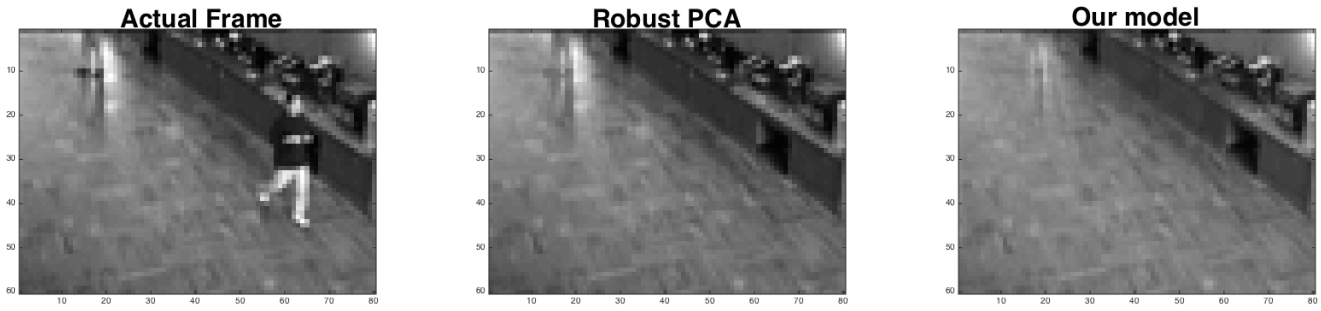
Figure 12: A comparison of the principal directions U learned with low-rank graph $\mathrm{tr}(L\Phi L^T)$ and principal components graph regularization$\mathrm{tr}(Q\Phi Q^T)$ for the CMU PIE dataset. Each row shows the 1^{st} two principal directions U and the corresponding singular values for each of the PCA models evaluated in this work. For the factorized models (PCA, GLPCA, RGLPCA and MMF) the principal directions U are explicitly learned. For RPCA and our model, U is obtained by singular value decomposition of the low-rank matrix L, *i.e.* $L = U\Sigma Q^{'}$. The low-rank graph for our model clearly helps in learning a corruption free U. Even though the 2^{nd} principal direction for our model has a large effect of occlusion, we note that the corresponding singular value (2^{nd} bar) is much smaller as compared to the 1^{st}. For all the other models, each of the two principal directions are affected by occlusions and the corresponding singular values are significant. **This explains why a low-rank graph helps in an enhanced low-rank recovery and a lower clustering error in low dimensional space than the principal components graph**.

A.9. Additional Low-Rank Recovery Results on CMU PIE Dataset

Low Rank recovery of occluded images of CMU PIE dataset with the best visual results highlighted in red

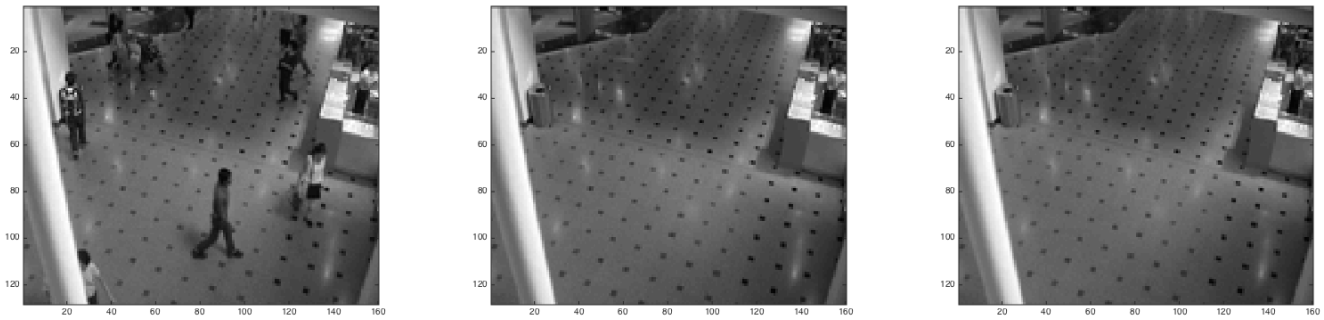
Figure 13: A comparison of the clean low-rank recovered images of the CMU PIE data set corresponding to each of the PCA models considered in this work. For this experiment, the images of one person are corrupted by block occlusions occupying 10% of the image. Each row corresponds to a different image of the same person occluded at a random position. 1^{st} figure in each row shows the actual occluded image. 2^{nd} and 3^{rd} figures show the whitened occluded and un-occluded images. Since PCA requires whitening, the recovered low-rank images in figures 4 to 8 using GLPCA, RGLPCA, MMF, RPCA and our model resemble the un-occluded whitened image. The best recovered low-rank representations (via a visual inspection) are highlighted with a red box.

