Doubly Stochastic Subspace Clustering

Derek Lim  
Department of Computer Science  
Cornell University  
dl772@cornell.edu

René Vidal  
Mathematical Institute for Data Science  
Johns Hopkins University  
{rvidal, bhaeffele}@jhu.edu

Abstract

Many state-of-the-art subspace clustering methods follow a two-step process by first constructing an affinity matrix between data points and then applying spectral clustering to this affinity. Most of the research into these methods focuses on the first step of generating the affinity matrix, which often exploits the self-expressive property of linear subspaces, with little consideration typically given to the spectral clustering step that produces the final clustering. Moreover, existing methods obtain the affinity by applying ad-hoc postprocessing steps to the self-expressive representation of the data, and this postprocessing can have a significant impact on the subsequent spectral clustering step. In this work, we propose to unify these two steps by jointly learning both a self-expressive representation of the data and an affinity matrix that is well-normalized for spectral clustering. In the proposed model, we constrain the affinity matrix to be doubly stochastic, which results in a principled method for affinity matrix normalization while also exploiting the known benefits of doubly stochastic normalization in spectral clustering. While our proposed model is non-convex, we give a convex relaxation that is provably equivalent in many regimes; we also develop an efficient approximation to the full model that works well in practice. Experiments show that our method achieves state-of-the-art subspace clustering performance on many common datasets in computer vision.

1. Introduction

Subspace clustering seeks to cluster a set of data points that are approximately drawn from a union of low dimensional linear (or affine) subspaces into clusters, where each linear (or affine) subspace defines a cluster (i.e., every point in a given cluster lies in the same subspace) [34]. The most common class of subspace clustering algorithms for clustering a set of $n$ datapoints are those that proceed in two stages: 1) Learning an affinity matrix $A \in \mathbb{R}^{n \times n}$ that defines the similarity between pairs of datapoints; 2) Applying a graph clustering technique, such as spectral clustering [36, 32], to produce the final clustering. Within this general framework, a large majority of research into subspace clustering methods has largely focused on how to construct the affinity matrix, with relatively little attention given to the subsequent spectral clustering step. In particular, arguably the most popular model for subspace clustering is to construct the affinity matrix by exploiting the ‘self-expressive’ property of linear (or affine) subspaces, where a point within a given subspace can be represented as a linear combination of other points within the subspace [7, 8, 25, 43, 44, 34].

For a dataset $X \in \mathbb{R}^{d \times n}$ of $n$, $d$-dimensional points, this is typically captured by an optimization problem of the form:

$$
\min_C \frac{1}{2} \| X - XC \|_F^2 + \lambda \theta(C),
$$

where the first term captures the self-expressive property, $X \approx XC$, and the second term, $\theta$, is some regularization term on $C$ to encourage that a given point is primarily represented by other points from its own subspace and to avoid trivial solutions such as $C = I$. Once the self-expressive representation $C$ has been learned from (1), the final affinity matrix $A$ is then typically constructed by rectifying and symmetrizing $C$, e.g., $A = (C + C^\top)/2$.

However, as we detail in Section 2.1, many subspace clustering methods require ad-hoc or unjustified postprocessing procedures on $C$ to work in practice; this is illustrated in Figure 1. These postprocessing steps serve to normalize $C$ to produce an affinity $A$ with better spectral clustering performance, but they add numerous arbitrary hyperparameters to the models and receive little mention in the associated papers. Likewise, it is well-established that some form of Laplacian normalization is needed for spectral clustering to be successful, for which the practitioner again has multiple choices of normalization strategy.

In this paper, we propose a new subspace clustering framework which explicitly connects the self-expressive step, the affinity normalization, and the spectral clustering step. To motivate our proposed model, we first discuss the desired properties of subspace clustering affinities one would like for successful spectral clustering. Specifically,
we desire affinities $A$ that have the following properties:

(A1) **Well-normalized for spectral clustering.** We should better leverage knowledge that $A$ will be input to spectral clustering. Many forms of ad-hoc postprocessing in subspace clustering perform poorly-justified normalization, and there is also a choice of normalization to be made in forming the graph Laplacian [36].

(C1) **Sparsity.** Much success in subspace clustering has been found with enforcing sparsity on $C$ [7, 8, 44]. In particular, many sparsity-enforcing methods are designed to ensure that the nonzero entries of the affinity matrix correspond to pairs of points which belong to the same subspace. This leads to high clustering accuracy, desirable computational properties, and provable theoretical guarantees.

(C2) **Connectivity.** There should be sufficiently many edges in the underlying graph of $A$ so that all points within the same subspace are connected [27, 40, 43]. Thus, there is a trade-off between sparsity and connectivity that subspace clustering methods must account for.

Properties (C1) and (C2) are well studied in the subspace clustering literature, and can be enforced on the self-expressive $C$, since rectifying and symmetrizing a $C$ with these properties, for example, maintains these properties. However, property (A1) must be enforced on $A$, so it is often neglected in several aspects and mostly handled by ad-hoc postprocessing methods, if handled at all. In working towards (A1), we first constrain $A \geq 0$, since nonnegativity of the affinity is necessary for interpretability and alignment with spectral clustering theory. Beyond this, spectral clustering also benefits from having rows and columns in the affinity matrix of the same scale; several subspace clustering algorithms postprocess the self-expressive $C$ by normalizing the columns of $C$ before forming an affinity, and many forms of Laplacian normalization for spectral clustering normalize rows and/or columns to have similar scale [36]. In particular, one form of normalization which is well established in the spectral clustering literature is to constrain the rows and columns to have unit $L_1$ norm. Because $A$ is additionally constrained to be nonnegative, this is equivalent to requiring each row sum and column sum of $A$ to be 1, resulting in constraints that restrict the affinities $A$ to be in the convex set of doubly stochastic matrices [14]:

$$\Omega_n = \{ A \in \mathbb{R}^{n \times n} \mid A \geq 0, A1 = 1, A^\top 1 = 1 \}. \quad (2)$$

Doubly stochastic matrices have been thoroughly studied for spectral clustering, and doubly stochastic normalization has been shown to significantly improve the performance of spectral clustering [45, 46, 39, 29, 12, 15, 37, 19]. Moreover, doubly stochastic matrices are invariant to the most widely-used types of Laplacian normalization [36], which removes the need to choose a Laplacian normalization scheme. Further, the authors of [46] show that various Laplacian normalization schemes can be viewed as attempting to approximate a given affinity matrix with a doubly stochastic matrix under certain distance metrics.

Beyond being well-normalized for spectral clustering (A1), the family of doubly stochastic matrices can also satisfy properties (C1) and (C2). Our proposed methods will give control over the sparsity-connectivity trade-off through interpretable parameters, and we in practice learn $A$ that are quite sparse. Additionally, doubly stochastic matrices have a guarantee of a certain level of connectivity due to the row sum constraint, which prohibits solutions with all-zero rows (that can occur in other subspace clustering methods).

