A Proof of Lemma 3.1

Proof. By the first order optimality condition of (2.3), we have:

$$A^\top(AUV^\top + E - Z)V + \frac{1}{\beta}U = 0.$$

It follows by a simple calculation that

$$A^\top AUV^\top V + \frac{1}{\beta}U = A^\top(Z-E)V.$$

It is known that the minimizer U in the above equation has a closed form solution. To derive it, we need the vectorization operator $\text{vec}(\cdot)$ which stacks all the columns of a matrix. Taking the vectorization form on both sides, and note that

$$\text{vec}(A^\top AUV^\top V) = (V^\top V)\otimes(A^\top A)\text{vec}(U),$$

which together give

$$\left((V^\top V)\otimes(A^\top A) + \frac{1}{\beta}I_{dn}\right)\text{vec}(U) = \text{vec}(A^\top(Z-E)V),$$

implying

$$\text{vec}(U) = \left((V^\top V)\otimes(A^\top A) + \frac{1}{\beta}I_{dn}\right)^{-1}\text{vec}(A^\top(Z-E)V). \tag{A.1}$$

□

B Technical Lemmas

We propose Lemma B.1, which is essentially parallel to the Elastic Net [50].

Lemma B.1. *Given matrix $M \in \mathbb{R}^{p\times n}$ and $N \in \mathbb{R}^{p\times d}$, optimizing the following convex program:*

$$\min_V \frac{\beta}{2}\left\|M - NV^\top\right\|_F^2 + \frac{1}{2}\|V\|_F^2 + \lambda_1\|V\|_1,$$

amounts to solving the the Lasso-type problem:

$$\min_V \frac{1}{2}\left\|M' - N'V^\top\right\|_F^2 + \lambda_1\|V\|_1,$$

where

$$M' = \begin{bmatrix}\sqrt{\beta}M\\ 0\end{bmatrix},\ N' = \begin{bmatrix}\sqrt{\beta}N\\ -I\end{bmatrix}.$$

Proof. By a simple algebraic calculation, we derive the desired result. □

Remark. Lemma B.1 tells us that optimizing a least-square loss with a combination of Frobenius norm and an ℓ_1 matrix norm can be reduced to minimizing a Lasso by data augmentation. With this lemma and put $M = Z - E$, $N = AU$, the local minimizer of V for NLRR_{21} (2.10) can be efficiently solved by popular Lasso solvers, e.g., [10, 22].

C Proof of Theorem 4.1

Proof. Put $M = Z - E$ and consider the derivative of $g(D, U, V, E, W, \mu)$ at the optimal points v_{ij} and v_{ik}, we have:

$$-\beta\boldsymbol{d}_j^\top(\boldsymbol{m}_i - D\boldsymbol{v}(i)^\top) + v_{ij} + \lambda_1\,\text{sign}(v_{ij}) = 0,$$
$$-\beta\boldsymbol{d}_k^\top(\boldsymbol{m}_i - D\boldsymbol{v}(i)^\top) + v_{ik} + \lambda_1\,\text{sign}(v_{ik}) = 0.$$

Subtracting the first equation from the second one, and using the fact that $\text{sign}(v_{ij}) = \text{sign}(v_{ik})$ since $v_{ij}v_{ik} > 0$, gives

$$v_{ij} - v_{ik} = \beta\left(\boldsymbol{d}_j - \boldsymbol{d}_k\right)^\top\left(\boldsymbol{m}_i - D\boldsymbol{v}(i)^\top\right). \tag{C.1}$$

On the other hand, we have

$$\begin{aligned} &g(D, U, V, E, W, \mu)\\ =\ &\frac{1}{2}\|U\|_F^2 + \sum_{i=1}^n\Big[\frac{\beta}{2}\left\|\boldsymbol{z}_i - \boldsymbol{e}_i - D\boldsymbol{v}(i)^\top\right\|_2^2\\ &+ \frac{1}{2}\|\boldsymbol{v}(i)\|_2^2 + \lambda_1\|\boldsymbol{v}(i)\|_1 + \lambda_2\|\boldsymbol{e}_i\|_1\Big]\\ &+ \langle W, D - AU\rangle + \frac{\mu}{2}\|D - AU\|_F^2, \end{aligned}$$

