A CRITIQUE OF SELF-EXPRESSIVE DEEP SUBSPACE CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Subspace clustering is an unsupervised clustering technique designed to cluster data that is supported on a union of linear subspaces, with each subspace defining a cluster with dimension lower than the ambient space. Many existing formulations for this problem are based on exploiting the self-expressive property of linear subspaces, where any point within a subspace can be represented as linear combination of other points within the subspace. To extend this approach to data supported on a union of non-linear manifolds, numerous studies have proposed learning an appropriate kernel embedding of the original data using a neural network, which is regularized by a self-expressive loss function on the data in the embedded space to encourage a union of linear subspaces prior on the data in the embedded space. Here we show that there are a number of potential flaws with this approach which have not been adequately addressed in prior work. In particular, we show the optimization problem is often ill-posed in multiple ways, which can lead to a degenerate embedding of the data, which need not correspond to a union of subspaces at all. We validate our theoretical results experimentally and additionally repeat prior experiments reported in the literature, where we conclude that a significant portion of the previously claimed performance benefits can be attributed to an ad-hoc post processing step rather than the clustering model.

1 INTRODUCTION AND BACKGROUND

Subspace clustering is a classical unsupervised learning problem, where one wishes to segment a given dataset into a prescribed number of clusters, and each cluster is defined as a linear (or affine) subspace with dimension lower than the ambient space. There have been a wide variety of approaches proposed in the literature to solve this problem Vidal et al. (2016), but a large family of state-of-the-art approaches are based on exploiting the self-expressive property of linear subspaces. That is, if a point lies in a linear subspace, then it can be represented as a linear combination of other points within the subspace. Based on this fact, a wide variety of methods have been proposed which, given a dataset $\mathbf{Z} \in \mathbb{R}^{d \times N}$ of N d-dimensional points, find a matrix of coefficients $\mathbf{C} \in \mathbb{R}^{N \times N}$ by solving the problem:

$$\min_{\mathbf{C} \in \mathbb{R}^{N \times N}} \left\{ F(\mathbf{Z}, \mathbf{C}) \equiv \frac{1}{2} \|\mathbf{Z}\mathbf{C} - \mathbf{Z}\|_F^2 + \lambda \theta(\mathbf{C}) = \frac{1}{2} \langle \mathbf{Z}^\top \mathbf{Z}, (\mathbf{C} - \mathbf{I})(\mathbf{C} - \mathbf{I})^\top \rangle + \lambda \theta(\mathbf{C}) \right\}. \quad (1)$$

Here, the first term $\|\mathbf{Z}\mathbf{C} - \mathbf{C}\|_F^2$ captures the self-expressive property by requiring every datapoint to represent itself as an approximate linear combination of other points, i.e., $\mathbf{Z}_i \approx \mathbf{Z}\mathbf{C}_i$, where $\mathbf{Z}_i$ and $\mathbf{C}_i$ are the ith columns of $\mathbf{Z}$ and $\mathbf{C}$, respectively. The second term, $\theta(\mathbf{C})$, is some regularization function designed to encourage each data point to only select other points within the correct subspace in its representation and to avoid trivial solutions (such as $\mathbf{C} = \mathbf{I}$). Once the $\mathbf{C}$ matrix has been solved for, one can then define a graph affinity between data points, typically based on the magnitudes of the entries of $\mathbf{C}$, and use an appropriate graph-based clustering method (e.g., spectral clustering von Luxburg (2007)) to produce the final clustering of the data points.

One of the first methods to utilize this approach was Sparse Subspace Clustering (SSC) Elhamifar & Vidal (2009; 2013), where θ takes the form $\theta_{SSC}(\mathbf{C}) = \|\mathbf{C}\|_1 + \delta(\text{diag}(\mathbf{C}) = \mathbf{0})$, with $\|\cdot\|_1$ denoting the ℓ_1 norm and δ an indicator function which takes value ∞ if an element of the diagonal of $\mathbf{C}$ is non-zero and 0 otherwise. By regularizing $\mathbf{C}$ to be sparse, a point must represent itself using

the smallest number of other points within the dataset, which in turn ideally requires a point to only select other points within its own subspace in the representation. Likewise other variants, with Low-Rank Representation (LRR) Liu et al. (2013) and Elastic-net Subspace Clustering (EnSC) You et al. (2016) being well-known examples, take the same form as (1) with different choices of regularization, $\theta_{LRR}(\mathbf{C}) = \|\mathbf{C}\|_*$ and $\theta_{EnSC}(\mathbf{C}) = \|\mathbf{C}\|_1 + \tau\|\mathbf{C}\|_F^2 + \delta(\text{diag}(\mathbf{C}) = \mathbf{0})$, respectively, where $\|\cdot\|_*$ denotes the nuclear norm (sum of the singular values). A significant advantage of the majority of these methods is that it can be proven (typically subject to some technical assumptions regarding the angles between the underlying subspaces and the distribution of the sampled data points within the subspaces) that the optimal $\mathbf{C}$ matrix in (1) will be "correct" in the sense that if $\mathbf{C}_{i,j}$ is non-zero then the ith and j^{th} columns of $\mathbf{Z}$ lie in the same linear subspace Soltanolkotabi & Candès (2012); Lu et al. (2012); Elhamifar & Vidal (2013); Soltanolkotabi et al. (2014); Wang et al. (2015); Wang & Xu (2016); You & Vidal (2015a;b); Tsakiris & Vidal (2018); Li et al. (2018); Robinson et al. (2019), which has led to these methods achieving state-of-the-art performance in many applications.

1.1 Self-Expressive Deep Subspace Clustering

Although subspace clustering techniques based on self-expression display strong empirical performance and provide theoretical guarantees, a significant limitation of these techniques is the requirement that the underlying dataset needs to be approximately supported on a union of linear subspaces. This has led to a strong motivation to extend these techniques to more general datasets, such as data supported on a union of non-linear low-dimensional manifolds. From inspection of the right side of (1), one can observe that the only dependence on the data $\mathbf{Z}$ comes in the form of the Gram matrix $\mathbf{Z}^\top\mathbf{Z}$. As a result, self-expressive subspace clustering techniques are amendable to the "kernel-trick", where instead of taking an inner product kernel between data points, one can instead use a general kernel $\kappa(\cdot,\cdot)$ Patel & Vidal (2014); Patel et al. (2015). Of course, such an approach comes with the traditional challenge of how to select an appropriate kernel so that the embedding of the data in the Hilbert space associated with the choice of kernel results in a union of linear subspaces in the embedded space.