**Contributions.** In this work, we develop a framework that uniﬁes the self-expressive representation step of subspace clustering with the spectral clustering step by formulating a model that jointly solves for a self-expressive representation $C$ and a doubly stochastic affinity matrix $A$. While our pro-
posed model is non-convex, we provide a convex relaxation; we
show theoretically that this relaxation is exactly equiva-
 lent to the original model in most relevant regimes and gives
close approximations to the non-convex model in all other
regimes. We additionally develop an efficient algorithm
based on linearized ADMM [26] to compute the solution
to our convex model. A closer analysis of our model allows
us to formulate an alternative algorithm to quickly compute
an approximate solution, in which we first efficiently learn
a self-expressive matrix and then subsequently fit a doubly
stochastic matrix by a regularized optimal transport prob-
lem [2, 23]. This approximate model further has interpre-
tations as a principled postprocessing for a wide class of
subspace clustering algorithms. Finally, we validate our ap-
proach with experiments on various standard subspace clus-
tering datasets, where we demonstrate that our models sig-
nificantly improve on the current state-of-the-art in a variety
of performance metrics.

1.1. Related Work

We review several methods that attempt to integrate
the self-expressive step with the final clustering or that
explore regularization similar to a doubly stochastic con-
straint. Structured SSC [20] seeks to connect the self-
expressive affinity learning step with the spectral clus-
tering step. It formulates an optimization problem in which
a self-expressive affinity and a spectral-clustering-related ma-
trix is learned, but they constrain their clustering matrix to
be a binary matrix — thus making their formulation non-
convex and requiring a full spectral clustering step in each
iteration of their optimization algorithm. In contrast, our
method only constrains the clustering matrix to be doubly
stochastic, thus maintaining a convex formulation and only
requiring a single spectral clustering at the end.

There has also been work in sparse representation that
processes a dissimilarity matrix (that can be viewed as the
complement of a corresponding affinity matrix) to learn a
sparse row stochastic representation matrix [6, 5]. These
works are mostly different in focus, as they seek to find rep-
resentatives for the data and do not use spectral clustering
in downstream tasks.

Scaled Simplex SSC learns a scaled column stochastic
matrix [42], but it only allows representation with nonneg-
ative coefficients. This and other subspace clustering meth-
ods that learn nonnegative self-expressive coefficients like
SSQP [38] fail in certain cases. They tend to only connect
points with positive correlation to each other, even though
points in the same subspace can have very negative correla-
tion (e.g. $x$ and $-x \in \mathbb{R}^d$ are in the same subspace).

While no subspace clustering methods consider learn-
ing doubly stochastic affinities, the use of doubly stochastic
affinities is well-established in spectral clustering. In par-
icular, Zass and Shashua find strong performance by ap-
plying spectral clustering to the nearest doubly stochastic
matrix to the input affinity in Frobenius norm [46]. Also,
there has been work on learning doubly stochastic approx-
imations of input affinity matrix subject to rank constraints
[29, 39]. Further, if one forms the affinity matrix using a
Gaussian kernel, which is often done in spectral clustering,
Landa et al. show that doubly stochastic normalization of
the kernel matrix by diagonal matrix scaling is robust to
heteroskedastic additive noise on the data points [19].

2. Doubly Stochastic Models

In this section, we further detail the motivation for
learning doubly stochastic affinities for subspace clustering.
Then we develop two models for subspace clustering with
doubly stochastic affinities. Our J-DSSC model provides a
formulation for jointly learning a self-expressive matrix and
doubly stochastic affinity, while our A-DSSC model is a fast
approximation that sequentially solves for a self-expressive
matrix and then a doubly stochastic affinity that approxi-
mates it.

2.1. Benefits of Doubly Stochastic Affinities

As discussed above, existing subspace clustering meth-
ods such as SSC [8], LRR [22], EDSC [16], EnSC [43], and
deep subspace clustering networks (DSC-Net) [17] require
various ad-hoc postprocessing methods to achieve strong
clustering performance on certain datasets. For example,
it has recently been shown experimentally that if one re-
moves the ad-hoc postprocessing step from DSC-Net [17]
then the performance drops considerably — performing no
better than simple baseline methods and far below state-of-
the-art [11]. Common postprocessing steps include keep-
ning only the top $l$ entries of each column of $C$, normaliz-
ing columns of $C$, and/or using SVD-based postprocess-
ing, each of which introduces extra hyperparameters and
degrees of freedom for the practitioner. Likewise, the prac-
titioner is also required to make a choice on the particular
type of Laplacian normalization to use [36].

To better connect the self-expressive step with the sub-
sequent spectral clustering step and to avoid the need for
ad-hoc postprocessing methods, we propose a framework
for directly learning an affinity $A$ that already has desired
properties (A1) and (C1)-(C2) for spectral clustering. Re-
stricting $A \in \Omega_n$ to be doubly stochastic achieves these
goals. While a wide variety of regularization types and
affinity constraints have been used in self-expressive sub-
space clustering methods [24], doubly stochastic constraints
have not been leveraged.

As noted in the introduction, doubly stochastic matrices
are already nonnegative, so we do not need to take abso-
lute values of some computed matrix. Also, each row and
column of a doubly stochastic matrix sums to one, so they
each have the same scale in $l_1$ norm — removing the need
to postprocess by scaling rows or columns. Importantly for (A1), doubly stochastic matrices are also invariant to most forms of Laplacian normalization used in spectral clustering. For example, for a symmetric affinity matrix \( A \) with row sums \( D = \text{diag}(A1) \), widely used Laplacian variants include the unnormalized Laplacian \( D - A \), the normalized Laplacian \( I = D^{-1/2}AD^{-1/2} \), and the random walk Laplacian \( I - D^{-1}A \) [36]. When \( A \) is doubly stochastic, the matrix of row sums satisfies \( D = I \), so all of the normalization variants are equivalent and give the same Laplacian \( I - A \).

In addition, many types of regularization and constraints that have been proposed for subspace clustering tend to desire sparsity (C1) and connectivity (C2), in the sense that they want \( A_{ij} \) to be small in magnitude or zero when \( x_i \) and \( x_j \) belong to different subspaces and to be nonzero for sufficiently many pairs \( (i, j) \) where \( x_i \) and \( x_j \) belong to the same subspace [27, 40]. The doubly stochastic matrices learned by our models can be tuned to achieve any desired sparsity level. Also, doubly stochastic affinities are guaranteed a certain level of connectedness, as each row must sum to 1. This means that there cannot be any zero rows in the learned affinity, unlike in methods that compute representations one column at a time such as SSC [8], EnSC [43], and SSC-OMP [44], where it is possible that a point is never used in the self-expressive representation of other points.