A.10. Additional Results for Background Separation from Videos



(a) Restaurant food counter. The lightning effect (not static) is less visible in right figure than the middle one.

(b) Airport hallway. The moving person (not a part of static background) is less visible in the right figure than the middle one.



(c) Shopping mall lobby. All the moving people are removed in both the middle and right figures.

Figure 14: Static background separation from three different videos. a) restaurant food counter, b) airport hallway and c) shopping mall lobby. In each of the cases: the leftmost figure shows one actual frame of the video, the middle figure shows the recovered static background using Robust PCA [6] with $\lambda = 1/\sqrt{\max(n,p)}$ and the rightmost figure shows the recovered background using our model with $\lambda = 1/\sqrt{\max(n,p)}$ and $\gamma = 10$. The effect of graph can be appreciated in the first two cases. In a) the changes in the illumination (which are not a part of static background) are more visible in the Robust PCA model than our model. In b) the moving person is more obvious in Robust PCA recovered background than our model. For c) the two models perform equally well. Please note that depending on the video, some tuning of parameter λ might be needed [8]. We recommend a range of $\lambda \in [1,2]/\sqrt{\max(n,p)}$ as shown in our low-rank recovery experiments for artificial datasets in Fig. 16. However, for the three videos used in this work $\lambda = 1/\sqrt{\max(n,p)}$ was found as the best parameter

.