The first approach to propose learning an appropriate embedding of an initial dataset $\mathbf{X} \in \mathbb{R}^{d_x \times N}$ (which does not necessarily have a union of subspaces structure) was given by Patel et al. (2013); Patel & Vidal (2014); Patel et al. (2015) who proposed first projecting the data into a lower dimensional space via a learned linear projector, $\mathbf{Z} = \mathbf{P}_l\mathbf{X}$, where $\mathbf{P}_l \in \mathbb{R}^{d \times d_x}$ $(d < d_x)$ is also optimized over in addition to $\mathbf{C}$ in (1). To ensure that sufficient information about the original data $\mathbf{X}$ is preserved in the low-dimensional embedding $\mathbf{Z}$, the authors further required that the linear projector satisfy the constraint that $\mathbf{P}_l\mathbf{P}_l^\top = \mathbf{I}$ and added an additional term to the objective with form $\|\mathbf{X} - \mathbf{P}_l^\top\mathbf{P}_l\mathbf{X}\|_F^2$.

More recently, given the recent success of deep neural networks, a large number of studies Peng et al. (2017); Ji et al. (2017); Zeng et al. (2019b;a); Xie et al. (2020); Sun et al. (2019); Li et al. (2019); Yang et al. (2019); Jiang et al. (2019); Tang et al. (2018); Kheirandishfard et al. (2020b); Zhou et al. (2019); Zhang et al. (2019b); Jiang et al. (2018); Abavisani & Patel (2018); Zhou et al. (2018); Zhang et al. (2018; 2019c;a); Kheirandishfard et al. (2020a) have attempted to learn an appropriate embedding of the data (which ideally would have a union of linear subspaces structure) via a neural network, $\Phi_E(\mathbf{X}, \mathcal{W}_e)$, where $\mathcal{W}_e$ denotes the parameters of a network mapping defined by Φ_E, which takes a dataset $\mathbf{X} \in \mathbb{R}^{d_x \times N}$ as input. In an attempt to encourage the embedding of the data, $\Phi_E(\mathbf{X}, \mathcal{W}_e)$, to have this union of subspaces structure, these approaches minimize a self-expressive loss term, with form given in (1), on the embedded data, and a large majority of these proposed techniques can be described by the general form:

$$\min_{\mathcal{W}_e, \mathbf{C}} \gamma F(\mathbf{Z}, \mathbf{C}) + g(\mathbf{Z}, \mathbf{X}, \mathbf{C}) \;\text{ subject to }\; \mathbf{Z} = \Phi_E(\mathbf{X}, \mathcal{W}_e) \tag{2}$$

where g is some function designed to discourage trivial solutions (for example $\Phi_E(\mathbf{X}, \mathcal{W}_e) = \mathbf{0}$) and $\gamma > 0$ is some hyper-parameter to balance the terms.

Several different choices of g have been proposed in the literature. The first is to place some form of normalization directly on $\mathbf{Z}$. For example, the authors of Peng et al. (2017) propose an **Instance Normalization** regularization, which attempts to constrain the norm of the embedded data points to be 1, which takes the form $g(\mathbf{Z}, \mathbf{X}, \mathbf{C}) = \sum_{i=1}^N (\mathbf{Z}_i^\top\mathbf{Z}_i - 1)^2$. Likewise, one could also consider **Dataset Normalization** schemes, which bound the norm of the entire embedded representation $\|\mathbf{Z}\|_F^2 \geq \tau$ or **Channel/Batch Normalization** schemes, which bound the norm of a channel of the

embedded representation (i.e., a row of $\mathbf{Z}$), $\|\mathbf{Z}^i\|_F^2 \geq \tau, \ \forall i$. We note that this is quite similar to the common Batch Norm operator Ioffe & Szegedy (2015) used in neural network training which attempts to constrain each row of $\mathbf{Z}$ to have zero mean and constant norm.

Another popular form of g is to also train a decoding network $\Phi_D(\cdot, \mathcal{W}_d)$ with parameters $\mathcal{W}_d$ to map the self-expressive representation, $\Phi_E(\mathbf{X}, \mathcal{W}_e)\mathbf{C}$, back to the original data to ensure that sufficient information is preserved in the self-expressive representation to recover the original data. This idea is essentially a generalization of the previously discussed work, which considered constrained linear encoder/decoder mappings Patel et al. (2013); Patel & Vidal (2014); Patel et al. (2015), to non-linear autoencodering neural networks and was first proposed by the authors of Ji et al. (2017). We will refer to this as **Autoencoder Regularization**, and the problem takes the general form:

$$\min_{\mathcal{W}_e, \mathcal{W}_d, \mathbf{C}} \gamma F(\mathbf{Z}, \mathbf{C}) + \ell(\mathbf{X}, \Phi_D(\mathbf{Z}\mathbf{C}, \mathcal{W}_d)) \ \text{ subject to } \ \mathbf{Z} = \Phi_E(\mathbf{X}, \mathcal{W}_e), \tag{3}$$

where the first term is the standard self-expressive subspace clustering loss applied to the embedded representation, and the second term is a standard auto-encoder loss, with ℓ typically chosen to be the squared loss. Note that here both the kernel (in the form of the encoding network) and the optimal self-expression encoding, $\mathbf{C}$, are trained jointly, and once problem (3) is solved one can use the recovered $\mathbf{C}$ matrix directly for clustering.