### 2.2. Joint Learning: J-DSSC

In developing our model to jointly learn a self-expressive matrix and doubly stochastic affinity, we build off of the general regularized self-expression form in (1). In particular, in addition to learning a self-expressive matrix \( C \) we also wish to learn a doubly stochastic affinity matrix \( A \in \Omega_n \). Note, that self-expressive models (roughly) model \( C_{ij} \) as proportional to the likelihood that \( x_i \) and \( x_j \) are in the same subspace, so we desire that our normalized affinity, \( A \), be close to \( |C| \) (optionally after scaling \( A \) by a constant to match the scale of \( |C| \)), which we incorporate via the use of a penalty function \( \Theta(\cdot, \cdot) \). Thus, we have a general framework:

\[
\min_{C,A} \frac{1}{2} \| X - XC \|_F^2 + \frac{\eta_1}{2} \| C \|_F^2 + \eta_2 \| A \|_F^2 + \eta_3 \| C \|_1
\]

s.t. \( A \in \Omega_n, \text{diag}(C) = 0 \) (3)

The zero-diagonal constraint on \( C \) is enforced to prevent each point from using itself in its self-representation, as this is not informative for clustering. For our particular choice of \( \Theta \), we use an \( l_2 \) distance penalty, as \( l_2 \) projection of an affinity matrix onto the doubly stochastic matrices often improves spectral clustering performance [46] and results in desirable sparsity properties [31, 2]. As a result, our proposed model takes the form of an optimization problem over \( C \) and \( A \):

\[
\min_{C,A} \frac{1}{2} \| X - XC \|_F^2 + \frac{\eta_1}{2} \| C \|_F^2 + \frac{\eta_2}{2} \| A \|_F^2 + \eta_3 \| C \|_1
\]

s.t. \( A \in \Omega_n, \text{diag}(C) = 0 \) (4)

where \( \eta_1, \eta_2 > 0, \eta_3 \geq 0 \) are hyperparameters.

The objective (4) is not convex because of the \( \|C\| \) term in the \( C \)-to-\( A \)-difference loss. To alleviate this issue, we relax the problem by separating the self-expressive matrix \( C \) into two nonnegative matrices \( C_p, C_n \geq 0 \), so that \( C_p - C_n \) takes the role of \( C \) and \( C_p + C_n \) approximately takes the role of \( |C| \) (with the approximation being exact if the nonzero support of \( C_p \) and \( C_n \) do not overlap). Thus, our convex model is given by:

\[
\min_{C_p,C_n,A} \frac{1}{2} \| X - X(C_p - C_n) \|_F^2 + \frac{\eta_1}{2} \| C_p + C_n \|_F^2 + \frac{\eta_2}{2} \| A \|_F^2 + \eta_3 \| C_p + C_n \|_1
\]

s.t. \( A \in \Omega_n, C_p, C_n \in \mathbb{R}_{\geq 0,\text{diag}=0}^{n \times n} \) (5)

where \( \mathbb{R}_{\geq 0,\text{diag}=0}^{n \times n} \) is the set of \( n \times n \) real matrices with nonnegative entries and zero diagonal\(^1\). This problem is now convex and can be efficiently solved to global optimality, as we show in Section 3.1. We refer to this model as Joint Doubly Stochastic Subspace Clustering (J-DSSC).

If the optimal \( C_p \) and \( C_n \) have disjoint support (where the support of a matrix is the set of indices \((i, j)\) where the matrix is nonzero), then by solving (5) we also obtain a solution for (4), since we can take \( C = C_p - C_n \), in which case \( |C| = |C_p - C_n| = C_p + C_n \). However, this final equality does not hold when the optimal \( C_p \) and \( C_n \) have overlapping nonzero support. The following proposition shows that our relaxation (5) is equivalent to the original problem (4) for many parameter settings of \((\eta_1, \eta_2, \eta_3)\).

In cases where the supports of \( C_p \) and \( C_n \) overlap, we can also bound the magnitude of the overlapping entries (hence guaranteeing a close approximation of the solution to (4) from solutions to (5)). A proof is given in the appendix.

**Proposition 1.** Consider (5) with parameters \( \eta_1 > 0 \) and \( \eta_2, \eta_3 \geq 0 \). Let \((C_p^*, C_n^*, A^*)\) be an optimal solution.

1. If \( \eta_1 \eta_2 \leq \eta_3 \), then the solution of the relaxation (5) is the same as that of (4), when taking \( C = C_p - C_n \). In particular, the supports of \( C_p^* \) and \( C_n^* \) are disjoint.

2. If \( \eta_1 \eta_2 > \eta_3 \), then the supports of \( C_p^* \) and \( C_n^* \) may overlap. At any index \((i, j)\) for which \( \| (C_p^*)_{ij} \|_1 > 0 \) and \( \| (C_n^*)_{ij} \|_1 > 0 \), it holds that

\[
\max \left( \| (C_p^*)_{ij} \|_1, \| (C_n^*)_{ij} \|_1 \right) < \frac{\eta_1 \eta_2 - \eta_3}{\eta_1}
\]

\(^1\)Note that the \( l_1 \) norm \( \| C_p + C_n \|_1 \) can be replaced by the sum of the entries of \( C_p + C_n \), because of the nonnegativity constraints.
2.3. Sequential Approximation: A-DSSC

To see an alternative interpretation of the model in (4), note the following by expanding the second term of (4):

$$
(4) = \min_{\mathbf{C}, \mathbf{A}} \frac{1}{2} \| \mathbf{X} - \mathbf{XC} \|_F^2 + \frac{\eta_1}{2} \| \mathbf{C} \|_F^2 + \eta_3 \| \mathbf{C} \|_1
$$

$$
- \langle \eta_1 | \mathbf{C} \rangle, \eta_2 \mathbf{A} \rangle + \frac{\eta_1 \eta_2}{2} \| \mathbf{A} \|_F^2
$$

s.t. \( \mathbf{A} \in \Omega_n \), \( \text{diag} (\mathbf{C}) = 0 \) \hspace{1cm} (7)

From this form, one can see that for an initialization of the affinity as \( \mathbf{A} = \mathbf{I} \), the minimization w.r.t. \( \mathbf{C} \) takes the form of Elastic Net Subspace Clustering [43],

$$
\min_{\mathbf{C}} \frac{1}{2} \| \mathbf{X} - \mathbf{XC} \|_F^2 + \frac{\eta_1}{2} \| \mathbf{C} \|_F^2 + \eta_3 \| \mathbf{C} \|_1
$$

s.t. \( \text{diag} (\mathbf{C}) = 0 \) \hspace{1cm} (8)

with \( \eta_1 \) controlling the \( l_2 \) regularization on \( \mathbf{C} \) and \( \eta_3 \) controlling the \( l_1 \) regularization. Likewise, for a fixed \( \mathbf{C} \), one can observe that the above problem w.r.t. \( \mathbf{A} \) is a special case of a quadratically regularized optimal transport problem [2, 23]:

$$
\min_{\mathbf{A}} \langle -\eta_1 | \mathbf{C} \rangle, \eta_2 \mathbf{A} \rangle + \frac{\eta_1 \eta_2}{2} \| \mathbf{A} \|_F^2 \text{ s.t. } \mathbf{A} \in \Omega_n. \tag{9}
$$

Thus, we see that \( \eta_2 \) controls the regularization on \( \mathbf{A} \), with a lower value of \( \eta_2 \) encouraging sparser \( \mathbf{A} \) and higher \( \eta_2 \) encouraging denser and more uniform \( \mathbf{A} \). In particular, as \( \eta_2 \to 0 \), (9) approaches a linear assignment problem, which has permutation matrix solutions (i.e., maximally sparse solutions) [30, 14]. In contrast, as \( \eta_2 \to \infty \), the optimal solution is densely connected and approaches the uniform matrix with all entries equal to \( \frac{1}{n} \). These insights are used to develop our approximation model.