A.11. Parameter Selection for the Proposed Model

A.11.1 Parameter Selection for Clustering

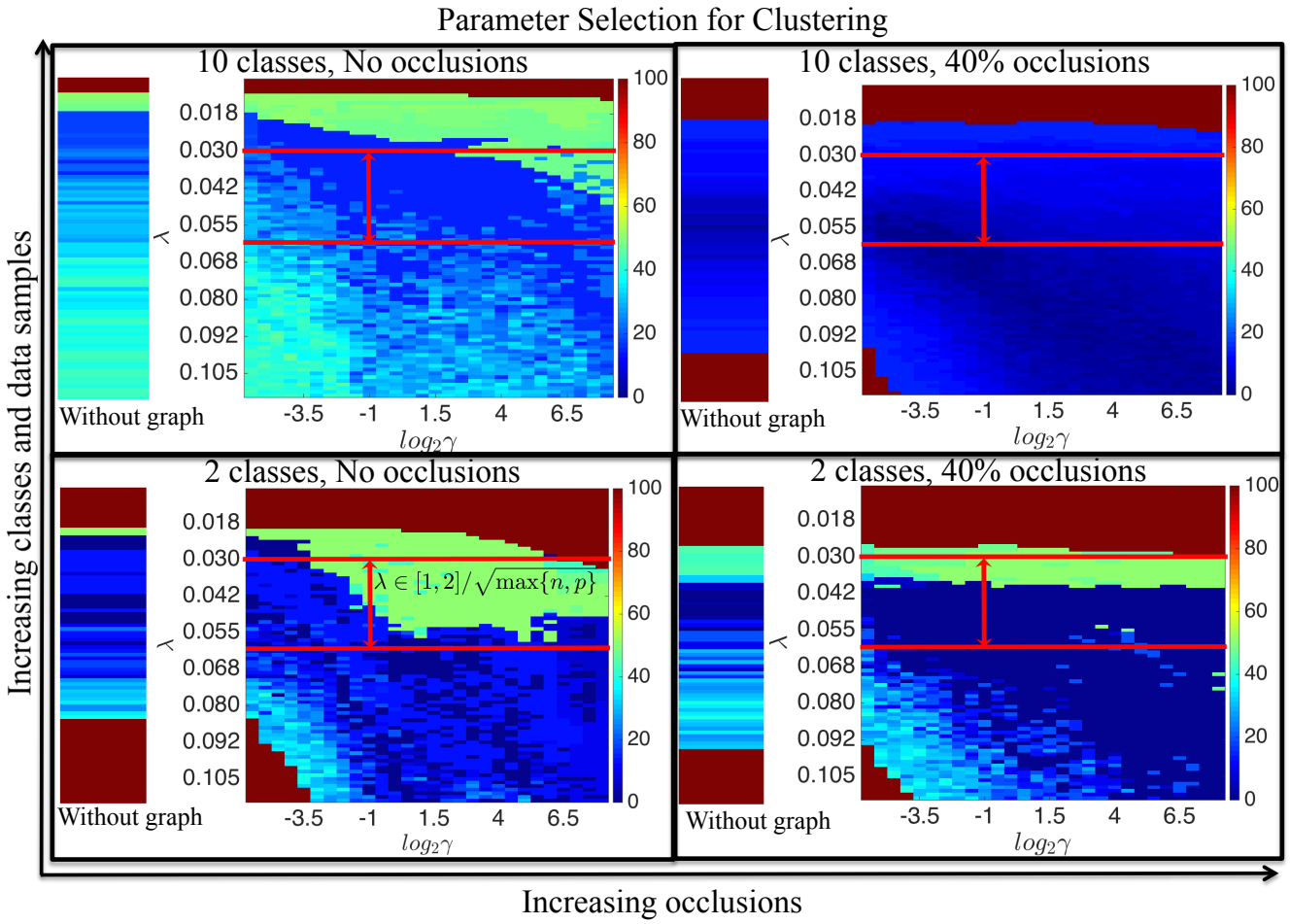


Figure 15: Variation of clustering error over (λ, γ) grid for different size, classes and occlusions in the CMU PIE data set. x-axis shows the variation with increasing occlusion size and y-axis with increasing size of the data matrix n and number of classes k. The horizontal red lines show a range of $\lambda \in [1,2]/\sqrt{\max(n,p)}$. The clustering error always attains a minimum value within this range of λ irrespective of the size of the data matrix, occlusions or the number of classes. Furthermore, a wide range of γ values attain a minimum clustering error in the region between the red lines. **Thus, a simple rule can be used to choose a set of good parameters**: *Fix $\lambda \in [1,2]/\sqrt{\max(n,p)}$ and then perform a cross-validation over a coarse range of γ values. In fact $\lambda = 1/\sqrt{\max(n,p)}$ is always a good choice for the images which have no occlusions and are only affected by shadows and illumination changes*. Each strip on the left of (λ, γ) plot shows the variation of clustering error with λ (y-axis) for $\gamma = 0$ (Robust PCA). Clearly, the use of a graph increases the range the possible λ values which attain a minimum clustering error.