1.2 Paper Contributions

Using the general formulation in (2) and the popular specific case in (3), Self-Expressive Deep Subspace Clustering (SEDSC) has been applied to a variety of applications, but there is relatively little that it known about it from a theoretical standpoint. Initial formulations for SEDSC were guided by the intuition that if the dataset is drawn from a union of linear subspaces, then solving problem (1) is known to induce desirable properties in $\mathbf{C}$ for clustering. By extension one might assume that if one also optimizes over the geometry of the learned embedding ($\mathbf{Z}$) this objective might induce a desirable geometry in the embedded space (e.g., a union of linear subspaces). However, a vast majority of the prior theoretical analysis for problems of the form in (1) only considers the case where the data is held fixed and analyze the properties of the optimal $\mathbf{C}$ matrix. Due to the well-known fact that neural networks are capable of producing highly-expressive mapping functions (and hence a network could produce many potential values for $\mathbf{Z}$), the use of a model such as (2)/(3) is essentially using (1) as a *regularization function* on $\mathbf{Z}$ to encourage a union of subspaces geometry. To date, however, models such as (2)/(3) have been guided largely by intuition and significant questions remain regarding what type of data geometry is encouraged by $F(\mathbf{Z}, \mathbf{C})$ when one optimizes over both the encoding matrix, $\mathbf{C}$, and the network producing the embedded data representation, $\mathbf{Z}$.

Here we explore these questions via theoretical analysis where we show that the use of $F(\mathbf{Z}, \mathbf{C})$ as a regularization function when learning a kernel from the data in an attempt to promote beneficial data geometries in the embedded space, as is done in (2), is largely insufficient in the sense that the optimal data geometries are trivial and not conducive to successful clustering in most cases. Specifically, we first note a basic fact that the Autoencoder Regularization formulation in (3) is typically ill-posed for most commonly used networks if constraints are not placed on the magnitude of the embedded data, $\Phi_E(\mathbf{X}, \mathcal{W}_e)$, either through regularization/constraints on the autoencoder weights/architecture or an explicit normalization of $\Phi_E(\mathbf{X}, \mathcal{W}_e)$, such as in the Instance/Batch/Dataset Normalization schemes. Then, even assuming that the embedded representation has been suitably normalized, we show that the optimal embedded data geometry encouraged by $F(\mathbf{Z}, \mathbf{C})$ is trivial in various ways, which will depend on how the data is normalized. We illustrate our theoretical predictions with experiments on both real and synthetic data. Further, we show experimentally that much of the claimed performance benefit of the SEDSC model which has been reported in previous work can be attributed to an ad-hoc post-processing of the $\mathbf{C}$ matrix first proposed in Ji et al. (2017).

Notation. We will denote matrices with capital boldfaced letters, $\mathbf{Z}$, vectors which are not rows/columns of a larger matrix with lower-case boldfaced letters, $\mathbf{z}$, and sets with calligraphic letters, $\mathcal{Z}$. The ith row of a matrix will be denoted with a superscript, $\mathbf{Z}^i$; the ith column of a matrix will be denoted with a subscript, $\mathbf{Z}_i$; and the $(i,j)^{\text{th}}$ entry of a matrix will be denoted as $\mathbf{Z}_{i,j}$. We will denote the minimum singular value of a matrix $\mathbf{Z}$ as $\sigma_{\min}(\mathbf{Z})$, and we will denote the nuclear, spectral, Frobenius, and ℓ_p norms for a matrix/vector as $\|\mathbf{Z}\|_*$, $\|\mathbf{Z}\|_*^\circ$, $\|\mathbf{Z}\|_F$, and $\|\mathbf{Z}\|_p$, respectively. $\delta(cnd)$ denotes an indicator function which takes value 0 if condition cnd is true and ∞ otherwise.

2 THEORETICAL ANALYSIS

We begin our analysis of the SEDSC model by considering the most popular formulation, which is to employ Autoencoder Regularization as in (3). Specifically, we note that without any regularization on the autoencoder network parameters $(\mathcal{W}_e, \mathcal{W}_d)$ or any normalization placed on the embedded representation, $\Phi_E(\mathbf{X}, \mathcal{W}_e)$, the formulation in (3) is often ill-posed in the sense that the value of $F(\Phi_E(\mathbf{X}, \mathcal{W}_e), \mathbf{C})$ can be made arbitrarily small without changing the value of the autoencoder loss by simply scaling-down the weights in the encoding network and scaling-up the weights in the decoding network in a way which doesn't change the output of the autoencoder but reduces the magnitude of the embedded representation.

As a basic example, consider fully-connected single-hidden-layer networks for the encoder/decoder, $\Phi_E(\mathbf{X}, \mathcal{W}_e) = \mathbf{W}_e^2(\mathbf{W}_e^1\mathbf{X})_+$ and $\Phi_D(\mathbf{Z}, \mathcal{W}_d) = \mathbf{W}_d^2(\mathbf{W}_d^1\mathbf{Z})_+$ where $(x)_+ = \max\{x, 0\}$ denotes the Rectified Linear Unit (ReLU) non-linearity applied entrywise. Then, note that for any $\alpha \geq 0$ one has $\Phi_E(\mathbf{X}, \alpha\mathcal{W}_e) = (\alpha\mathbf{W}_e^2)(\alpha\mathbf{W}_e^1\mathbf{X})_+ = \alpha^2\mathbf{W}_e^2(\mathbf{W}_e^1\mathbf{X})_+ = \alpha^2\Phi_E(\mathbf{X}, \mathcal{W}_e)$ and similarly for Φ_D. As a result one can scale $\mathcal{W}_e$ by any $\alpha > 0$ and $\mathcal{W}_d$ by α^{-1} without changing the output of the autoencoder, $\Phi_D(\Phi_E(\mathbf{X}, \alpha\mathcal{W}_e), \alpha^{-1}\mathcal{W}_d) = \Phi_D(\Phi_E(\mathbf{X}, \mathcal{W}_e), \mathcal{W}_d)$, but $\|\Phi_E(\mathbf{X}, \alpha\mathcal{W}_e)\|_F = \alpha^2\|\Phi_E(\mathbf{X}, \mathcal{W}_e)\|_F$. From this, taking α to be arbitrarily small the magnitude of the embedding can also be made arbitrarily small, so if $\theta(\mathbf{C})$ is something like a norm (as is typically the case) the value of $F(\mathbf{Z}, \mathbf{C})$ in (3) can also be made arbitrarily small without changing the reconstruction loss of the autoencoder ℓ. This implies one is simply training an autoencoder without any contribution from the $F(\mathbf{Z}, \mathbf{C})$ term (other than trying to reduce the magnitude of the embedded representation). This basic idea can be easily generalized and formalized in the following statement:

Proposition 1. *Consider the objective in* (3) *and suppose* $\theta(\mathbf{C})$ *is a function which achieves its minimum at* $\mathbf{C} = \mathbf{0}$ *and satisfies* $\theta(\mu\mathbf{C}) < \theta(\mathbf{C}),\ \forall\mu \in (0, 1)$. *If for any choice of* $(\mathcal{W}_d, \mathcal{W}_e)$ *and* $\tau_1, \tau_2 \in (0, 1)$ *in there exists* $(\hat{\mathcal{W}}_d, \hat{\mathcal{W}}_e)$ *such that* $\Phi_E(\mathbf{X}, \hat{\mathcal{W}}_e) = \tau_1\Phi_E(\mathbf{X}, \mathcal{W}_e)$ *and* $\Phi_D(\tau_2\mathbf{Z}, \hat{\mathcal{W}}_d) = \Phi_D(\mathbf{Z}, \mathcal{W}_d)$ *then the* F *term in* (3) *can be made arbitrarily small without changing the value of the loss function* ℓ. *Further, if the final layers of the encoding and decoding networks are positively-homogeneous with degree* $p \neq 0$, *such a* $(\hat{\mathcal{W}}_d, \hat{\mathcal{W}}_e)$ *will always exist simply by scaling the network weights of the linear (or affine) operator parameters.*

Proof. All proofs are provided in the Supplementary Material. □

Recall, a positively-homogeneous function of degree p is any function which satisfies $f(\alpha x) = \alpha^p f(x),\ \forall\alpha \geq 0$, and that most commonly used non-linearities in neural networks are positively homogenous, with Rectified Linear Units (ReLUs), leaky ReLUs, and max-pooling being common examples (note that the above proposition also applies to networks employing bias terms). As a result, most autoencoder architectures employed for SEDSC require some form or regularization or normalization for the F term to have any impact on the geometry of the embedded representation (other than trivially becoming smaller in magnitude). In practice, most of the previously cited studies employing the SEDSC model use networks which satisfy the conditions of Proposition 1 without regularization, and we note that such models can never be solved to global optimality (only approach it asymptotically) and are inherently ill-posed.

From this, one can conclude that Autoencoder Regularization by itself is often insufficient to prevent trival solutions to (2). This specific issue can be easily fixed (although prior work often does not) if one ensures that the magnitude of the embedded representation is constrained to be larger than some minimum value - either through regularization/constraints placed on the network weights, such as using weight decay or coupling weights between the encoder and decoder, or explicit normalization of the embedded representation. However, the question remains what geometries are promoted by (1) even if the basic issue described above is corrected, which we explore next.

2.1 THE EFFECT OF THE AUTOENCODER LOSS

Before presenting the remainder of our formal results, we first pause to discuss the effect of the autoencoder loss (ℓ) in (3). The use of the autoencoder loss is designed to ensure that the embedded representation $\mathbf{Z}$ retains sufficient information about the original data $\mathbf{X}$ so that the data can be reconstructed by the decoder, but we note that this *does not* necessarily impose significant constraints on the geometric arrangement of the embedded points in $\mathbf{Z}$. While there is potentially some constraint on the possible geometries of $\mathbf{Z}$ which can be expressed by the choice of encoder/decoder

architectures, for reasonably expressive encoders and decoders data points can be mapped essentially arbitrarily into the embedded space and still decoded accurately, provided the embedding of any two points is still distinct to within some ϵ perturbation. Further, because we only evaluate the mapping on a finite set of points $\mathbf{X}$, this is a much easier (and can be achieved with much simpler networks) than the well-known universal approximation regime of neural networks (which requires good approximation over the entire continuous data domain), and it is well documented that typical network architectures can optimally fit fairly arbitrary finite training data Zhang et al. (2017).

Given the above discussion, in our analysis we will first focus on the setting where the autoencoder is highly expressive. In this regime, the autoencoder can select weight parameters to produce an essentially arbitrary choice of $\mathbf{Z}$ embedding, so the global optimum of the model in (3) (and likewise (2)) will approach a choice of parameters which minimizes $F(\mathbf{Z}, \mathbf{C})$. As we already know from Prop 1, this is ill-posed without some additional form of regularization, so in the following subsections we explore optimal solutions to $F(\mathbf{Z}, \mathbf{C})$ when one optimizes over both $\mathbf{C}$ and the embedded representation $\mathbf{Z} = \Phi_E(\mathbf{X}, \mathcal{W}_e)$ subject to three different constraints on $\mathbf{Z}$: (1) Dataset Normalization $\|\mathbf{Z}\|_F^2 \geq \tau$, (2) Channel/Batch Normalization $\|\mathbf{Z}^i\|_2 = 1 \ \forall i$, and (3) Instance Normalization $\|\mathbf{Z}_i\|_2 = 1 \ \forall i$. Finally, after characterizing solutions to $F(\mathbf{Z}, \mathbf{C})$ we then give a specific example of a very simple (i.e., not highly expressive) family of architectures which can achieve these solutions.

2.2 Dataset and Batch/Channel Normalization

We will first consider the Dataset and Batch/Channel Normalization schemes, which will both result in very similar optimal solutions for the embedded data geometry. Recall, that this considers when the entire dataset is constrained to have a norm greater than some minimum value in the embedded space or when the norms of the rows of the dataset are constrained to have a norm greater than some minimum value. We note that the latter case is very closely related to batch normalization Ioffe & Szegedy (2015), which requires that each channel/feature (in this case a row of the embedded representation) has zero mean and constant norm in expectation over the draw of a mini-batch. Additionally, while we do not explicitly enforce a zero-mean constraint, we will see that optimal solutions will exist which have zero mean. Now, if one optimizes $F(\mathbf{Z}, \mathbf{C})$ jointly over $(\mathbf{Z}, \mathbf{C})$ subject to the above constraint(s) on $\mathbf{Z}$, then the following holds:

Theorem 1. *Consider the following optimization problems which jointly optimize over* $\mathbf{Z}$ *and* $\mathbf{C}$*:*

$$\text{(P1)} \quad \min_{\mathbf{C},\mathbf{Z}} \left\{ F(\mathbf{Z}, \mathbf{C}) \ \text{ s.t. } \ \|\mathbf{Z}\|_F^2 \geq \tau \right\} \qquad \text{(P2)} \quad \min_{\mathbf{C},\mathbf{Z}} \left\{ F(\mathbf{Z}, \mathbf{C}) \ \text{ s.t. } \ \|\mathbf{Z}^i\|_F^2 \geq \tfrac{\tau}{d} \ \ \forall i \right\} \tag{4}$$