In particular, we consider an alternating minimization process to obtain approximate solutions for \( \mathbf{C} \) and \( \mathbf{A} \), where we first initialize \( \mathbf{A} \) as \( \mathbf{A} = \mathbf{I} \) and solve (8) for \( \mathbf{C} \). Then, holding \( \mathbf{C} \) fixed we solve (9) for \( \mathbf{A} \). Taking the solution to this problem as the final affinity \( \mathbf{A} \), we obtain our one-step approximation to J-DSSC, which we refer to as Approximate Doubly Stochastic Subspace Clustering (A-DSSC).

Besides being an approximation to the joint model (5), A-DSSC also has an interpretation as a postprocessing method for certain subspace clustering methods that can be expressed in the form of (8), such as SSC, EnSC, and LSR [8, 43, 25]. Instead of arbitrarily making choices about how to normalize/postprocess \( \mathbf{A} \) and form the normalized Laplacian, we instead take \( \mathbf{A} \) to be doubly stochastic; as discussed above, this provides a principled means of generating an affinity \( \mathbf{A} \) that is suitably normalized for spectral clustering and invariant to the most common types of Laplacian normalization by solving a well-motivated convex optimization problem (9).

3. Optimization and Implementation

Here we outline the optimization procedures to compute our J-DSSC and A-DSSC models. Further algorithmic and implementation details are given in the appendix.

3.1. Joint Solution

In order to efficiently solve (5), we develop an algorithm in the style of linearized ADMM [26]. Specifically, we reparameterize the problem by introducing additional variables \( \mathbf{Y} = \mathbf{A} \) and \( \mathbf{Z} = \mathbf{X} (\mathbf{C}_p - \mathbf{C}_n) \). This gives us the equivalent problem

$$
\min_{\mathbf{C}_p, \mathbf{C}_n, \mathbf{A}, \mathbf{Y}, \mathbf{Z}} \frac{1}{2} \| \mathbf{X} - \mathbf{Z} \|_F^2
$$

$$
+ \frac{\eta_1}{2} \| [\mathbf{C}_p + \mathbf{C}_n] - \eta_2 \mathbf{A} \|_F^2
$$

$$
+ \eta_3 \| \mathbf{C}_p + \mathbf{C}_n \|_1
$$

s.t. \( \mathbf{A} \geq 0, \mathbf{C}_p, \mathbf{C}_n \in \mathbb{R}_{\geq 0}^{n \times n}, \text{diag} = 0 \),

$$
\mathbf{Y}^\top \mathbf{1} = \mathbf{1}, \mathbf{Y} = \mathbf{A}, \mathbf{Z} = \mathbf{X} (\mathbf{C}_p - \mathbf{C}_n)
$$

which we solve by a variant of linearized ADMM. In particular, we take linearized ADMM steps on \( \mathbf{C}_p \) and \( \mathbf{C}_n \). Then we can efficiently minimize the associated augmented Lagrangian w.r.t. \( \mathbf{A}, \mathbf{Y}, \) and \( \mathbf{Z} \), by closed form solutions. Finally, we take dual ascent steps and repeat this process until convergence. The outline of the algorithm is given in Algorithm 1, with full details given in the appendix.

---

Footnote:

2\text{Note that additional alternating minimization steps can also be used as (7) is convex w.r.t. } \mathbf{C} \text{ if } \mathbf{A} \text{ is held fixed (for any feasible } \mathbf{A} \text{) and vice-versa.}
3.2. Approximation Solution

As previously suggested, the approximate model A-DSSC can be more efficient to solve than J-DSSC. First, the minimization over \( C \) in (8) can be solved by certain scalable subspace clustering algorithms. In general, it is equivalent to Elastic Net Subspace Clustering (EnSC), for which scalable algorithms have been developed [43]. Likewise, in the special case of \( \eta_1 > 0, \eta_3 = 0 \), it is equivalent to Least Squares Regression (LSR) [25], which can be computed by a single \( n \times n \) linear system solve.

The regularized optimal transport problem (9) to solve for \( A \) can be solved through the dual [2, 23], which gives an unconstrained problem over two vectors in \( \alpha, \beta \in \mathbb{R}^n \):

\[
\max_{\alpha, \beta} -1^\top (\alpha + \beta) - \frac{1}{2\eta_2} \left\| \left[ |C| - \alpha 1^\top - 1\beta^\top \right]_+ \right\|^2_2 \quad (11)
\]

where \([\cdot]_+\) denotes half-wave rectification, meaning that \( |x|_+ = \max\{0, x\} \), applied entry-wise. The optimal matrix \( A \) is then recovered as:

\[
A = \frac{1}{\eta_2} \left[ |C| - \alpha 1^\top - 1\beta^\top \right]_+ \quad (12)
\]

We use L-BFGS [21] to solve for \( \alpha \) and \( \beta \) in (11). The outline of the algorithm to compute A-DSSC is given in Algorithm 2, with full details in the appendix.

3.3. Model Selection

Our method has three hyperparameters \( \eta_1, \eta_2, \eta_3 \) to set. If we fix \( A \), the problem is simply EnSC [43]. This means that we may choose the parameters \( \eta_1 \) and \( \eta_3 \) in J-DSSC as is done in EnSC. As, noted in Section 2.2, \( \eta_2 \) controls the sparsity of the final \( A \), with smaller \( \eta_2 \) tending to give sparser \( A \). Thus, we choose \( \eta_2 \) such that \( A \) is within a desired average sparsity range. In addition, to avoid solutions that are too sparse, we can consider the number of connected components of the graph associated to the solution \( A \). If we want to obtain \( k \) clusters, we can adjust \( \eta_2 \) so that the number of connected components is at most \( k \). The parameters for A-DSSC can be selected in an analogous manner.

4. Experiments

In this section, we empirically study our models J-DSSC and A-DSSC. We show that our algorithm achieves state-of-the-art subspace clustering performance on a variety of real datasets with multiple performance metrics. Also, we investigate the behavior of our models on small datasets, showing various desirable properties of A-DSSC and J-DSSC.

4.1. Experimental Setup

Algorithms. We compare against several state-of-the-art subspace clustering algorithms, which are all affinity-based: SSC [7, 8], EnSC [43], LSR [25], LRSC [35], TSC [13], SSC-OMP [44], S\(^3\)COMP [4], and DSC-Net [17]. For these methods, we run the experiments on our own common framework, and we note that the results we obtain are very similar to those that have been reported previously in the literature on the same datasets for these methods. However, we do not have access to an implementation of S\(^3\)COMP [4], so we report partial results from their paper, which we still include due to it being a recent well-performing model. Besides TSC, which uses an inner product similarity between data points, these methods all compute an affinity based on some self-expressive loss.