A.11.2 Parameter Selection for Low-Rank Recovery

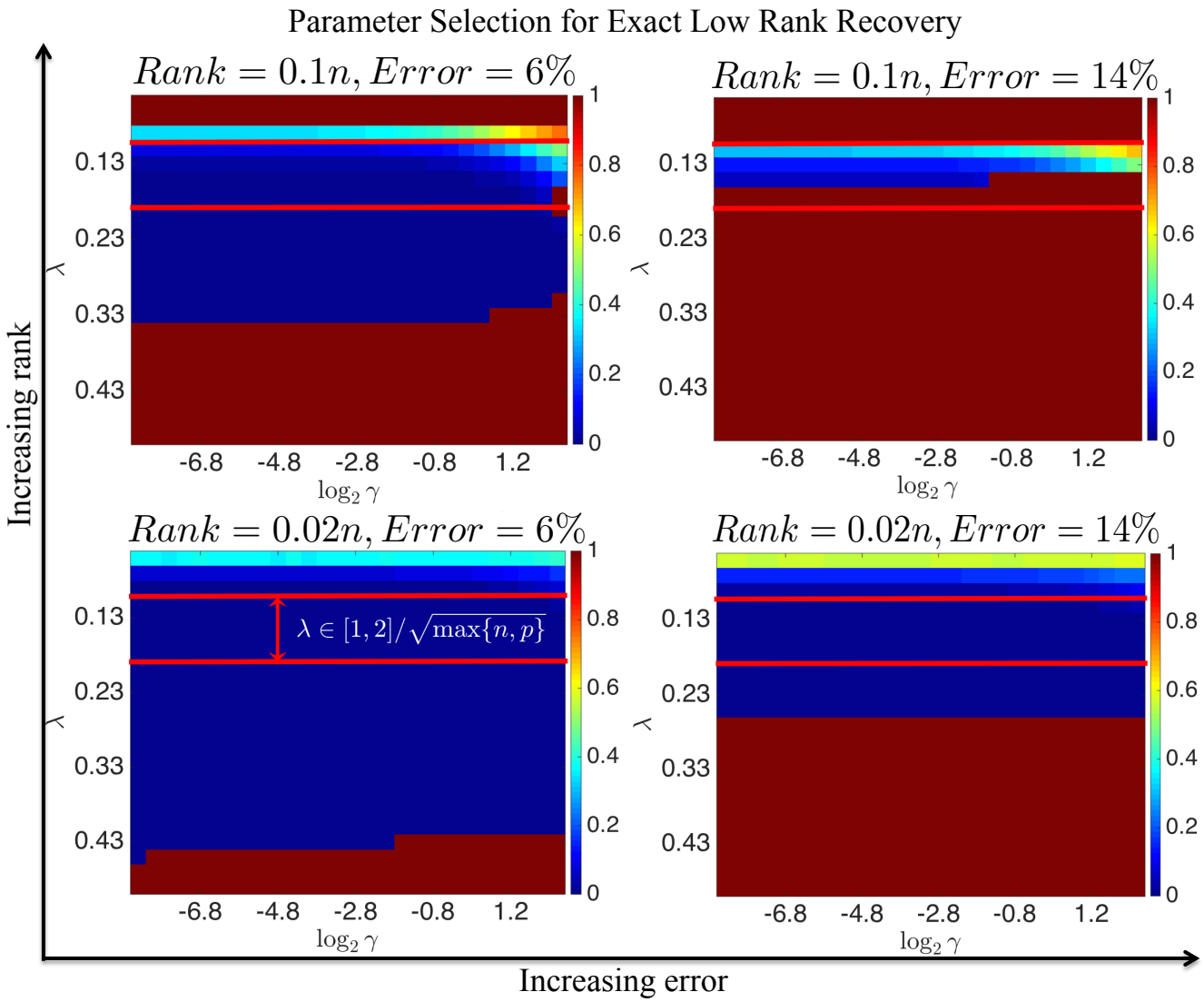


Figure 16: Variation of low-rank normalized reconstruction error over (λ, γ) grid for different ranks and errors (corruptions) in the artificially generated dataset with Bernoulli support and random sign scheme for sparse errors (Section 8.1). x-axis shows the variation with increasing fraction of errors and y-axis with increasing rank of the data matrix X. The horizontal red lines show a range of $\lambda \in [1, 2]/\sqrt{\max(n, p)}$. The clustering error always attains a minimum value within this range of λ irrespective of the size of the $(rank, error)$ setting. Furthermore, a wide range of γ values attain a minimum clustering error in the region between the red lines. **Thus, the same simple rule as clustering can be used to choose a set of good parameters**: *Fix* $\lambda \in [1, 2]/\sqrt{\max(n, p)}$ *and then perform a cross-validation over a coarse range of* γ *values*. Similar to our observation for clustering error, $\lambda = 1/\sqrt{\max(n, p)}$ is always a good choice for the datasets with low fraction of errors.

A.12. Comparison of Computation Times

Table 7: A comparison of computational times (in secs) for one run of each model with increasing size of data matrix. For this experiment, different size of the CMU PIE dataset (n = 300, 600 and 1200) is corrupted with 20% occlusions and the computation time and accuracy is computed for one run of each model. To perform a fair comparison between the models, the parameter tuple for each model is chosen using the procedure of Section A.5. The non-convex models are run 10 times and the computational time and accuracy of the run with the minimum clustering error are reported. The convex models are run only once. Clearly, RGLPCA has the highest computation time, followed by our proposed model. However, the trade-off between the clustering error and computational time is worth observing. Our model takes more time to converge but attains the minimum clustering error. This large computation time is dominated by the expensive SVD step in every iteration. **Our future work will be focused on reducing the complexity of this model by exploiting the distributed and parallel computation techniques.**

	Model	**n = 300**		**n = 600**		**n = 1200**	
		Time (secs)	**Clustering error (%)**	**Time (secs)**	**Clustering error (%)**	**Time (secs)**	**Clustering error (%)**
1	NCuts	0.24	47.2	0.72	48.3	1.31	49.0
2	LE	0.24	48.1	0.70	50.2	1.24	52.7
3	PCA	0.11	34.3	0.17	35.9	0.20	38.3
4	GLPCA	0.12	37.5	0.43	36.1	1.60	37.2
5	RGLPCA	150.4	34.8	356.6	35.3	1187.6	37.2
6	MMF	0.13	36.7	0.32	37.5	1.52	38.9
7	NMF	0.15	43.9	0.62	45.6	1.30	47.4
8	GNMF	1.20	41.2	0.62	40.3	1.30	45.9
9	RPCA	59.8	10.5	159.3	12.6	678.3	14.9
10	Our model	69.7	8.4	169.6	10.9	869.8	13.8