Then optimal values for $\mathbf{C}$ *for both* (P1) *and* (P2) *are given by*

$$\mathbf{C}^* \in \arg\min_{\mathbf{C}} \tfrac{1}{2}\sigma_{\min}^2(\mathbf{C} - \mathbf{I})\tau + \lambda\theta(\mathbf{C}). \tag{5}$$

Moreover, for any optimal $\mathbf{C}^*$*, let* r *be the multiplicity of the smallest singular value of* $\mathbf{C}^* - \mathbf{I}$ *and let* $\mathbf{Q} \in \mathbb{R}^{N \times r}$ *be an orthonormal basis for the subspace spanned by the left singular vectors associated with the smallest singular value of* $\mathbf{C}^* - \mathbf{I}$*. Then we have that optimal values for* $\mathbf{Z}$ *are given by:*

$$\mathbf{Z}^* \in \{\mathbf{B}\mathbf{Q}^\top \colon \mathbf{B} \in \mathbb{R}^{d \times r} \cap \mathcal{B}\}, \ \ \mathcal{B} = \begin{cases} \begin{cases} \{\mathbf{B} : \|\mathbf{B}\|_F^2 = \tau\} & \sigma_{\min}(\mathbf{C}^* - \mathbf{I}) > 0 \\ \{\mathbf{B} : \|\mathbf{B}\|_F^2 \geq \tau\} & \sigma_{\min}(\mathbf{C}^* - \mathbf{I}) = 0 \end{cases} & \text{(P1)} \\ \begin{cases} \{\mathbf{B} : \|\mathbf{B}^i\|_F^2 = \frac{\tau}{d}, \forall i\} & \sigma_{\min}(\mathbf{C}^* - \mathbf{I}) > 0 \\ \{\mathbf{B} : \|\mathbf{B}^i\|_F^2 \geq \frac{\tau}{d}, \forall i\} & \sigma_{\min}(\mathbf{C}^* - \mathbf{I}) = 0 \end{cases} & \text{(P2)} \end{cases} \tag{6}$$

From the above result, one notes from (5) that optimizing $F(\mathbf{Z}, \mathbf{C})$ jointly over both $\mathbf{Z}$ and $\mathbf{C}$ is equivalent to finding a $\mathbf{C}$ which minimizes a trade-off between the minimum singular value of $\mathbf{C} - \mathbf{I}$ and the regularization $\theta(\mathbf{C})$. Further, we note that if such an optimal $\mathbf{C}$ results in the minimum singular value of $\mathbf{C} - \mathbf{I}$ having a multiplicity of 1, then this implies that every data point in $\mathbf{Z}$ will simply be a scaled version of the same point. Obviously, such an embedding is not useful for subspace clustering. Characterizing optimal solutions to (5) is somewhat complicated in the general case due to the fact that the smallest singular value is a concave function of a matrix and θ is typically chosen to be a convex regularization function, resulting in the minimization of a convex+concave function. Instead, we will focus on the most commonly used choice of regularization, $\theta_{SSC}(\mathbf{C}) = \|\mathbf{C}\|_1 + \delta(\text{diag}(\mathbf{C}) = \mathbf{0})$, where we derive the optimal solution in the case where $\sigma_{min}(\mathbf{C} - \mathbf{I}) = 0$.

We note that this corresponds to the case where $\mathbf{Z}^* = \mathbf{Z}^*\mathbf{C}^*$ which one typically obtains as λ in (1) is taken to be small. We also provide results for other popular choices of $\theta(\mathbf{C}) = \|\mathbf{C}\|_*$ and $\theta(\mathbf{C}) = \|\mathbf{C}\|_F$, which have qualitatively similar optimal geometries, in the Supplementary Material.

Theorem 2. *Optimal solutions to the problems*

$$\text{(P1)} \quad \min_{\mathbf{Z},\mathbf{C}} \|\mathbf{C}\|_1 \ \text{ s.t. } \ \text{diag}(\mathbf{C}) = \mathbf{0},\ \mathbf{Z} = \mathbf{Z}\mathbf{C},\ \|\mathbf{Z}\|_F^2 \geq \tau \tag{7}$$

$$\text{(P2)} \quad \min_{\mathbf{Z},\mathbf{C}} \|\mathbf{C}\|_1 \ \text{ s.t. } \ \text{diag}(\mathbf{C}) = \mathbf{0},\ \mathbf{Z} = \mathbf{Z}\mathbf{C},\ \|\mathbf{Z}^i\|_F^2 \geq \tfrac{\tau}{d}\ \ \forall i \tag{8}$$

are characterized by the set

$$(\mathbf{Z}^*, \mathbf{C}^*) \in \{[\mathbf{z} \quad \mathbf{z} \quad \mathbf{0}_{d\times N-2}]\,\mathbf{P}\} \times \left\{ \mathbf{P}^\top \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & & 0 \end{bmatrix} \mathbf{P} \right\}, \tag{9}$$

where $\mathbf{P} \in \mathbb{R}^{N\times N}$ is an arbitrary signed-permutation matrix and $\mathbf{z} \in \mathbb{R}^d$ is an arbitrary vector which satisfies $\|\mathbf{z}\|_F^2 \geq \tau/2$ for (P1) *and $\mathbf{z}_i^2 \geq \tau/(2d), \forall i \in [1, d]$ for* (P2).

From the above result, we have shown that if $\mathbf{Z}$ is normalized to have a lower-bounded norm on either the entire embedded representation or for each row, then the effect of the $F(\mathbf{Z}, \mathbf{C})$ loss will be largely similar to the situation described by Proposition 1 in the sense that the loss will still attempt to push all of the points to 0 with the exception of two points, which will be copies of each other (potentially to within a sign-flip). Again, the optimal embedded representation is clearly ill-posed for clustering since all but two of the points are trivially driven towards 0 in the embedded space.