Metrics. As is standard in evaluations of clustering, we use clustering accuracy (ACC) [44] and normalized mutual information (NMI) [18] metrics, where we take the denominator in NMI to be the arithmetic average of the entropies. Also, we consider a subspace-preserving error (SPE), which is given by \( \frac{1}{n} \sum_{i=1}^n \sum_{j \in S_{y_i}} |A_{ij}| / \|A_j\|_1 \), where \( S_{y_i} \) denotes the (subspace) cluster that point \( x_i \) belongs to, and \( A_{ij} \) is the \( i \)th column of \( A \). This measures the proportion of mass in the affinity that is erroneously given to points in different (subspace) clusters, and is often used in evaluation of subspace clustering algorithms [44, 33]. We report the sparsity of learned affinities by the average number of nonzeros per column (NNZ). Although sparse affinities are generally preferred, the sparsest affinity is not necessarily the best.

Datasets. We apply these algorithms for subspace clustering on the Extended Yale-B face dataset [9], Columbia Object Image Library (COIL-40 and COIL-100) [28], and UMIST face dataset [10]. We run experiments with the raw pixel data from the images and for the COIL and UMIST datasets also run separate experiments with features obtained from a scattering convolution network [3] reduced to dimension 500 by PCA [41]. This has been used in previous subspace clustering works [43, 4] to better embed the raw pixel data in a union of linear subspaces (whereas the raw pixel data does not necessarily have a union of subspaces structure). We do not use scattering transform features on the Yale-B dataset as it is already assumed to be well approximated by a union of linear subspaces [1]. Also, we do not evaluate DSC-Net on scattered features as the network architecture is only compatible with image data.

Clustering setup. For each baseline method we learn the \( C \) from that method, form the affinity as \( \hat{A} = (|C| + |C|_1^\top) / 2 \), take the normalized Laplacian \( L = I - D^{-1/2}AD^{-1/2} \) (where \( D = \text{diag}(\hat{A}1) \)), and then apply spectral clustering with the specified number \( k \) of clusters [36]. For our DSSC methods, we form \( \hat{A} = (A + A^\top) / 2 \) and use \( L = I - \hat{A} \) directly for spectral clustering (recall our affinity is invariant to Laplacian normalization). For the spectral clustering we take the \( k \) eigenvectors corresponding to the \( k \) smallest eigenvalues of \( L \) as an embedding of the data. After normal-
Figure 2. Affinity matrices $A$ learned by J-DSSC on the first five classes of Yale-B. We fix $\eta_1 = .25$ and $\eta_3 = 0$ (so there is no $l_1$ regularization on $C$), while varying $\eta_2$. Red dashed lines mark the boundaries between classes. Dark blue points mark nonzero entries in $A$. The learned affinity $A$ is sparser as $\eta_2$ decreases, while it is more connected as $\eta_2$ increases.

Figure 3. Objective value and clustering accuracy for different number of alternating minimization steps over $C$ in (8) then $A$ in (9) on the UMIST dataset. Our A-DSSC model is equivalent to 1 alternating minimization step. (Left) Shows relative error in the objective value compared to the true minimum. (Right) Shows clustering accuracy of using either $A$ or $C$ in spectral clustering.

doubly stochastic postprocessing, outperforms SSC, EnSC, and LSR across all datasets. This suggests that the doubly stochastic affinity matrices learned by our models are indeed effective as inputs into spectral clustering for the task of subspace clustering, and that our method can also potentially be combined with other subspace clustering methods as a principled postprocessing and normalization step.

Beyond just clustering accuracy, our models also learn affinity matrices with other desirable properties. The SPE of our models are usually the lowest among all methods, meaning that our learned affinities do not have significant misplaced mass connecting points from different subspaces. Likewise, the learned affinities tend to be rather sparse, with on average less than 15 nonzeros per column for the chosen parameters across all datasets. At the same time, the learned affinities have sufficient connectivity, as shown by the strong clustering accuracy.

4.3. Behavior of DSSC Models

Here, we explore the quality of our approximation A-DSSC by comparing the natural extension of this approximation to multiple steps of alternating minimization over $C$ in (8) and $A$ in (9). In particular, we note that additional alternating minimization steps do not bring a significant benefit over just a single alternating minimization iteration. Figure 3 displays the results of our A-DSSC model on
Table 1. Subspace clustering results on different datasets. ‘—’ indicates that we do not have results or the method cannot be run for the given entry. Best results with respect to ACC and NMI are bolded.

| Dataset   | Metric | Shallow Affinity-Based | Neural | Doubly Stochastic |
|-----------|--------|------------------------|--------|-------------------|
|           |        | SSC | EnSC | LSR | LSRSC | TSC | SSC-OMP | S3COMP | DSC-Net | DSC-Net-PP | J-DSSC | A-DSSC |
| Yale-B    | ACC    | .654 | .616 | .659 | .662 | .514 | .780 | .874 | .691 | .971 | .924 | .917 |
|           | NMI    | .734 | .689 | .743 | .739 | .629 | .844 | — | .746 | .961 | .952 | .947 |
|           | SPE    | .217 | .249 | .869 | .875 | .217 | .179 | .203 | .881 | .038 | .080 | .080 |
|           | NZN    | 22.9 | 29.2 | 2413 | 2414 | 4.3 | 9.1 | — | 2414 | 22.1 | 14.5 | 14.4 |
| COIL-40   | ACC    | .799 | .649 | .577 | .617 | .617 | .617 | .617 | .617 | .617 | .617 | .617 |
|           | NMI    | .940 | .826 | .761 | .736 | .605 | .605 | .605 | .605 | .605 | .605 | .605 |
|           | SPE    | .034 | .103 | .926 | .877 | .057 | .057 | .057 | .057 | .057 | .057 | .057 |
|           | NZN    | 3.1  | 31.5 | 2879 | 2880 | 4.5 | 2.9 | — | 2880 | 13.1 | 8.9  | 5.2  |
| COIL-100  | ACC    | .704 | .561 | .492 | .476 | .723 | .313 | .789 | .493 | .635 | .796 | .824 |
|           | NMI    | .919 | .816 | .753 | .733 | .904 | .588 | — | .752 | .875 | .943 | .946 |
|           | SPE    | .052 | .163 | .945 | .946 | .057 | .052 | .032 | .958 | .384 | .049 | .037 |
|           | NZN    | 2.2  | 25.4 | 7199 | 7200 | 3.6 | 3.0 | — | 7200 | 44.7 | 9.9  | 5.8  |
| UMIST     | ACC    | .537 | .499 | .462 | .494 | .661 | .509 | — | .456 | .708 | .732 | .725 |
|           | NMI    | .718 | .681 | .645 | .662 | .829 | .680 | — | .611 | .848 | .858 | .851 |
|           | SPE    | .079 | .254 | .811 | .824 | .035 | .091 | — | .834 | .393 | .036 | .034 |
|           | NZN    | 4.9  | 25.6 | 7199 | 7200 | 3.7 | 2.9 | — | 7200 | 23.2 | 5.1  | 4.8  |
| UMIST     | ACC    | .764 | .757 | .724 | .745 | .745 | .745 | .745 | .745 | .745 | .745 | .745 |
|           | NMI    | .834 | .747 | .701 | .711 | .855 | .416 | — | — | .939 | .935 |
|           | SPE    | .038 | .151 | .758 | .804 | .030 | .252 | — | — | 9.33 | 9.35 |
|           | NZN    | 2.7  | 14.9 | 574  | 575  | 3.8 | 9.7 | — | 7.2  | 5.4  | — | — |

We also investigate the sparsity and connectivity of the affinity matrix learned by our models. In Figure 2, we show sparsity plots of the learned matrices A for J-DSSC models with varying $\eta_2$ on the first five classes on the Extended Yale-B dataset [9]. The points are sorted by class, so intra-subspace connections are on the block diagonal, and inter-subspace connections are on the off-diagonal blocks. Here, it can be seen that the sparsity level of the recovered A affinity is controlled by choice of the $\eta_2$ parameter even when there is no sparse regularization on C (i.e., $\eta_3 = 0$), with the number of nonzeros increasing as $\eta_2$ increases.