implying $(U, \boldsymbol{v}(i), \boldsymbol{e}_i)$ minimizes the ith term in the above summations. Thus, constructing a trivial solution as $(U', \boldsymbol{v}(i)', \boldsymbol{e}_i') = (0, 0, \boldsymbol{z}_i)$ and utilizing the optimality of $(U, \boldsymbol{v}(i), \boldsymbol{e}_i)$ gives

$$\frac{\beta}{2}\|\boldsymbol{z}_i - \boldsymbol{e}_i - D\boldsymbol{v}(i)\|_2^2 \le \lambda_2\|\boldsymbol{z}_i\|_1 \le \lambda_2 c.$$

Thus,

$$\|\boldsymbol{z}_i - \boldsymbol{e}_i - D\boldsymbol{v}(i)\|_2 \le \sqrt{2\lambda_2 c/\beta}.$$

Using this inequality gives an upper bound for the left-hand side of (C.1):

$$\begin{aligned} |v_{ij} - v_{ik}| &\le \beta\|\boldsymbol{d}_j - \boldsymbol{d}_k\|_2 \cdot \left\|\boldsymbol{m}_i - D\boldsymbol{v}(i)^\top\right\|_2\\ &\le \sqrt{2\lambda_2 c\beta}\,\|\boldsymbol{d}_j - \boldsymbol{d}_k\|_2. \end{aligned}$$

□

Remark. The proof is essentially inspired by [50] with one difference. In our work, we are working on the matrix case. Thus, we require a uniform boundedness on the observation Z, i.e., $\|\boldsymbol{z}_i\|_1 \le c$ holds for any i. Furthermore, c should be an absolute constant that cannot go to infinity, which is commonly satisfied in real-world applications.

D Full Comparisons on MNIST

We record the full comparison results in Table 5. Note that NRPCA and DL are not originally used for subspace clustering. However, as suggested by [26], the matrix $VV^\top$

Table 5: **Full results on MNIST dataset.** Top: MNIST-2K. Bottom: MNIST-10K. We use $A = Z$ for LRR, NLRR and its variants, and use $A = A_0$ for LRR2, NLRR2 and its variants.

	NRPCA	DL	SSC	LRSSC	LRR	LRR2	NLRR	NLRR_{21}	NLRR_{1}	NLRR2	NLRR2_{21}	NLRR2_{1}
Acc. (%)	50.90	40.40	46.45	51.00	54.60	55.00	55.40	**59.15**	52.95	51.75	51.20	45.45
Acc. (%)	54.06	47.76	44.90	53.4	55.15	53.67	58.63	59.50	**59.67**	52.67	53.26	53.61

can be used as the similarity matrix for clustering, where V is the optimal solution of (2.7). Likewise, we use the output of DL. (2.8) for clustering.

Regarding the parameters of NRPCA and DL, we note that from the unified formulation we discussed in Section 2, the two problems are comparable to NLRR_{21}. Hence, the parameters of NRPCA and DL are chosen as the corresponding ones used in NLRR_{21}. This setting also make us focus on the effect of the difference of D and V for each problem rather than parameter tuning. For LRSSC [44], we use the parameters provided in their source code.

Although we observe from Figure 2 that setting $A = A_0$ can boost the clustering accuracy for simulation data, it does not hold for the realistic dataset MNIST. Even worse is that using $A = A_0$ will dramatically decrease the performance of LRR, NLRR and its variants. See the bottom panel of Table 5. We also emphasize that finding the A_0 is computationally expensive. For MNIST-2K, it takes 34 minutes to learn A_0 while that for MNIST-10K is 168 minutes, hence not practical.

The method LRSSC [44] combines LRR and SSC by penalizing a weighted nuclear norm of X (i.e., the one in (2.1)) in SSC (2.9). We find the resulted program is too slow to optimize. Moreover, although theoretically sound, [44] did not show how to pick a proper balancing parameter so as to achieve their theoretical guarantee. In fact, we tried several choices of the parameters for LRSSC, but the results are comparable to the one shown in the table. Due to the efficiency of LRSSC, it is hard for practitioners to tune the parameters. Hence, LRSSC appears not practical for large scale problems.