2.3 Instance Normalization

To explicitly prevent the case where most of the points in the embedded space are trivially driven to 0 as in the prior two normalization schemes, another potential normalization strategy which has been proposed for (2) is to use Instance Normalization Peng et al. (2017), where the ℓ_2 norm of each embedded data point is constrained to be equal to some constant. Here again we will see that this results in somewhat trivial data geometries. Specifically, we will again focus on the choice of the $\theta_{SSC}(\mathbf{C})$ regularization function when we have exact equality, $\mathbf{Z} = \mathbf{Z}\mathbf{C}$, for simplicity of presentation. From this we have the following result:

Theorem 3. *Optimal solutions to the problem*

$$\min_{\mathbf{Z},\mathbf{C}} \|\mathbf{C}\|_1 \ \text{ s.t. } \ \text{diag}(\mathbf{C}) = \mathbf{0},\ \ \mathbf{Z} = \mathbf{Z}\mathbf{C},\ \ \|\mathbf{Z}_i\|_F^2 = \tau\ \ \forall i \tag{10}$$

must have the property that for any column in $\mathbf{Z}^*$, $\mathbf{Z}_i^*$, there exists another column, $\mathbf{Z}_j^*$ $(i \neq j)$, such that $\mathbf{Z}_i^* = s_{i,j}\mathbf{Z}_j^*$ where $s_{i,j} \in \{-1, 1\}$. Further, $\|\mathbf{C}_i^*\|_1 = 1\ \forall i$ and $\mathbf{C}_{i,j} \neq 0 \implies \mathbf{Z}_i^* = \pm\mathbf{Z}_j^*$.

The above result is quite intuitive in the sense that because a given point cannot use itself in its representation (due to the $\text{diag}(\mathbf{C}) = \mathbf{0}$ constraint), the next best thing is to have an exact copy of itself in another column. While this result is more conducive to successful clustering in the sense that points which are close in the embedded space are encouraged to 'merge' into a single point, there are still numerous pathological geometries that can result. Specifically, there is no constraint on the number of 'distinct' points in the representation (i.e., the number of vectors which are not copies of each other to within a sign flip), other than it must be less than $N/2$. As a result, the optimal $\mathbf{C}^*$ matrix can also contain an arbitrary number (in the range $[1, N/2]$) of connected components in the affinity graph, resulting in somewhat arbitrary spectral clustering.

Example of Degenerate Geometry with Simple Networks. In section 2.1 we discussed how if the encoder/decoder are highly expressive then the optimal embedding will approach the solutions we give in our theoretical analysis. Here we show that trivial embeddings can also occur with relatively simple encoders/decoders. Specifically, consider basic encoder/decoder architectures which consist of two affine mappings with a ReLU non-linearity (denoted as $(\cdot)_+$) on the hidden layer:

$$\Phi_E(\mathbf{x}, \mathcal{W}_e) = \mathbf{W}_e^2(\mathbf{W}_e^1\mathbf{x} + \mathbf{b}_e^1)_+ + \mathbf{b}_e^2 \qquad \Phi_D(\mathbf{z}, \mathcal{W}_d) = \mathbf{W}_d^2(\mathbf{W}_d^1\mathbf{z} + \mathbf{b}_d^1)_+ + \mathbf{b}_d^2 \tag{11}$$

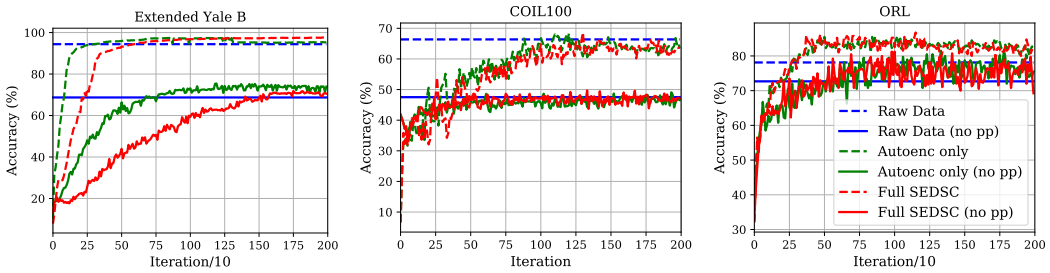

Figure 1: Clustering accuracy results for YaleB (38 faces), COIL100, and ORL datasets with **(Dashed Lines)** and without **(Solid Lines)** the post-processing step on $\mathbf{C}$ matrix proposed in Ji et al. (2017). **(Raw Data)** clustering on the raw data. **(Autoenc only)** Clustering features from a pre-trained autoencoder. **(Full SEDSC)** The full model in (3).

where the linear operators ($\mathbf{W}$ matrices) can optionally be constrained (for example for convolution operations $\mathbf{W}$ could be required to be a Toeplitz matrix). Now if we have that the embedded dimension d is equal to the data dimension d_x we will say that linear operators can **express identity** if there exists parameters $(\mathbf{W}^2, \mathbf{W}^1)$ such that $\mathbf{W}^2\mathbf{W}^1 = \mathbf{I}$. Note that if the architectures in (11) are fully-connected this implies that for a general $\mathbf{X}$ the number of hidden units is greater than or equal to $d = d_x$, while if the linear operators are convolutions, then we only need one convolutional channel (with the kernel being the delta function). Within this setting we have the following result:

Proposition 2. *Consider encoder and decoder networks with the form given in* (11) *where the linear operators in both the encoder/decoder can express identity. Then, given any dataset* $\mathbf{X} \in \mathbb{R}^{d_x \times N}$ *there exist network parameters* $(\mathcal{W}_e, \mathcal{W}_d)$ *which satisfy the following:*

1. *Embedded points are arbitrarily close:* $\|\Phi_E(\mathbf{X}_i, \mathcal{W}_e) - \Phi_E(\mathbf{X}_j, \mathcal{W}_e)\| \leq \epsilon \; \forall (i,j)$ *and* $\forall \epsilon > 0$.
2. *Embedded points have norm arbitrarily close to 1:* $|\|\Phi_E(\mathbf{X}_i, \mathcal{W}_e)\|_F - 1| \leq \epsilon \; \forall i$ *and* $\forall \epsilon > 0$.
3. *Embedded points can be decoded exactly:* $\Phi_D(\Phi_E(\mathbf{X}_i, \mathcal{W}_e), \mathcal{W}_d) = \mathbf{X}_i, \; \forall i$.