5. Conclusion

In this work, we propose a method to unify the self-expressive modeling of subspace clustering with the subsequent spectral clustering based on learning a doubly stochastic affinity matrix. We demonstrate experimentally that our methods achieve substantial improvements in subspace clustering accuracy over state-of-the-art models, without using any of the ad-hoc postprocessing steps that many other methods rely on. Our work further suggests many promising directions for future research by better merging the self-expressive modeling of subspace clustering with spectral clustering (or other graph clustering).

Acknowledgements. We thank Tianjiao Ding for helpful discussions. This work was partially supported by the Northrop Grumman Mission Systems Research in Applications for Learning Machines (REALM) initiative, the Johns Hopkins University Research Experience for Undergraduates in Computational Sensing and Medical Robotics (CSMR REU) program, NSF 1704458, and NSF 1934979.
Appendices

A. Derivation of A-DSSC Dual

Here, we give a derivation of the dual (11) of the minimization over \( A \) in A-DSSC. To see the derivation of the dual problem, first recall that the primal problem is:

\[
\arg\min_A \left( -\eta_1 |C|, \eta_2 A \right) + \frac{\eta_2}{2} \|A\|_F^2 \quad \text{s.t.} \quad A \in \Omega_n
\]  

(13)

Introducing Lagrange multipliers \( \alpha \) and \( \beta \in \mathbb{R}^n \) for the row and column sum constraints, we have the equivalent problem:

\[
\min_{A \geq 0, \alpha, \beta \in \mathbb{R}^n} \max_{\|A\|_F \leq 1} \left( -|C|, A \right) + \frac{\eta_2}{2} \|A\|_F^2 \\
+ \langle \alpha, A 1 - 1 \rangle + \langle \beta, A^T 1 - 1 \rangle
\]  

(15)

As the primal problem is convex and has a strictly feasible point, strong duality holds, so this is equivalent to:

\[
\max_{\alpha, \beta \in \mathbb{R}^n} \min_{A \geq 0} \left( -|C|, A \right) + \frac{\eta_2}{2} \|A\|_F^2 \\
+ \langle \alpha, A 1 - 1 \rangle + \langle \beta, A^T 1 - 1 \rangle
\]  

(16)

\[
= \max_{\alpha, \beta \in \mathbb{R}^n} -\left( \alpha + \beta \right)^T 1 + \min_{A \geq 0} \left( -|C|, A \right) + \frac{\eta_2}{2} \|A\|_F^2 \\
+ \langle \alpha 1^T + \beta A^T, A \rangle
\]  

(17)

Letting \( K = |C| - \alpha 1^T - \beta^T \), the inner minimization takes the form:

\[
\min_{A \geq 0} \left( -K, A \right) + \frac{\eta_2}{2} \|A\|_F^2
\]  

(18)

\[
= \eta_2 \min_{A \geq 0} \left( -\frac{1}{\eta_2} K, A \right) + \frac{1}{2} \|A\|_F^2
\]  

(19)

\[
= -\frac{1}{2\eta_2} \|K\|_F^2 + \eta_2 \min_{A \geq 0} \left( \frac{1}{\eta_2} K - A \right) F^2
\]  

(20)

\[
= -\frac{1}{2\eta_2} \|K\|_F^2 + \frac{1}{2\eta_2} \|K\|_F^2
\]  

(21)

\[
= -\frac{1}{\eta_2} \|K\|_F^2
\]  

(22)

Where \([K]_+ = \min\{K, 0\}\) taken elementwise. Thus, the final version of the dual is

\[
\max_{\alpha, \beta \in \mathbb{R}^n} -\left( \alpha + \beta \right)^T 1 - \frac{1}{2\eta_2} \left\| \left| C \right| - \alpha 1^T - \beta^T \right\|_F^2
\]  

(23)

and the optimal value of \( A \) is as given as

\[
A = \frac{1}{\eta_2} \left[ \left| C \right| - \alpha 1^T - \beta^T \right]_+
\]  

(24)

Algorithm 3 J-DSSC

**Input:** Data matrix \( X \), parameters \( \eta_1, \eta_2, \eta_3 \), step sizes \( \rho \) and \( \tau \)

**while** Not converged **do**

Update \( C_p \) and \( C_n \) by linearized ADMM steps (28) to (31)

Update \( A, Y, \) and \( Z \) by minimization steps (32) to (36)

Update \( \lambda_1, \lambda_2, \lambda_1 \), and \( \lambda_2 \) by dual ascent steps (37) to (40)

**end while**

Apply spectral clustering on Laplacian \( I - \frac{1}{2}(A + A^T) \)

**Output:** Clustering result

B. Algorithm Details

B.1. Linearized ADMM for J-DSSC

Here, we give details on the computation of J-DSSC. Consider the equivalent problem (10) with the added variables \( Y = A \) and \( Z = X[C_p - C_n] \). The augmented Lagrangian takes the form:

\[
\mathcal{L}(C_p, C_n, A, Y, Z, \lambda, \Lambda) = \frac{1}{2} \|X - Z\|_F^2 \\
+ \frac{\eta_1}{2} \left[ \|C_p + C_n\|_F^2 + \eta_3 \|C_p + C_n\|_1 \right]
\]

\[
+ \langle \lambda_1, Y^T 1 - 1 \rangle + \langle \lambda_2, Y 1 - 1 \rangle + \langle A_1, Y - A \rangle
\]

\[
+ \frac{\rho}{2} \left[ \|Y^T 1 - 1\|_F^2 + \frac{\rho}{2} \|Y 1 - 1\|_F^2 + \frac{\rho}{2} \|Y - A\|_F^2 \right]
\]

\[
+ \langle A_2, Z - X[C_p - C_n] \rangle + \frac{\rho}{2} \|Z - X[C_p - C_n]\|_F^2
\]

\[
+ \sum_{i \geq 0} (A) + \sum_{i \geq 0, \text{diag} = 0} (C_p) + \sum_{i \geq 0, \text{diag} = 0} (C_n)
\]

(25)

where \( I_s \) is the indicator function for the set \( S \), which takes values of 0 in \( S \) and \( \infty \) outside of \( S \), and where \( \rho > 0 \) is a chosen constant.