From the above simple example, we can see that even with very simple network architectures (i.e., not necessarily highly expressive) it is still possible to have solutions which are arbitrarily close to the global optimum described in Theorem 3 while still having a perfect reconstruction of $\mathbf{X}$.

3 Experiments

Here, we present experiments on both real and synthetic data that verify our theoretical predictions experimentally. We first evaluate the Autoencoder Regularization form given in (3) by repeating all of the experiments from Ji et al. (2017), showing that the optimization problem never reaches a stationary point due to the pathology described by Prop 1 (see Supplementary Material) and that the improvement in performance is largely attributable to an ad-hoc post-processing step. Then, we present experiments on a simple synthetic dataset to illustrate our theoretical results.

3.1 Repeating the Experiments of Ji et al. (2017)

First we use the code provided by the authors of Ji et al. (2017) to repeat all of their original clustering experiments on the Extended Yale-B (38 faces), ORL, and COIL100 datasets. As baseline methods, we also train the autoencoder network without the $F(\mathbf{Z}, \mathbf{C})$ term (i.e., $\gamma = 0$ and $\mathbf{C}$ fixed at $\mathbf{I}$ in (3)), then using the autoencoder as a fixed feature extractor perform subspace clustering on embedded features. Likewise, we also perform subspace clustering on the raw data (see Supplementary Material for further details).

Table 1: Clustering accuracy shown in Fig 1. To be consistent with Ji et al. (2017), we report the results at $1000/120/700$ iterations for Yale B / COIL100 / ORL, respectively.

	With post-processing			Without post-processing		
	Raw Data	Autoenc only	Full SEDSC	Raw Data	Autoenc only	Full SEDSC
YaleB	94.40%	**97.12**%	96.79%	68.71%	**71.96**%	59.09%
COIL100	66.47%	**68.26**%	64.96%	**47.51**%	44.84%	45.67%
ORL	78.12%	83.43%	**84.10**%	72.68%	**73.73**%	73.50%

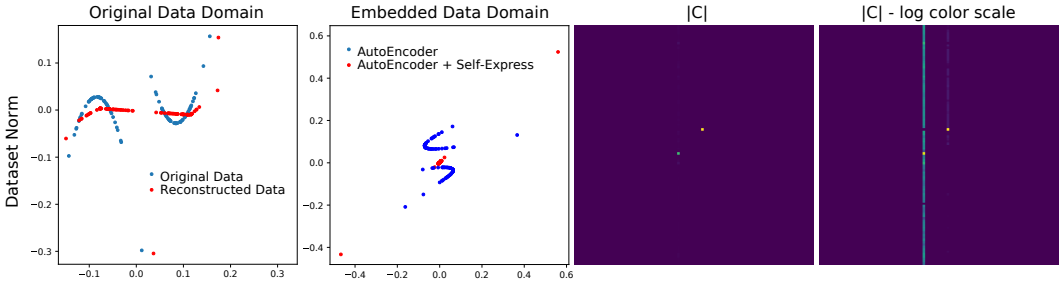

Figure 2: Results for synthetic data using the dataset normalization scheme. **(Left)** Original data points (Blue) and the data points at the output of the autoencoder when the full model (3) is used (Red). **(Center Left)** Data representation in the embedded domain when just the autoencoder is trained without the $F(\mathbf{Z}, \mathbf{C})$ term (Blue) and the full SEDSC model is used (Red). **(Center Right)** The absolute value of the recovered $\mathbf{C}$ encoding matrix when trained with the full model. **(Right)** Same plot as the previous column but with a logarithmic color scale to visualize small entries.

In addition to proposing the model in (3), the authors of Ji et al. (2017) also implement a somewhat arbitrary post-processing of the $\mathbf{C}$ matrix recovered from the SEDSC model before the final spectral clustering, which involves 1) entrywise hard thresholding, 2) applying the shape interaction matrix method Costeira & Kanade (1995) to $\mathbf{C}$, and 3) raising $\mathbf{C}$ to a power, entrywise. Likewise, many subsequent works on SEDSC follow Ji et al. (2017) and employ very similar post-processing steps on $\mathbf{C}$. As shown in Figure 1 and Table 1 there is little observed benefit for using SEDSC, as it achieves comparable (or worse) performance than baseline methods in almost all settings when the post-processing of $\mathbf{C}$ is applied consistently (or not) across all methods.

3.2 Synthetic Data Experiments

Finally, to illustrate our theoretical predictions we construct a simple synthetic dataset which consists of 100 points drawn from the union of two parabolas in $\mathbb{R}^2$, where the space of the embedding is also $\mathbb{R}^2$. We then train the model given in (3) with $\theta(\mathbf{C}) = \theta_{SSC}(\mathbf{C}) = \|\mathbf{C}\|_1 + \delta(\text{diag}(\mathbf{C}) = \mathbf{0})$, the encoder/decoder networks being simple single-hidden-layer fully-connected networks with 100 hidden units, and ReLU activations on the hidden units. Fig 2 shows the resulting solution that results when we directly add a normalization operator to the encoder network which normalizes the output of the encoder to have unit Frobenius norm over the entire dataset (Dataset Normalization). Additional experiments for other normalization schemes and a description of the details of our synthetic data experiments can be found in the Supplementary Material.

From Figure 2 one can see that our theoretical predictions are largely confirmed experimentally. Namely, one sees that when the full SEDSC model is trained the embedded representation is very close to as predicted by Theorem 2, with almost all of the embedded points (Left Center - Red points) being near the origin with the exception of two points, which are co-linear with each other. Likewise, the $\mathbf{C}$ matrix is dominated by two non-zero entries with the remaining non-zero entries only appearing on the log-scale color scale. We note that as this is a very simple dataset (i.e., two parabolas without any added noise) one would expect most reasonable manifold clustering/learning algorithms to succeed; however, due to the deficiencies of the SEDSC model we have shown in our analysis a trivial solution results.

4 Conclusions

We have presented a theoretical and experimental analysis of the Self-Expressive Deep Subspace Clustering (SEDSC) model. We have shown that in many cases the SEDSC model is ill-posed and results in trivial data geometries in the embedded space. Further, our attempts to replicate previously reported experiments lead us to conclude that much of the claimed benefit of SEDSC is attributable to other factors such as post-processing of the recovered encoding matrix, $\mathbf{C}$, and not the SEDSC model itself. Overall, we conclude that considerably more attention needs to be given to the issues we have raised in this paper in terms of both how models for this problem are designed and how they are evaluated to ensure that one arrives at meaningful solutions and can clearly demonstrate the performance of the resulting model without other confounding factors.

REFERENCES

Mahdi Abavisani and Vishal M Patel. Deep multimodal subspace clustering networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1601–1614, 2018.

Joao Costeira and Takeo Kanade. A multi-body factorization method for motion analysis. In *IEEE International Conference on Computer Vision*, pp. 1071–1076. IEEE, 1995.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2790–2797, 2009.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *Advances in Neural Information Processing Systems*, pp. 24–33, 2017.

Yangbangyan Jiang, Zhiyong Yang, Qianqian Xu, Xiaochun Cao, and Qingming Huang. When to learn what: Deep cognitive subspace clustering. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 718–726, 2018.

Yangbangyan Jiang, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Duet robust deep subspace clustering. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 1596–1604, 2019.

Mohsen Kheirandishfard, Fariba Zohrizadeh, and Farhad Kamangar. Deep low-rank subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 864–865, 2020a.

Mohsen Kheirandishfard, Fariba Zohrizadeh, and Farhad Kamangar. Multi-level representation learning for deep subspace clustering. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 2039–2048, 2020b.

Chun-Guang Li, Chong You, and René Vidal. On geometric analysis of affine sparse subspace clustering. *IEEE Journal on Selected Topics in Signal Processing*, 12(6):1520–1533, 2018.

Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, and Qinghua Hu. Reciprocal multi-layer subspace learning for multi-view clustering. In *IEEE International Conference on Computer Vision*, pp. 8172–8180, 2019.

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *European Conference on Computer Vision*, pp. 347–360, 2012.

N. Parikh and S. Boyd. Proximal Algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

V. M. Patel and R. Vidal. Kernel sparse subspace clustering. In *IEEE International Conference on Image Processing*, pp. 2849–2853, 2014.

V. M. Patel, H. V. Nguyen, and R. Vidal. Latent space sparse subspace clustering. In *IEEE International Conference on Computer Vision*, pp. 225–232, 2013.

V. M. Patel, H. V. Nguyen, and R. Vidal. Latent space sparse and low-rank subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):691–701, 2015.

Xi Peng, Jiashi Feng, Shijie Xiao, Jiwen Lu, Zhang Yi, and Shuicheng Yan. Deep sparse subspace clustering. *arXiv preprint arXiv:1709.08374*, 2017.

Daniel P Robinson, Rene Vidal, and Chong You. Basis pursuit and orthogonal matching pursuit for subspace-preserving recovery: Theoretical analysis. *arXiv preprint arXiv:1912.13091*, 2019.

Mahdi Soltanolkotabi and Emmanuel J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2012.

Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *Annals of Statistics*, 42(2):669–699, 2014.

Xiukun Sun, Miaomiao Cheng, Chen Min, and Liping Jing. Self-supervised deep multi-view subspace clustering. In *Asian Conference on Machine Learning*, pp. 1001–1016, 2019.

Xiaoliang Tang, Xuan Tang, Wanli Wang, Li Fang, and Xian Wei. Deep multi-view sparse subspace clustering. In *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, pp. 115–119, 2018.

Manolis C. Tsakiris and René Vidal. Theoretical analysis of sparse subspace clustering with missing entries. In *International Conference on Machine Learning*, pp. 4975–4984, 2018.

René Vidal, Yi Ma, and Shankar Sastry. *Generalized Principal Component Analysis*. Springer Verlag, 2016.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Yining Wang, Yu-Xiang Wang, and Aarti Singh. A deterministic analysis of noisy sparse subspace clustering for dimensionality-reduced data. In *International Conference on Machine Learning*, pp. 1422–1431, 2015.

Yu-Xiang Wang and Huan Xu. Noisy sparse subspace clustering. *Journal of Machine Learning Research*, 17(12):1–41, 2016.

Yuan Xie, Jinyan Liu, Yanyun Qu, Dacheng Tao, Wensheng Zhang, Longquan Dai, and Lizhuang Ma. Robust kernelized multiview self-representation for subspace clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Shuai Yang, Wenqi Zhu, and Yuesheng Zhu. Residual encoder-decoder network for deep subspace clustering. *arXiv preprint arXiv:1910.05569*, 2019.

C. You and R. Vidal. Subspace-sparse representation. *Arxiv*, abs/1507.01307, 2015a.

Chong You and René Vidal. Geometric conditions for subspace-sparse recovery. In *International Conference on Machine Learning*, pp. 1585–1593, 2015b.

Chong You, Chun-Guang Li, Daniel P. Robinson, and René Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3928–3937, 2016.

Meng Zeng, Yaoming Cai, Zhihua Cai, Xiaobo Liu, Peng Hu, and Junhua Ku. Unsupervised hyperspectral image band selection based on deep subspace clustering. *IEEE Geoscience and Remote Sensing Letters*, 16(12):1889–1893, 2019a.

Meng Zeng, Yaoming Cai, Xiaobo Liu, Zhihua Cai, and Xiang Li. Spectral-spatial clustering of hyperspectral image based on laplacian regularized deep subspace clustering. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2694–2697. IEEE, 2019b.

C. Zhang, S. Bengio, M. Hardt, and B. Recht. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. Generalized latent multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):86–99, 2018.

Junjian Zhang, Chun-Guang Li, Chong You, Xianbiao Qi, Honggang Zhang, Jun Guo, and Zhouchen Lin. Self-supervised convolutional subspace clustering network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5473–5482, 2019a.

Tong Zhang, Pan Ji, Mehrtash Harandi, Wenbing Huang, and Hongdong Li. Neural collaborative subspace clustering. *arXiv preprint arXiv:1904.10596*, 2019b.

Tong Zhang, Pan Ji, Mehrtash Harandi, Huang Wenbing, and Hongdong Li. Neural collaborative subspace clustering. In *ICML*, 2019c.

Lei Zhou, Bai Xiao, Xianglong Liu, Jun Zhou, Edwin R Hancock, et al. Latent distribution preserving deep subspace clustering. In *28th International Joint Conference on Artificial Intelligence*. York, 2019.

Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1596–1604, 2018.