In each iteration, we take a linearized ADMM step in \( C_p \) and \( C_n \), then alternatingly minimize over \( A, Y, \) and \( Z, \) and lastly take gradient ascent steps on each of the Lagrange multipliers. For a step size \( \tau > 0 \), the linearized ADMM step over \( C_p \) takes the form of a gradient descent step on

\[
h(C_p) = \langle A_2, Z - X[C_p - C_n] \rangle + \frac{\rho}{2} \|Z - X[C_p - C_n]\|_F^2,
\]

(26)

followed by application of a proximal operator. Letting \( C_p' \) be the intermediate value of \( C_p \) after the gradient descent step, \( C_p \) is updated as the solution to

\[
\min_{C_p} \frac{\eta_1}{2} \|C_p + C_n\|_F^2 + \eta_3 \|C_p + C_n\|_1 \\
+ \frac{1}{2\tau} \|C_p' - C_p\|_F^2 + \sum_{i \geq 0, \text{diag} = 0} (C_p)
\]

(27)

For a matrix \( E \), let \([E]_+ \) be the half-wave rectification of \( E \), and let \([E]_{+d=0} \) be the matrix \( E \) with all negative entries and the diagonal set to zero. Then the linearized ADMM updates are given
as follows:

\[
C_p' \leftarrow C_p - \tau \left( -X^T A_2 + \rho X^T (X[C_p - C_n] - Z) \right) \tag{28}
\]

\[
C_p \leftarrow \left[ \frac{1}{\eta_1 + \tau} \left( \frac{1}{\tau} C_p' - \eta_1 C_n + \eta_1 \eta_2 A - \eta_3 11^T \right) \right] + d \rightarrow 0 \tag{29}
\]

\[
C_n' \leftarrow C_n - \tau (X^T A_2 - \rho X^T (X[C_p - C_n] - Z)) \tag{30}
\]

\[
C_n \leftarrow \left[ \frac{1}{\eta_1 + \tau} \left( \frac{1}{\tau} C_n' - \eta_1 C_p + \eta_1 \eta_2 A - \eta_3 11^T \right) \right] + d \rightarrow 0 \tag{31}
\]

The sequential minimizations over \( A \) then \( Y \) then \( Z \), holding the other variables fixed and using the most recent value of each variable, take the form:

\[
A \leftarrow \left[ \frac{1}{\eta_1 \eta_2 + \rho} \left( \eta_1 \eta_2 [C_p + C_n] + A_1 + \rho Y \right) \right] + \varepsilon \tag{32}
\]

\[
Y \leftarrow \frac{1}{\rho} \left( V - \frac{1}{n + 1} PV11^T - 11^T V \frac{1}{n + 1} P \right) \tag{33}
\]

where \( V = \rho A + 2\rho 11^T - 1\lambda_1^2 - \lambda_2 11^T - A \)

\[
P = I - \frac{1}{2n + 1} 11^T \tag{34}
\]

\[
Z \leftarrow \frac{1}{1 + \rho} \left( X - A_2 + \rho X[C_p - C_n] \right) \tag{36}
\]

The dual ascent steps on the Lagrange multipliers take the form:

\[
\lambda_1 \leftarrow \lambda_1 + \rho (Y^T 1 - 1) \tag{37}
\]

\[
\lambda_2 \leftarrow \lambda_2 + \rho (Y 1 - 1) \tag{38}
\]

\[
\Lambda_1 \leftarrow \Lambda_1 + \rho (Y - A) \tag{39}
\]

\[
\Lambda_2 \leftarrow \Lambda_2 + \rho (Z - X[C_p - C_n]) \tag{40}
\]

Thus, the full J-DSSC algorithm is given in Algorithm 3.

To derive the minimization steps for a variable \( E \) that is either \( C_p, C_n, A, \) or \( Z \), we use the method of completing the square to change the corresponding minimization into one of the form \( \min_{E \in S} \| E - E' \|_F^2 \) for some feasible set \( S \) and some matrix \( E' \).

For instance, in the case of \( C_p \), the set \( S \) is the set of \( n \times n \) nonnegative matrices with zero diagonal, and the solution is \( E' \) for \( E \).

To minimize over \( Y \), we use different methods often used in dealing with row or column sum constraints [46, 39]. Define \( V = \rho A + 2\rho 11^T - 1\lambda_1^2 - \lambda_2 11^T - A \). Setting the gradient of the augmented Lagrangian with respect to \( Y \) equal to zero, we can show that

\[
\rho Y + \rho Y 11^T + \rho 11^T Y = V \tag{41}
\]

Multiplying both sides by \( 1 \) on the right and noting that \( 1^T 1 = n \), we have

\[
\rho (I + nI + 11^T) Y 1 = V 1 \tag{42}
\]

\[
\rho Y 1 = (I + nI + 11^T)^{-1} V 1 \tag{43}
\]

Likewise, multiplying by \( 1^T \) on the left gives that

\[
\rho 1^T Y = 1^T V (I + nI + 11^T)^{-1} \tag{44}
\]

We use the Woodbury matrix identity to compute the inverse

\[
(I + nI + 11^T)^{-1} = \frac{1}{n + 1} \left( I - \frac{1}{2n + 1} 11^T \right). \tag{45}
\]

We define \( P = I - \frac{1}{2n + 1} 11^T \), so the inverse is \( I - P \). Thus, using both (43) and (44) in the left hand side of (41) gives that

\[
\rho Y + \frac{1}{n + 1} PV11^T + \rho 11^T V 1 + 1 P = V \tag{46}
\]

and thus the minimizing \( Y \) is given as

\[
Y = \frac{1}{\rho} \left[ V - \frac{1}{n + 1} PV11^T - 11^T V \frac{1}{n + 1} P \right]. \tag{47}
\]

B.2. Implementation Details

We implement our methods in the Julia programming language. For the J-DSSC algorithm, we take the step sizes to be \( \rho = .5 \) and \( \tau = .0001 \) for our experiments.

Complexity Analysis. Let \( X \in \mathbb{R}^{d \times n} \), so there are \( n \) data points in \( \mathbb{R}^d \). For J-DSSC, each update on \( C_p \) and \( C_n \) takes \( O(d n^2) \) operations due to matrix multiplications. Each update on \( A \) and \( Y \) takes \( O(n^2) \) operations, while each update on \( Z \) takes \( O(d n^2) \). The dual ascent steps also take \( O(d n^2) \) operations in total. Thus, each iteration of the linearized ADMM part takes \( O(d n^2) \). We note in particular that J-DSSC does not require the solution of any linear systems or any singular value decompositions, while many other subspace clustering methods do. For A-DSSC, solving for \( C \) is done by some efficient EnSC, SSC, or LSR solver. The quadratically regularized optimal transport step takes \( O(n^2) \) operations per objective function evaluation and gradient evaluation.

The spectral clustering step requires a partial eigendecomposition of \( I - \frac{1}{2} (A + A^T) \), although we note that in practice the matrix that is learned tends to be sparse for the parameter settings we choose, so we can leverage sparse eigendecomposition algorithms to compute the \( k \) eigenvectors corresponding to the smallest eigenvalues. Recall that all subspace clustering methods that use spectral clustering (which is the majority of methods) require computing an eigendecomposition of equivalent size, though not all subspace clustering affinities can leverage sparse eigendecomposition methods.

C. Proof of Proposition 1

Proof. Suppose that there is an index \((i, j)\) such that

\[
a = (C_p^*)_{ij} > 0, \quad b = (C_n^*)_{ij} > 0. \tag{48}
\]

Define matrices \( C_p' \) and \( C_n' \) such that \( C_p' \) matches \( C_p^* \) at every index but \((i, j)\) and similarly for \( C_n' \). At index \((i, j)\), we define

\[
(C_p')_{ij} = \max(a - b, 0), \quad (C_n')_{ij} = \max(b - a, 0) \tag{49}
\]

We consider the value of the expanded J-DSSC objective (7) at \((C_p', C_n', A^*)\) compared to the value at \((C_p^*, C_n^*, A^*)\). Note that \( C_p' - C_n' = C_p^* - C_n^* \). Thus, there are only three summands in the objective that are different between the two groups of variables.
Suppose $a \geq b$. The differences are computed as:

$$\frac{\eta_1}{2} \left| C_p + C_n \right|_F^2 - \frac{\eta_1}{2} \left| C_p + C_n' \right|_F^2$$

$$= \frac{\eta_1}{2} \left( |a - b|^2 - (a + b)^2 \right)$$

$$= -2\eta_1 ab - \eta_1 \left| C_p + C_n' \right|_F^2$$

$$= \eta_1 \eta_2 (A^*)_{ij} \left( -|a - b| + (a + b) \right)$$

$$= 2\eta_1 \eta_2 (A^*)_{ij} b$$

$$\eta_3 \left| C_p + C_n' \right|_1 - \eta_3 \left| C_p + C_n' \right|_1$$

$$= \eta_3 \left( |a - b| - (a + b) \right)$$

$$= -2\eta_1 b$$

Now, we bound the difference between the objective functions for the two variables:

$$0 < -2\eta_1 ab + 2\eta_1 \eta_2 (A^*)_{ij} b - 2\eta_3 b$$

$$\leq -2\eta_1 ab + 2\eta_1 \eta_2 b - 2\eta_3 b$$

where the first bound is due to $(C_p + C_n', A^*)$ being the optimal solution, and the upper bound is due to $\left( A^* \right)_{ij} \leq 1$. From these bounds, we have that

$$b \leq a < \frac{\eta_1 \eta_2 - \eta_3}{\eta_3}.$$ (53)

Where we recall that we assumed $b \leq a$. A similar upper bound holds with $b$ taking the place of $a$ in the case where $b > a$. Now, note that $a > 0$ means that $\eta_1 \eta_2 > \eta_3$, thus proving 1) through the contrapositive. This bound clearly gives the one in (6), so we have also proven 2).

D. Data Details

Figure 4. Sample images from Extended Yale-B (top row), COIL-100 (middle row), and UMIST (bottom row).

Table 2. Statistics and properties of datasets.

| Dataset   | # Images ($n$) | Dimension | # Classes |
|-----------|----------------|-----------|-----------|
| Yale-B    | 2414           | $48 \times 42$ | 38        |
| COIL-40   | 2880           | $32 \times 32$ | 40        |
| COIL-100  | 7200           | $32 \times 32$ | 100       |
| UMIST     | 575            | $32 \times 32$ | 20        |

For our experiments, we run subspace clustering on the Extended Yale-B face dataset [9], Columbia Object Image Library (COIL-40 and COIL-100) [28], and UMIST face dataset [10]. Sample images from these datasets are shown in Figure 4. Basic statistics and properties of these datasets are given in Table 2.

The scattered data is computed as in [43, 44], by using a scattering convolution network [3] to compute a feature vector of $3,472$ dimensions for UMIST and $3 \cdot 3,472 = 10,416$ dimensions for COIL (since there are three color channels), then projecting to 500 dimensions by PCA. For both pixel data and scattered data, we normalize each data point to have unit $L_2$ norm for all experiments, noting that scaling a data point does not change its subspace membership.

For Yale-B, we remove the 18 images labeled as corrupted in the dataset, leaving 2414 images for our experiments. Adding the corrupted images back in does not appear to qualitatively change our results. We resize Yale-B to size $32 \times 42$. The COIL images and UMIST images are resized to size $32 \times 32$.

E. Parameter Settings

| Table 3. Parameter settings for J-DSSC and A-DSSC. |
|-----------------------------------|
| Dataset                  | $\eta_1$ | $\eta_2$ | $\eta_3$ |
|---------------------------|---------|---------|---------|
| Yale-B                    | .25     | .2      | 0       |
| COIL-40                   | .25     | .01     | .1      |
| COIL-40 (Scattered)       | .25     | .2      | 0       |
| COIL-100                  | .25     | .01     | .1      |
| COIL-100 (Scattered)      | .25     | .1      | 0.025   |
| UMIST                     | 1       | .05     | .05     |
| UMIST (Scattered)         | .01     | .2      | 0       |

| Table 4. Parameters searched over for different methods. |
|----------------------------------------------------------|
| Method                          | Parameters and Variant |
| SSC [7]                         | $\gamma \in \{1, 5, 10, 25, 50, 100, 1000\}$, noisy data variant |
| EnSSC [43]                      | $\gamma \in \{1, 1.5, 5, 10, 50, 100, 200\}$, $\lambda = .5$ |
| LSR [25]                        | $\lambda \in \{.01, .1, .5, 1, 10, 50, 100\}$, diag($\mathbf{C}$) = 0 variant |
| LRSC [35]                       | $\tau \in \{1, 1.1, 10, 50, 100, 500, 1000\}$, $\alpha = \tau/2$, noisy data and relaxed constraint variant |
| TSC [13]                        | $q \in \{2, 3, \ldots, 14, 15\}$ |
| SSC-OMP [44]                   | $k_{\text{max}} \in \{2, 3, \ldots, 14, 15\}$ |
| J-DSSC (Ours)                   | $\eta_1 \in \{.01, .25, 1, 25\}$, $\eta_2 \in \{.01, .05, .1, .2\}$, $\eta_3 \in \{0, .1\}$ |
| A-DSSC (Ours)                   | $\eta_1 \in \{1, 1.1, 10, 25, 50\}$, $\eta_2 \in \{.0005, .001, .01, .025, .05, .1\}$, $\eta_3 = 0$ |

Here, we give parameter settings for the experiments in the paper. The chosen parameter settings for our DSSC models are given

13
in Table 3. We note that the experiments on UMIST to explore the objective value over iterations of alternating minimization in Figure 3 use the same parameter settings as in this table.

The parameters searched over for the non-neural methods are given in Table 4.

For DSC-Net, we use the hyperparameters and pre-trained autoencoders as in their original paper [17] for Yale-B and COIL-100. For DSC-Net on UMIST, we take a similar architecture to that of [47], with encoder kernel sizes of $5 \times 5, 3 \times 3, 3 \times 3$, as well as 15, 10, 5 channels per each layer. The regularization parameters are set as $\lambda_1 = 1$ and $\lambda_2 = 2$. The DSC-Net post-processing on UMIST is taken to be similar to that used for ORL. For COIL-40, we use one encoder layer of kernel size $3 \times 3$, 20 channels, and regularization parameters $\lambda_1 = 1, \lambda_2 = 100$; we also use the same postprocessing that is used for COIL-100.